Université d'Angers

Habilitation à diriger des recherches

Contributions à l'analyse de données médicales : méthodes et implémentations

Jean-Baptiste Fasquel
Jean-Baptiste.Fasquel@univ-angers.fr

Soutenue le 30 Août 2018 devant le jury suivant :

Isabelle Bloch	Rapporteur	Professeure, Télécom ParisTech
Olivier Lézoray	Rapporteur	Professeur, Université de Caen
Su Ruan	Rapporteur	Professeure, Université de Rouen
Regine Le Bouquin-Jeannès	Examinateur	Professeure, Université de Rennes 1
François Rousseau	Examinateur	Professeur, Institut Mines Télécom Atlantique
Luc Soler	Examinateur	Directeur scientifique, IRCAD, Strasbourg
Anne Humeau-Heurtier	Examinateur	Professeure, Université d'Angers

Table des matières

Résumé

Ce document résume et contextualise certaines de mes contributions scientifiques. Les travaux présentés sont centrés sur l'analyse de données médicales, principalement sous forme d'images, et couvrent trois volets, détaillés dans ce manuscrit. Le deux premiers concernent des méthodes d'analyse de données tandis que le dernier est dédié à la problématique de l'implémentation matérielle et logicielle.

Le premier volet est dédié à l'usage de connaissances a priori structurelles, exprimées de manière qualitative, afin de guider le processus d'interprétation des images. La question sous-jacente est la suivante. Comment, et pour quels bénéfices, exploiter ces connaissances qualitatives, éloignées des informations bas-niveau de l'image, pour aider à en analyser son contenu ?

Le second volet est dédié aux applications médicales. Ce volet est traité, dans le contexte de mes contributions, en considérant tout d'abord le traitement d'images médicales pour la modélisation pré-opératoire de patients et pour la quantification des flux sanguins dans les artères carotides. J'aborde ensuite succinctement une activité de recherche émergente, centrée sur l'utilisation des objets connectés en santé pour le suivi régulier et à distance de l'état physiologique de patients.

Le dernier volet se focalise sur l'implémentation matérielle et logicielle de méthodes d'analyse, en considérant également des fonctionnalités connexes que l'on rencontre par exemple dans le contexte des logiciels médicaux. Les enjeux sous-jacents sont la performance et la généricité des systèmes, avec des applications centrées sur le médical.

Table des figures

Introduction

Ce manuscrit a été rédigé en vue d'obtenir l'habilitation à diriger des recherches. Je présente et contextualise, dans ce document, mon activité de recherche et d'encadrement de la recherche depuis l'obtention du titre de docteur de l'université de Strasbourg en 2002. Ce document retrace ainsi mes travaux en tant que chargé de recherches de 2002 à 2009 à l'Institut de Recherche contre les Cancers de l'Appareil Digestif (IRCAD) de Strasbourg, puis en tant qu'enseignant-chercheur, depuis 2009, à l'université d'Angers, et plus précisément au Laboratoire Angevin de Recherche Ingénierie des Systèmes (LARIS).

Ce manuscrit traite de méthodes d'analyse de données principalement médicales. Un point important concerne l'utilisation de connaissances a priori structurelles exprimées de manière qualitative afin de guider l'interprétation d'images, notamment médicales. Même si ce document est essentiellement centré sur le traitement d'images, le terme "donnée" est privilégié afin d'englober des travaux naissants autour des objets connectés en santé. Un point important abordé dans ce manuscrit concerne la problématique de l'implémentation matérielle et logicielle d'algorithmes et de fonctionnalités connexes, autour de l'analyse de données.

Le document est structuré de la manière suivante :

- Le premier chapitre présente mon curriculum vitæainsi qu'un survol général de mes activités de recherche, en incluant les expériences d'encadrement et les autres activités connexes. Une bibliographie relative aux travaux auxquels j'ai participé est dressée en fin de chapitre.
- Le deuxième chapitre fait une synthèse plus détaillée de travaux autour des méthodes de traitement d'images principalement. La première partie porte sur un volet plus "théorique", autour de l'usage de connaissances structurelles qualitatives, et ayant pour vocation de proposer des méthodes génériques. La seconde partie présente des travaux plus appliqués, en soulignant en fin de chapitre mon implication récente autour des objets connectés en santé.
- Le troisième chapitre traite de mes travaux autour des problématiques de performance et de généricité auxquels on peut être confronté lors de l'implémentation matérielle et logicielle de méthodes de traitement. Ceci inclut, pour la partie logicielle, le couplage des algorithmes de traitement avec d'autres fonctionnalités connexes relevant, par exemple, des systèmes robotisés, des interfaces hommes-machines et de la visualisation 3D.
- La quatrième et dernière partie est consacrée au bilan et aux perspectives.

Sont fournis en annexe des articles publiés dans des revues internationales, ainsi que quelques communications récentes. Ceci permettra au lecteur de disposer d'une information plus détaillée sur les différents points contextualisés et synthétisés dans ce manuscrit.

Chapitre 1

Présentation synthétique

1.1 Curriculum Vitæ

1.1.1 Identité

- Jean-Baptiste Fasquel
- Né le 23 mars 1975 à Le Mans (72)
- Vie maritale, 2 enfants
- Adresse professionnelle : ISTIA - LARIS, 62, avenue Notre-Dame du Lac, 49000 Angers
- Téléphone : 33 (0)2 44 68 75 56
- Courriel : Jean-Baptiste.Fasquel@univ-angers.fr
- Site web : http ://perso-laris.univ-angers.fr/∼fasquel/

1.1.2 Formation et titres universitaires

1998-2002 : Doctorat Thèse de doctorat délivrée par l'université de Strasbourg. Titre : "Une méthode opto-informatique de détection et de reconnaissance d'objets d'intérêt : application à la détection des lésions cancéreuses du foie et à la vérification en temps réel des signatures manuscrites", sous la direction de M. Bruynooghe, Professeur d'informatique à l'Université de Strasbourg (Laboratoire des Systèmes Photoniques – maintenant intégré au laboratoire ICube). Cette thèse a été soutenue le 19 mars 2002 devant le jury suivant :

- Pr. C. Ronse (rapporteur), Professeur d'informatique de l'ULP de Strasbourg (Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection, Nouvellement ICube).
- Pr. P. Ambs (rapporteur), Président du jury et Professeur à l'Université de Haute Alsace de Mulhouse (Laboratoire "Fonctions Optiques et Traitement de l'Information").
- Pr. F. Prêteux (rapporteur), Professeur à l'Institut National des Télécommunications de Paris.
- Dr. L. Soler (examinateur), Professeur associé et Directeur de recherche à l'IRCAD, hôpital civil de Strasbourg.

1997-1998 : DEA Photonique et Traitement d'images de l'Université de Strasbourg. Sujet de recherche : "Système de mesure non destructif de la composition de plasmas". Stage de DEA réalisé à l'Université technique de Graz (Autriche).

1997-1998 : Diplôme d'ingénieur de l'école Télécom-Physique Strasbourg (anciennement Ecole Nationale Supérieure de Physique de Strasbourg).

1993-1995 : Classes préparatoires aux grandes écoles Classes de Mathématiques supérieures et spéciales, Lycée Montesquieu, Le Mans.

1.1.3 Fonctions occupées

Depuis 2009 : Maître de conférences à l'Université d'Angers (promu hors classe par le CNU en 2017).

- Enseignant à l'Institut des Sciences et Techniques de l'Ingénieur d'Angers (ISTIA), école d'ingénieurs de l'Université d'Angers.
- Chercheur au Laboratoire Angevin de Recherche en Ingénierie des Systèmes (LARIS) EA 7315 (anciennement Laboratoire d'Ingénierie des Systèmes Automatisés (LISA) EA 4094).

2002 - 2009 : Chargé de recherche de statut privé à l'Institut de Recherche contre les Cancers de l'Appareil Digestif (IRCAD), Strasbourg. Enseignant vacataire à l'Université de Strasbourg (UFR de sciences et école d'ingénieurs Télécom-Physique Strasbourg).

1998 - 2002 : Doctorant au Laboratoire des Systèmes Photoniques (maintenant intégré au laboratoire ICube) de l'Université de Strasbourg. Enseignant vacataire à l'Université de Strasbourg (école d'ingénieur Télécom-Physique Strasbourg).

1.2 Activités d'enseignement et charges administratives associées

Depuis 2009 : Enseignement à l'Université d'Angers, avec un volume annuel d'environ 350 heures équivalent TD. Les disciplines enseignées concernent l'informatique et le traitement d'images : UNIX, génie logiciel, développement web, interfaces homme-machine [1], réalité virtuelle[1], vision industrielle [2], réalité augmentée, imagerie médicale numérique. J'enseigne essentiellement en cycle ingénieur à l'ISTIA (bac+3 à bac+5), et marginalement à la faculté de médecine (imagerie médicale numérique en niveau L1, dans le cadre de la réforme du PACES).

1998-2009 : Enseignant vacataire à l'Université de Strasbourg, avec un volume annuel d'environ 90 heures équivalent TD. Les disciplines enseignées concernaient l'imagerie médicale, le traitement du signal et des images, le traitement optique de l'information, les mathématiques, la programmation et UNIX. J'enseignais à des publics de L3 à M2.

Responsabilités administratives en enseignement :

- Depuis 2011 : Responsable de la dernière année de l'option Systèmes Automatisés et Génie Informatique de l'ISTIA (Gestion des projets/stages des étudiants, des intervenants extérieurs, des semestres à l'étranger, participation à l'évolution du programme de formation, ...).
- Depuis 2012 : Membre de la commission de validation des acquis professionnels (VAP85).

1. http ://perso-laris.univ-angers.fr/∼fasquel/teaching_ihm.html
2. http ://perso-laris.univ-angers.fr/∼fasquel/teaching_vision.html

- Depuis 2014 : Président du jury de fin d'études de l'option Génie des Systèmes Industriels de l'ISTIA.

1.3 Activités en ingénierie et développement logiciel

L'équipe de l'IRCAD que j'ai intégrée en 2002 étant naissante (équipe restreinte d'environ 4 personnes, composée de chercheurs et ingénieurs). L'objectif de cette équipe, dirigée par Luc Soler, était de développer des méthodes et logiciels pour la chirurgie assistée par ordinateur. Les logiciels étaient à destination des chirurgiens, radiologues, et des manipulateurs en radiologie. Ils devaient, d'une part, aider à mener à bien des projets de recherche et, d'autre part, pouvoir être utilisés en routine clinique.

Ainsi, durant la période 2002 à 2009 en qualité de chargé de recherche à l'IRCAD, une part importante de mon activité a été consacrée au développement de logiciels. A noter que cette activité ne concernait pas seulement la production de code, mais également son industrialisation, avec, par exemple, les problématiques sous-jacentes de l'organisation du développement (e.g. méthodes agiles de type Scrum et XP), de la qualité (e.g. tests unitaires et fonctionnels, certification CMMI et CE). En effet, il s'agissait de développer des outils fiables, utilisables en routine clinique et, à terme, commercialisables.

A noter enfin que ces développements ont été réalisés dans un contexte de travaux de recherche en partie portés par des projets européens auxquels l'IRCAD participait (e.g. Odysseus [3], Passport [4]).

Indépendamment des résultats issus de la recherche, ces années consacrées, en partie, au développement de logiciels, se sont notamment traduites par :

- Le logiciel VR-Render [5] : logiciel multiplateforme gratuit de visualisation d'images médicales, à destination de médecins.
- La fabrique de logiciels fw4spl [6], associée à des travaux de recherche en génie logiciel [ACL12] : il s'agit d'une architecture logicielle permettant de simplifier la conception d'une famille de logiciels autour, dans notre cas, des fonctionnalités de visualisation 3D, de traitement d'images, de réalité augmentée, de planification chirurgicale et enfin de robotique chirurgicale.
- La création de la startup "Visible Patient[7]" en 2013 : l'objectif initial de cette société a été de fournir un service de reconstruction 3D en ligne à destination des chirurgiens. Le socle technique de "Visible Patient" repose essentiellement sur fw4spl [6].

Cette activité d'ingénierie aura été riche en enseignements, du fait de la proximité avec l'utilisateur final des logiciels (le médecin et le manipulateur en radiologie) et de la confrontation aux problématiques liées au développement de logiciels.

Cette expérience a particulièrement influencé l'orientation de mes recherches sur l'utilisation de connaissances a priori structurelles qualitatives et non quantitatives pour aider à l'interprétation d'images (ceci étant détaillé ultérieurement). Plus précisément, pour l'interprétation des images à l'aide d'un logiciel de traitement interactif, les médecins et manipulateurs raisonnent

3. http ://www.eurekanetwork.org/content/e-3184-odysseus
4. http ://www.ircad.fr/fr/la-commission-europeenne-felicite-lircad-pour-le-projet-passport/
5. http ://www.ircad.fr/fr/linstitut/reconnaissance-internationale/
6. http ://fw4spl-org.github.io/
7. https ://www.visiblepatient.com/fr/company/history/

de manière très qualitative et éprouvent de réelles difficultés à directement interagir avec les algorithmes et surtout avec leurs paramètres.

Cette expérience en développement a également fortement influencé mes contributions scientifiques autour de la conception de logiciels, avec les problématiques sous-jacentes de performance et de généricité.

1.4 Activités de recherche

Depuis la soutenance de ma thèse en 2002, mon activité scientifique s'est organisée autour de deux axes, principalement centrés sur le traitement d'images :

- L'axe "méthode" concerne les méthodes de traitement des images, comprenant :
  - un volet théorique : l'exploitation de connaissances a priori structurelles et qualitatives pour l'interprétation d'images et les méthodes à base de graphes.
  - un volet applicatif centré sur l'aide au diagnostic et, plus récemment, les objets connectés en santé.
- L'axe "implémentation", comprenant :
  - un volet "matériel", associé à l'implémentation opto-électronique de méthodes de traitement d'images.
  - un volet "logiciel", associé à la conception de logiciels autour du traitement des images.

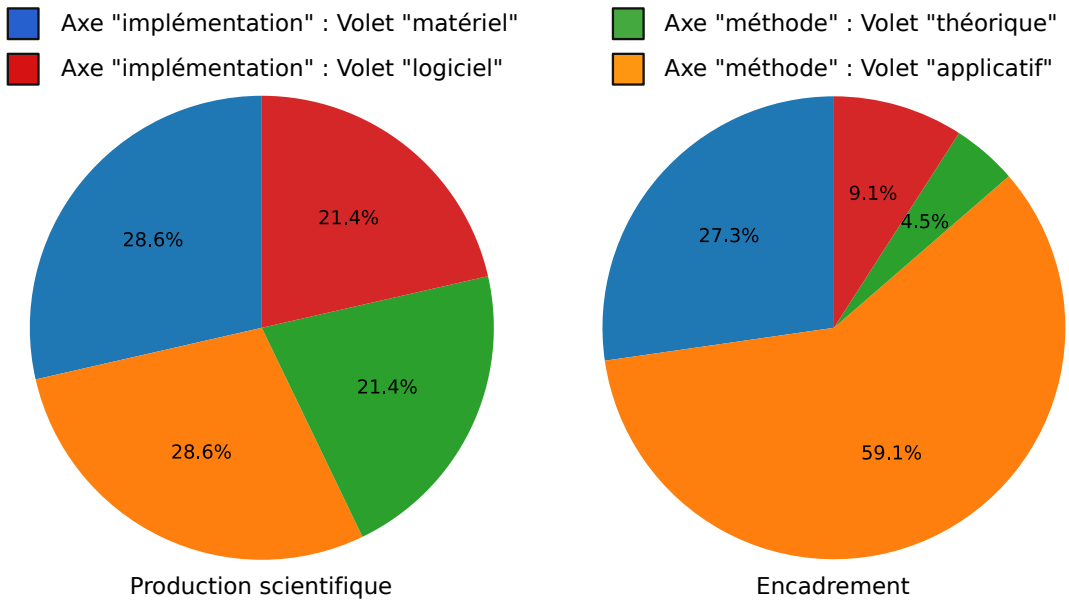

Figure 1.1 : Répartition thématique. Seuls les articles postérieurs à la thèse et publiés dans des revues internationales sont considérés. Les encadrements intègrent les étudiants de master et les étudiants ayant soutenu leur thèse (facteur 6 par rapport aux masters).

La figure 1.1 donne une vue de la répartition thématique de mes activités, en matière de production scientifique (détaillée dans la section 1.7) et d'expérience d'encadrement (détaillée dans la section 1.6).

Après quelques considérations chronologiques (résumées par la figure 1.2), je développe, de manière synthétique, mes contributions sur chacun de ces 2 axes.

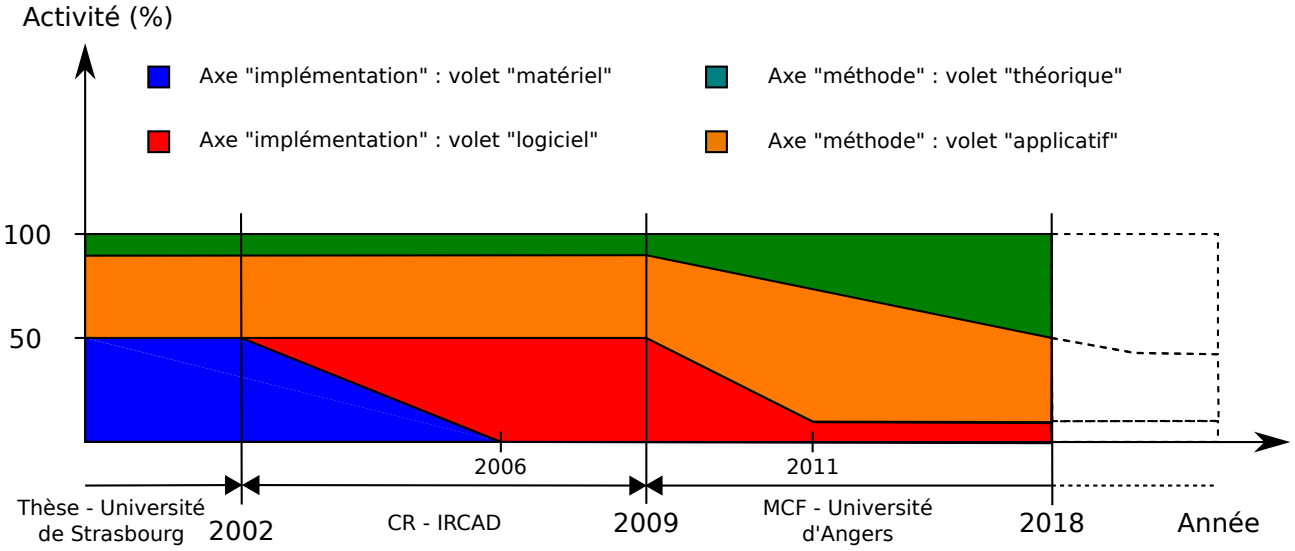

Figure 1.2 : Activité de recherche : considérations chronologiques.

1.4.1 Considérations chronologiques

Avant de détailler les différentes contributions, il est important de souligner les aspects chronologiques de cette activité, depuis 2002 (figure 1.2).

Selon l'axe "implémentation", le volet "matériel" m'a essentiellement mobilisé de 2002 à 2006, dans la continuité d'une partie de mes travaux de thèse [ACL7, ACL8]. Celui-ci a été porté par un projet de recherche collaboratif que j'ai initié en fin de thèse entre plusieurs organismes, dont l'IRCAD, dans le cadre d'une thèse CIFRE qui s'est déroulée de 2003 à 2006. Mon activité sur ce volet a pris fin en 2006, afin de privilégier le volet "logiciel". Ce volet "logiciel" a débuté avec mon recrutement en 2002 à l'IRCAD en qualité de chargé de recherches. Il s'agit d'une problématique centrale de l'IRCAD, l'enjeu étant de concevoir et développer des logiciels rapidement utilisables en routine clinique. J'ai été particulièrement actif sur ce sujet jusqu'à mon recrutement en qualité de Maître de Conférences à l'Université d'Angers en 2009. Depuis la publication de mon dernier article dans une revue internationale dédiée au génie logiciel en 2011, j'ai considérablement réduit mon activité sur ce sujet, en cohérence avec les activités de recherche du laboratoire LARIS au sein duquel j'effectue ma recherche depuis 2009.

L'axe "méthode" concerne le développement de méthodes de traitement pour l'analyse des images. Cet axe est omniprésent depuis 2002, dans la continuité d'une partie de mes travaux de thèse, antérieurs à 2002. De 2002 à 2009, j'ai partagé mon activité de recherche sur cet axe avec l'axe "implémentation". Depuis 2009, et mon recrutement à l'Université d'Angers, cet axe concentre l'essentiel de mon activité de recherche, et correspond aux travaux que je compte privilégier à l'avenir.

Ces différents axes et la chronologie associée sont intimement liés à mon sujet de thèse et aux différents organismes au sein desquels j'ai travaillé (Université de Strasbourg, puis l'IRCAD et

enfin l'Université d'Angers). J'ai ainsi dû adapter mon activité aux différents contextes, propres à chacun de ces organismes. Malgré les efforts sous-jacents d'adaptation, assimilables à des reconversions thématiques, ceci m'a permis de découvrir différentes problématiques, aussi riches que variées, auxquelles je me suis efforcé de contribuer.

1.4.2 Axe "implémentation"

1.4.2.1 Volet "matériel"

Ce volet concerne essentiellement la période de 2002 à 2006, suite à ma thèse couplant méthodes de traitement d'images et implémentation sur des architectures matérielles opto-électroniques. Durant cette période, mon activité sur ce sujet a été portée par une collaboration que j'ai initiée entre l'Université de Strasbourg, le laboratoire ICube (anciennement Phase et InESS), l'IRCAD et l'entreprise MicroModule [8]. Cette collaboration s'est traduite par l'obtention du financement de la thèse de Morgan Madec [T1] (thèse CIFRE), que j'ai co-encadré de 2003 à 2006.

Cette thèse a porté sur une architecture opto-électronique pour la reconstruction rapide de données tomographiques (typiquement les images médicales tomodensitométriques). La contribution principale a concerné l'implémentation de la transformée de Radon, au cœur du processus de reconstruction, à l'aide de dispositifs de traitement couplant de l'optique et de l'électronique. L'objectif était ainsi d'augmenter la puissance de calcul afin d'accélérer la génération des images médicales tomodensitométriques (TDM), en essayant de profiter du potentiel du traitement par voie optique.

Le verrou scientifique portait sur la décomposition de l'algorithme de reconstruction et la conception de l'architecture matérielle de traitement, en prenant en compte différentes contraintes. Les deux premières contraintes étaient le bruit induit par la partie analogique du système (composants optiques, interfaces opto-électroniques) et la dynamique limitée des interfaces opto-électroniques (précision des calculs et perte d'information). Il fallait donc étudier et quantifier l'impact de ces contraintes sur la qualité des images médicales 3D finalement générées. La dernière contrainte concernait la décomposition de l'algorithme sur les différents éléments constitutifs du système (éléments numériques, optiques et opto-électroniques), afin d'optimiser les temps de calcul (e.g. les interfaces opto-électroniques représentant généralement le principal goulot d'étranglement). Ces travaux ont conduit à la publication de 4 articles dans des revues internationales [ACL13, ACL14, ACL15, ACL16], à différentes communications internationales [ACTI9, ACTI10] et à un brevet [BREVET1].

1.4.2.2 Volet "logiciel"

Dès mon recrutement par l'IRCAD en 2002, j'ai commencé à m'intéresser à la problématique de la conception de logiciels, dans le contexte du traitement d'images médicales. L'enjeu était de gérer la variété, autour du traitement d'images, de fonctionnalités complémentaires telles que la gestion des données médicales, les interfaces graphiques, la visualisation 3D, la réalité augmentée et enfin l'interopérabilité avec les systèmes robotisés.

8. http ://www.micromodule.fr/

Dans ce contexte, la difficulté était d'allier performance, généricité et modularité des algorithmes et composants logiciels. Cette période a été l'occasion de proposer des modèles de conception, qui ont fait l'objet de plusieurs publications, autour de trois points :

- L'extension d'un patron de conception [ACTI1], [ACL4] ("design pattern") pour contraindre le traitement des images au sein de régions d'intérêt, de manière complètement transparente pour une implémentation donnée d'un algorithme.
- L'architecture d'un logiciel de modélisation de patients [ACTI3],[ACL6], afin de gérer conjointement et de manière modulaire, les algorithmes de traitement d'images, les fonctionnalités de visualisation 3D, les interfaces graphiques et enfin les données des patients.
- L'architecture d'une série de logiciels ("fabrique de logiciels"), ayant pour objectif de simplifier la génération de logiciels de traitement d'images, de réalité augmentée, de visualisation d'images médicales, de supervision/contrôle de systèmes robotisés d'aide au geste chirurgical. Cette architecture repose sur une proposition de modèle de conception couplant les concepts de "rôles" et de composants [ACTI4, ACTI5],[ACL12].

Le dernier article sur ce sujet a été publié dans un journal international de la communauté en génie logiciel en 2011 [ACL12]. Même si je me concentre, depuis 2009, sur les méthodes et algorithmes de traitement, je continue à apporter une attention particulière à l'implémentation logicielle, en essayant de partager cette expérience transversale entre méthodes de traitement et implémentation logicielle.

On peut par exemple citer une communication récente, en collaboration avec Christian Jean-Guillaume du LARIS, au colloque national de la communauté signal-image Gretsi [ACTN3]. Il s'agissait de présenter l'architecture de l'écosystème des librairies scientifiques autour du langage "Python", en se focalisant principalement sur le calcul scientifique en traitement du signal et de l'image, ainsi que sur quelques fonctionnalités connexes.

On peut également mentionner la dernière communication internationale sur les outils logiciels dans l'enseignement du contrôle, avec Alban Vergnaud et Laurent Autrique [ACTI16].

Enfin, plus récemment, je me suis intéressé à la problématique de l'architecture logicielle dans le domaine des objets connectés appliqués à la santé. Ceci s'est traduit par une contribution sur ce sujet (encadrement d'un étudiant de master recherche), ayant fait l'objet de deux communications à des conférences internationales [ACTI7, ACTI8]. Ce travail a porté sur la conception et la mise en œuvre de l'architecture logicielle et matérielle d'un système d'agrégation de données physiologiques hétérogènes. Ce travail a été réalisé en collaboration avec Mehdi Lhommeau du LARIS, le CHU d'Angers, l'Institut de Cancérologie de l'Ouest (ICO) et l'entreprise Bioparhom. A noter que cette activité autour des objets connectés appliqués à la santé se poursuit actuellement, sous un angle applicatif, selon l'axe « méthode » présenté ci-après.

1.4.3 Axe "méthode"

Mon activité sur ce sujet a été omniprésente depuis 2002, et peut se décomposer en deux volets. Le premier volet est de nature théorique, et a pour vocation de proposer des méthodes d'analyse génériques, c'est-à-dire détachées d'une application particulière. Le second volet est appliqué et vise à répondre à des problèmes spécifiques, essentiellement en rapport avec l'imagerie médicale et, plus généralement, avec la santé.

1.4.3.1 Volet théorique

Ce premier volet concerne l'exploitation de connaissances a priori structurelles et de nature qualitative, pour aider à l'interprétation du contenu des images (e.g. segmentation et reconnais-

sance simultanée de régions). Il s'agit d'un axe auquel je m'intéresse depuis plusieurs années, et que je fais cohabiter avec des travaux plus appliqués et avec l'axe "implémentation" précédent pour la période antérieure à 2009.

Ce type d'approche, supervisée, vise, dans le cas qui m'intéresse, à se focaliser sur des informations a priori qualitatives, que l'on suppose plus simple à formuler et à manipuler (e.g. par un "non expert") que les informations issues d'approches "quantitatives" (e.g. modèle a priori statistique). Il s'agit d'informations telles que par exemple "les tumeurs hépatiques sont incluses dans le foie", "les tumeurs hypodenses sont plus sombres que les vaisseaux". Mes contributions sur ce sujet reposent sur l'utilisation et l'adaptation de méthodes à base de graphes, où les régions à reconnaître dans une image représentent les nœuds, et les informations structurelles qualitatives sont modélisées par des arêtes orientées et non valuées.

En collaboration avec Vincent Agnus (chargé de recherche à l'IRCAD), ma première contribution a concerné une méthode de traitement d'images exploitant les relations d'intersection afin de faciliter la segmentation séquentielle d'une image. A noter qu'une procédure de segmentation séquentielle d'une image consiste à segmenter les régions les unes après les autres, chacune d'elles l'étant avec un (ou plusieurs) algorithme(s) spécifique(s). Par exemple, dans le cas d'images médicales tomodensitométriques de l'abdomen, cela peut consister à tout d'abord segmenter la peau, puis les poumons, et enfin le foie et ses éléments constitutifs. L'idée a été de déterminer la région d'intérêt au sein de laquelle les calculs doivent être effectués, afin de réduire les temps de calcul et d'éliminer les données "polluantes" lors de la segmentation d'une entité donnée. Cette méthode a fait l'objet de plusieurs publications internationales [ACL5],[ACTI2]. J'ai étendu ces travaux avec l'intégration d'un second type de graphe, embarquant les informations photométriques (relations d'ordre entre les intensités moyennes des régions), toujours dans le cadre d'un processus de segmentation séquentiel. L'objectif était alors de combiner ces deux types d'informations qualitatives, afin de guider la segmentation (région d'intérêt et contraintes sur les paramètres de l'algorithme de segmentation). Cette information qualitative a priori est également utilisée pour identifier les régions segmentées par appariement exact de graphes. Ce travail a débuté par l'encadrement d'un étudiant en master recherche. Il a ensuite fait l'objet d'une communication au colloque GRETSI [ACTN2], avant d'être récemment repris pour conduire à une publication dans une revue internationale [ACL9].

Plus récemment, en collaboration avec Nicolas Delanoue du LARIS, j'ai développé une méthode d'interprétation (segmentation et identification des régions) du contenu sur la base d'une sur-segmentation initiale, intégrant les relations qualitatives, a priori connues entre les régions, d'inclusion et de photométrie. Cette approche a été formalisée sous forme d'un problème d'appariement inexacte de graphes. La contribution porte sur la mise en correspondance entre les nœuds des graphes du modèle (connaissances a priori) et les nœuds des graphes construits automatiquement à partir de l'image réelle sur-segmentée. Ce dernier travail a été récemment accepté pour publication dans la revue internationale [ACL10]. Le programme informatique associé est mis librement à disposition de la communauté [L3], ainsi que les données expérimentales utilisées pour évaluer cette méthode[9,10].

1.4.3.2 Volet applicatif

Le second volet tourné vers l'application, constitue une part importante de mon activité. Ceci concerne des méthodes de traitement spécifiques, ayant pour but de résoudre des problèmes liés

9. https ://github.com/jbfasquel/skgtimage
10. http ://perso-laris.univ-angers.fr/∼fasquel/scikit-gtimage.html

à des applications spécifiques, souvent médicales. Ce type d'activité se situe dans un cadre de recherche collaborative très pluridisciplinaire (voir les collaborations résumées dans la section 1.5).

On peut citer l'exemple de travaux récents, en collaboration avec le CHU d'Angers, dédiés à la quantification automatisée de flux sanguins dans la carotide à partir d'images IRM. La méthode développée a consisté en l'adaptation, pour cette application, de contours actifs orientés "régions", et en l'évaluation expérimentale de son efficacité. Ces travaux ont été réalisés à l'occasion du co-encadrement de la thèse de Guillaume Trébuchet [T2], en collaboration avec Serge Willoteau (CHU d'Angers) et Christine Cavaro-Ménard (LARIS), et ont conduit à plusieurs publications [ACL11],[ACTI14, ACTI15].

Certaines de mes contributions concernent des actions ponctuelles sur des travaux portés par d'autres collègues. Ainsi, durant la période 2002 à 2009, on peut citer les publications intégrant les différents travaux de l'équipe autour des applications médicales [OS2], [ACTI11, ACTI12, ACTI13]. Depuis mon arrivée à Angers, on peut citer le cas de la résonance stochastique avec David Rousseau et François Chapeau-Blondeau, du LARIS [ACL3] [ACTN1], ou la problématique de la caractérisation de la fibrose hépatique avec Paul Calès, du CHU d'Angers [ACL1, ACL2] (articles dans la communauté médicale).

Dans ce contexte applicatif, je m'intéresse depuis peu à l'usage des objets connectés en santé. Hormis les travaux mentionnés précédemment sur l'axe "logiciel" [ACTI8, ACTI7], je me suis récemment intéressé au suivi, au moyen d'un tensiomètre connecté, de patient traités pour un cancer. Ce travail a été réalisé en collaboration avec Samir Henni du CHU d'Angers et Anne Humeau-Heurtier du LARIS, lors de l'encadrement du Master recherche d'Antoine Jamin. L'objectif a été d'identifier une méthode permettant, à terme, de prédire l'évolution/l'efficacité d'une chimiothérapie en suivant quotidiennement et à distance l'évolution de leur tension. Ce travail a fait l'objet d'une publication à une conférence internationale [ACTI6]. Plus récemment, j'ai récemment établi une collaboration avec la société Axelife qui développe un dispositif médical connecté (bracelet). Sur la base de cette collaboration, la thèse de Franck Mouney, que je co-encadre avec deux collègues du LARIS, vient de débuter en partenariat avec cette entreprise. L'objectif est d'analyser les données récoltées avec le bracelet connecté afin de prévenir les risques d'accident cardiovasculaire, en employant des méthodes de caractérisation et de classification de données physiologiques hétérogènes (e.g. onde de pouls, température et déplacements (accéléromètre)).

1.5 Rayonnement et animation scientifique

Outre l'exercice régulier de relecture d'articles scientifiques, je résume ci-après quelques-unes de mes activités, relevant du rayonnement et de l'animation scientifique (quelques coopérations, jurys de thèse, invitations à des séminaires et participation à l'organisation de colloques).

Coopérations nationales attestées par des publications :

- CHU de Strasbourg : cela concerne la plupart des travaux publiés durant la période 2002-2009 (médecins membres de l'IRCAD et du CHU de Strasbourg)
- CHU d'Angers : Pr. S. Willoteaux [ACL11], [ACTI14, ACTI15] (département de radiologie), Pr. P. Calès [ACL1, ACL2] (service hépato gastro entérologie et oncologie digestive), Pr. G. Leftheriotis (service explorations fonctionnelles vasculaires) [ACTI6, ACTI7, ACTI8], Pr. P. Abraham (service explorations fonctionnelles vasculaires) [ACTI6] et Dr.

S. Henni (service explorations fonctionnelles vasculaires) [ACTI6, ACTI7, ACTI8].
- Institut de cancérologie de l'Ouest (pôle d'Angers) : Dr. Sophie Abadie-Lacourtoisie [ACTI8]
- Equipe Automatique Vision et Robotique du Laboratoire ICube (UMR CNRS) de Strasbourg : Pr. M. De Mathelin [ACTI4]
- Systèmes et Microsystèmes Hétérogènes du Laboratoire ICube (UMR CNRS) de Strasbourg : Pr. W. Uhring et Y. Hervé [ACL13, ACL14, ACL15, ACL16], [ACTI4, ACTI13] et [BREVET1].
- Equipe Instrumentation et Procédés Photoniques du Laboratoire ICube (UMR CNRS) de Strasbourg : P. Twardowski, Pr. J. Fontaine [ACTI11].
- Laboratoire MIPS (EA 2332) de l'Université de Haute Alsace : E. Hueber [ACL16], [ACTI9]
- Entreprise Bioparhom (bio-impédancemétrie) : [ACTI8]
- Entreprise Micromodule (systèmes opto-électroniques) : [ACL13, ACL14, ACL15] et [ACTI10].

Jury de thèse : Examinateur de la thèse de Shadi Khawandi, soutenue le 21 Décembre 2012, et intitulée : "Interface Homme-Machine pour la surveillance automatisée des personnes âgées."
- Thèse délivrée par l'Université d'Angers
- Jury : P. Chauvet (directeur de thèse), B. Daya (co-encadrant ; IUT-Liban), M. Hajjar (rapporteur), M. Khalil (rapporteur), F. Jouen (examinateur), J.B. Fasquel (examinateur), B. Noury (examinatrice)

Invitation à des séminaires :
- 2018 : J.-B. Fasquel, "L'IOT en Santé". Séminaire invité au "Hacking Health Camp 2018 - Innovations en Santé" organisé à Strasbourg [11], 23 mars 2018.
- 2015 : J.-B. Fasquel, "Python for Medical image processing and visualization", Séminaire invité à l'école polytechnique de Sétubal, Portugal, 20 mai 2015.
- 2010 : J.-B. Fasquel, "Généricité en traitement d'images : niveau algorithmique et logiciel", Séminaire invité au Laboratoire de Recherche et Développement de l'EPITA (Ecole pour l'informatique et les techniques avancées), Paris, 31 mars 2010. [http ://seminaire.lrde.epita.fr/archives.php]

Organisation de conférences nationales :
- 2016 : Membre du comité scientifique de la conférence nationale S2CA (Santé connectée à Angers), qui s'est tenue le 25 Novembre 2016 [12]. Le S2CA est co-organisé par le CHU d'Angers et l'Université d'Angers. Le thème a été "objets connectés : atouts, nuisances, gadgets ?". Mon rôle a principalement concerné la participation à l'établissement du programme.
- 2013 : Membre du comité d'organisation des "4ièmes Journées Démonstrateurs 2013", Angers, Juin 2013.

Organisation de conférences internationales :
- Principal organisateur local, en collaboration avec le "general chair" M. U. Ahmed de l'Université de Mälardalen (Suède), de la conférence internationale "healthyIoT2017" [13],

11. http ://hackinghealth.camp
12. http ://calenda.org/373283
13. http ://healthyiot.org/2017/show/home

qui s'est tenue à Angers les 24-25 Octobre 2017 (50 auteurs, 17 articles retenus, 10 pays, 22 participants). Avec le soutien de collègues du LARIS, mon action a consisté à :

- Gérer et superviser l'organisation locale de la conférence : gérer le budget et les demandes de subvention, superviser la logistique de la préparation de la conférence et son déroulement (sélection et venue du "keynote speaker", salle de conférence, repas, dîner de gala, attestations pour les visas, communication, site web, programme, "session chairs"...).
- Superviser, avec M. U. Ahmed et S. Begum (Université de Mälardalen - Suède), le processus de sélection des articles et d'édition des actes (co-éditeur des actes [OS1]).
- Intégration de "healthyIoT2017" dans une semaine dédiée au numérique à Angers (semaine connectée [14] (21-28 Octobre 2017)), portée par la ville d'Angers à l'occasion de l'accueil du World Electronic Forum. Ma participation a consisté en l'intégration de "HealthIoT" dans cette semaine [15], en tenant compte de différentes contraintes, afin d'en harmoniser le déroulement (communication, lieu, date, ...).

1.6 Encadrement

J'ai eu l'opportunité de co-encadrer deux thèses (une troisième débute actuellement) et d'encadrer dix étudiants en master recherche. J'ai attaché une importance particulière à ce que ces étudiants se forment, dès le master, non seulement à l'exercice de l'étude bibliographique, du raisonnement scientifique et de l'évaluation, mais également à la dissémination des travaux par la communication scientifique. Ainsi, trois étudiants de masters ont eu l'occasion de cosigner des articles pour des conférences internationales.

Encadrement d'étudiants en thèse :

[T3] Février 2018-... Thèse de Franck Mounier (co-encadrement à 30%) avec Mickaël Dinomais (directeur de thèse – LARIS et CHU d'Angers) et Téodor Tiplica (co-encadrement à 30%). Titre provisoire : "Bracelet connecté et détection précoce d'accidents cardiovasculaires". Thèse en collaboration avec l'entreprise AxeLife.

[T2] 2010-2013 Thèse de Guillaume Trébuchet co-encadrée (50%) avec C. Cavaro-Ménard et S. Willoteaux (directeur de thèse – CHU d'Angers). Titre : "Segmentation par contours actifs de séquences de vélocimétrie IRM : Application aux artères carotides".

- Thèse délivrée par l'Université d'Angers.
- Rapporteurs : Pr. Su Ruan (Université de Rouen), Pr. Amine Nait-Ali (Université Paris-Est Créteil), Pr. Jean-Michel Serfaty (PUPH au CHU de Nantes).
- Publications associées : [ACL11], [ACTI7, ACTI6].

[T1] 2003-2006 Thèse de Morgan Madec, co-encadrée (40%) avec W. Uhring et Y. Hervé (Directeur de thèse). Titre : "Conception, simulation et réalisation d'un processeur optoélectronique pour la reconstruction d'images médicales".

- Thèse délivrée par l'Université de Strasbourg.
- Rapporteurs : Pr. Fabrice Heitz (Université de Strasbourg), Pr. Pierre Ambs (Université de Haute-Alsace) et Pr. Olivier Bonnaud (Université de Rennes I).
- Morgan Madec est enseignant chercheur à l'Université de Strasbourg.

14. http ://connectedweek-angers.fr/
15. http ://connectedweek-angers.fr/conference-healthy-iot

- Publications associées : [ACL15, ACL13, ACL14, ACL16], [ACTI4, ACTI13].
- Brevet : [BREVET1].

Encadrement d'étudiants de master recherche :

[M10] 2018 Master recherche de Colin Gachet (encadrement à 100%), sur une application de vision dans le domaine agricole.

[M9] 2018 Master recherche de Martin Schmidt (encadrement à 100%), sur le thème des objets connectés en santé.

[M8] 2017 Master recherche d'Antoine Jamin (encadrement à 100%), sur le thème des objets connectés en santé.

- Antoine Jamin a débuté une thèse CIFRE fin 2017.
- Publications associées : [ACTI8, ACTI7, ACTI6].

[M7] 2011 Master recherche de Christophe Rigaud (encadrement à 100%), sur le thème des connaissances qualitatives pour l'interprétation d'images.

- Christophe Rigaud a effectué une thèse (2011-2014) au L3i de La Rochelle.

[M6] 2009 Master recherche de Guillaume Chabre (encadrement à 100%), sur le thème de la conception de logiciel pour des applications médicales.

- Publications associées : [ACTI4].

[M5] 2009 Master recherche de Vivianne Leguy (encadrement à 100%), sur le thème du traitement d'images médicales.

[M4] 2008 Master recherche de Julien Waechter (encadrement à 100%), sur le thème de la conception de logiciel pour des applications médicales.

- Publications associées : [ACTI5].

[M3] 2007 Master recherche de Mickael Hoarau (encadrement à 100%), sur le thème du traitement d'images médicales.

[M2] 2006 Master recherche de Stéphan Kiss (encadrement à 100%), sur le thème du traitement d'images médicales.

[M1] 2005 Master recherche de Benoît Deville (encadrement à 100%), sur le thème du traitement d'images médicales.

1.7 Production scientifique

Ma production scientifique est résumée par le tableau 1.1 ci-dessous (les deux articles publiés dans des revues internationales et portant sur ma thèse sont distingués du reste). En ne considérant que les articles dans des revues internationales postérieurs à ma thèse, 14 ont été publiés sur la période 2002-2018 (environ 3.5 par intervalle de 4 ans). Je me suis attaché à régulièrement contribuer en premier auteur : 7 articles sur 14 sur la période (soit environ un article en premier auteur tous les 2 ans).

Table 1.1 Synthèse de la production scientifique.

Nature de la production	Nombre
Chapitre d'ouvrage - Actes de congrès	2
Brevet	1
Articles dans des revues internationales (période post-doctorale)	14
Articles dans des revues internationales (période doctorale)	2
Articles dans des conférences internationales	16
Articles dans des conférences nationales	13
Logiciels	3

Logiciels

[L3] Auteur de la librairie "scitkit-gtimage" [16,17]. Cette librairie est associée à la publication [ACL10].

[L2] Co-auteur du logiciel VR-Render (freeware multi-plateforme pour la visualisation d'images médicales), ayant remporté le prix du biomedical visualization contest organisé à la conférence MICCAI de 2008. Logiciel basé sur [L1].

[L1] Co-auteur de la "fabrique de logiciels fw4spl" [18]. L'architecture et les concepts associés ont fait l'objet d'une publication dans une revue internationale [ACL12].

Ouvrages et chapitres d'ouvrage

[OS1] M. U. Ahmed, S. Begum, and J.-B. Fasquel, editors. *Internet of Things (IoT) Technologies for HealthCare, Proceedings of the 4th International Conference, HealthyIoT 2017, Angers, France, October 24-25, 2017*. Springer, 2018.

[OS2] L. Soler, S. Nicolau, A. Hostettler, J.-B. Fasquel, V. Agnus, A. Charnoz, J. Moreau, B. Dallemagne, D. Mutter, and J. Marescaux. *Computational Surgery and Dual Training*, part III, ch. I, pages 139–154. Springer, first edition, 2010.

Articles dans des revues internationales

[ACL1] P. Calès, J. Chaigneau, G. Hunault, S. Michalak, C. Cavaro-Menard, J.-B. Fasquel, S. Bertrais, and M.-C. Rousselet. Virtual expert staging by automated morphometry improves diagnosis and prognostic value of liver fibrosis. *Journal of Hepatology*, 60, 2014.

[ACL2] P. Calès, J. Chaigneau, G. Hunault, S. Michalak, C. Cavaro-Menard, J.-B. Fasquel, S. Bertrais, and M.-C. Rousselet. Automated morphometry provides accurate and reproducible virtual staging of liver fibrosis in chronic hepatitis c. *Journal of Pathology Informatics*, 6, 2015.

16. https ://github.com/jbfasquel/skgtimage
17. http ://perso-laris.univ-angers.fr/ fasquel/scikit-gtimage.html
18. https ://github.com/fw4spl-org/fw4spl

[ACL3] A. delahaies, D. Rousseau, J.-B. Fasquel, and F. Chapeau-Blondeau. Local-feature-based similarity measure for stochastic resonance in visual perception of spatially structured images. *Journal of the Optical Society of America A*, 29 :1211–1216, 2012.

[ACL4] J.-B. Fasquel, V. Agnus, and J. Lamy. An efficient and generic extension to ITK to process arbitrary shaped regions of interest. *Computer Methods and Programs in Biomedicine*, 81(1), 2006.

[ACL5] J.-B. Fasquel, V. Agnus, J. Moreau, L. Soler, and J. Marescaux. An interactive medical image segmentation system based on the optimal management of regions of interest using topological medical knowledge. *Computer Methods and Programs in Biomedicine*, 82 :216–230, 2006.

[ACL6] J.-B. Fasquel, G. Brocker, J. Moreau, N. Papier, V. Agnus, C. Koehl, L. Soler, and J. Marescaux. A modular and evolutive component oriented software architecture for patient modeling. *Computer Methods and Programs in Biomedicine*, 83(3) :222 – 233, 2006.

[ACL7] J.-B. Fasquel and M. Bruynooghe. New hybrid opto-electronic method for fast and unsupervised object detection. *Optical Engineering*, 42(11) :3352–3364, 2003.

[ACL8] J.-B. Fasquel and M. Bruynooghe. A hybrid opto-electronic method for fast off-line handwritten signature verification. *Document Analysis and Recognition*, 7(1) :56–68, 2004.

[ACL9] J.-B. Fasquel and N. Delanoue. An approach for sequential image interpretation using a priori binary perceptual topological and photometric knowledge and k-means based segmentation. *Journal of the Optical Society of America A. To appear*, 2018.

[ACL10] J.-B. Fasquel and N. Delanoue. A graph based image interpretation method using a priori qualitative inclusion and photometric relationships. *IEEE Transactions on Pattern Analysis and Machine Intelligence. To appear*, 2018.

[ACL11] J.-B. Fasquel, A. Lécluse, C. Cavaro-Ménard, and S. Willoteaux. A semi-automated method for measuring the evolution of both lumen area and blood flow in carotid from phase contrast MRI. *Computers in Biology And Medicine*, 66, 2015.

[ACL12] J.-B. Fasquel and J. Moreau. A design pattern coupling role and component concepts : Application to medical software. *The Journal of Systems and Software*, 84 :847–863, 2011.

[ACL13] M. Madec, J.-B. Fasquel, W. Uhring, P. Joffre, and Y. Hervé. Optical implementation of the filtered backprojection algorithm. *Optical Engineering*, 46, 2007.

[ACL14] M. Madec, J.-B. Fasquel, W. Uhring, P. Joffre, and Y. Hervé. Optoelectronic implementation of helical cone-beam computed tomography algorithms. *Optical Engineering*, 47, 2008.

[ACL15] M. Madec, W. Uhring, J.-B. Fasquel, P. Joffre, and Y. Hervé. Compatibility of temporal multiplexed spatial light modulator with optical image processing. *Optics Communications*, 275 :27–37, 2007.

[ACL16] M. Madec, W. Uhring, E. Hueber, J.-B. Fasquel, J. Bartinger, and Y. Hervé. Methods for improvement of spatial light modulator image rendering. *Optical Engineering*, 48, 2009.

Communications avec actes dans un congrés international

[ACTI1] J.-B. Fasquel and V. Agnus. Improving genericity and performances of medical systems and softwares based on image analysis. In *The 18th IEEE International Symposium on Computer-Based Medical Systems, Dublin, Ireland*, 2005.

[ACTI2] J.-B. Fasquel, V. Agnus, L. Soler, and J. Marescaux. A hierarchical topological knowledge based image segmentation approach optimizing the use of contextual regions of interest : Illustration for medical image analysis. In *In Proceedings of the IEEE International Conference on Image Processing, Atlanta (USA)*, pages 777–780, 2006.

[ACTI3] J.-B. Fasquel, G. Brocker, J. Moreau, V. Agnus, N. Papier, C. Koehl, L. Soler, and J. Marescaux. A modular and evolutive software for patient modelling using components, design patterns and a formal xml-based component management system. In *In proceedings of the 19th IEEE International Symposium on Computer-Based Medical Systems, Salt Lake City, USA*, 2006.

[ACTI4] J.-B. Fasquel, G. Chabre, P. Zanne, S. Nicolau, V. Agnus, L. Soler, M. De Mathelin, and J. Marescaux. A role-based component architecture for computer assisted interventions : illustration for electromagnetic tracking and robotized motion rejection in flexible endoscopy. In *Workshop on Systems Architecture for Computer Assisted Intervention, the 12th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), London, UK*, 2009.

[ACTI5] J.-B. Fasquel, J. Waechter, S. Nicolau, V. Agnus, L. Soler, and J. Marescaux. A xml based component oriented architecture for image guided surgery : illustration for the video based tracking of a surgical tool. In *Workshop on Systems Architecture for Computer Assisted Intervention, the 11th International Conference on Medical Image Computing and Computer Assisted Intervention, New York, USA*, 2008.

[ACTI6] A. Jamin, J.-B. Fasquel, A. Humeau-Heurtier, P. Abraham, G. Leftheriotis, and S. Henni. Characterization of home-acquired blood pressure time series using multiscale entropy for patients treated against kidney cancer. In *Fourth International Conference on IoT Technologies for HealthCare, Angers, France*, 2017.

[ACTI7] A. Jamin, J.-B. Fasquel, M. Lhommeau, S. Abadie-Lacourtoisie, and G. Leftheriotis S. Henni. An example of homemade box for iot-based in-home health monitoring. In *IFAC, Toulouse, France*, 2017.

[ACTI8] A. Jamin, J.-B. Fasquel, M. Lhommeau, E. Cornet, S. Abadie-Lacourtoisie, S. Henni, and G. Leftheriotis. An aggregation plateform for iot-based healthcare : Illustration for bioimpedancemetry, temperature and fatigue level monitoring. In *Third International Conference on IoT Technologies for HealthCare, Vasteras, Sweden*, 2016.

[ACTI9] M. Madec, E. Hueber, W. Uhring, J.-B. Fasquel, and Y. Hervé. Procedures for slm image quality improvement. In *European Optical Society Annual Meeting, Paris, France*, 2008.

[ACTI10] M. Madec, W. Uhring, J.-B. Fasquel, P. Joffre, and Y. Hervé. Flc-slm dynamic improvement with temporal multiplexing : Application to optical image processing. In *Proc. SPIE, Photonics Europe 2006, Strasbourg, France*, pages 390–399, 2006.

[ACTI11] G. Sittler, P. Twardowski, J.-B. Fasquel, and J. Fontaine. Conceiving a specific holographic combiner for an augmented reality hmd dedicated to surgical applications. In *Proceeding SPIE, Biophotonics and New Therapy Frontiers, Photonics Europe, Strasbourg*, 2006.

[ACTI12] L. Soler, S. Nicolau, J.-B. Fasquel, V. Agnus, A. Charnoz, A. Hostettler, J. Moreau, C. Forest, D. Mutter, and J. Marescaux. Virtual reality and augmented reality applied to laparoscopic and notes procedures. In *Biomedical Imaging : From nano to macro, 5th IEEE International Symposium on Biomedical Imaging (ISBI)*, 2008.

[ACTI13] L. Soler, S. Nicolau, A. Hostettler, J.-B. Fasquel, V. Agnus, A. Charnoz, J. Moreau, B. Dallemagne, D. Mutter, and J. Marescaux. Virtual reality and augmented reality applied to endoscopic and notes procedures. In *IFMBE (Medical Physics and Biomedical Engineering) proceedings (springer ed.)*, 2009.

[ACTI14] G. Trébuchet, J.-B. Fasquel, A. Lecluse, C. Cavaro Ménard, and S. Willoteaux. Region-based active contours for computer-aided analysis of carotid phase contrast MRI. In *The 26th IEEE International Symposium on Computer-Based Medical Systems, Porto, Portugal*, 2013.

[ACTI15] G. Trébuchet, J.-B. Fasquel, C. Cavaro Ménard, and S. Willoteaux. Coupling anatomical and functional information for the computer-aided delineation of phase-contrast MRI images using active contours. In *IPTA 2012 : International Conference on Image Processing Theory, Tools and Applications, Istanbul, Turkey*, 2012.

[ACTI16] A. Vergnaud, J.-B. Fasquel, and L. Autrique. Python-based internet tools in control education. In *IFAC Workshop on Internet Based Control Education IBCE15 Brescia (Italy)*, 2015.

Communications avec actes dans un congrés national (les plus récents)

[ACTN1] A. Delahaies, D. Rousseau, J.-B. Fasquel, and F. Chapeau-Blondeau. Influence de la structure spatiale des images lors de la quantification de la résonance stochastique dans la perception visuelle. In *Colloque Gretsi*, 2011.

[ACTN2] J.-B. Fasquel and N. Delanoue. Exploitation des relations qualitatives topologiques et photométriques a priori pour le seuillage d'images en niveaux de gris par k-means. In *Colloque Gretsi*, 2013.

[ACTN3] J.-B. Fasquel and C. Jeanguillaume. Python en signal-image : éléments sur le langage, les outils, l'architecture, les fonctionnalités et la communauté. In *Colloque Gretsi*, 2015.

Brevets

[BREVET1] M. Madec, J.-B. Fasquel, and W. Uhring. Processeur opto-électronique de reconstruction de données tomographiques. Dépôt de brevet français FR0608361, Septembre 2006.

Chapitre 2

Méthodes d'analyse de données médicales

2.1 Introduction

Ce chapitre porte sur mon activité de recherche dédiée aux méthodes d'analyse de données, concernant principalement le traitement d'images médicales. Ces travaux ont été réalisés en parallèle, notamment sur la période 2002-2011, de mes autres activités de recherche autour de la problématique de l'implémentation.

Ce chapitre comprend deux parties. La première partie (section 2.2), que je qualifie de "théorique", concerne des contributions à caractère générique, qui ont pour vocation de dépasser un cadre applicatif particulier. La seconde partie (section 2.3) concerne des contributions dédiées à des applications particulières.

La première partie (section 2.2) traite de l'usage de connaissances structurelles qualitatives a priori sur le contenu d'une image afin de guider le processus d'analyse de son contenu. Il s'agit d'une problématique à laquelle j'ai commencé à m'intéresser durant ma période d'activité à l'IRCAD. Dans le contexte de la chirurgie guidée par l'image, un point important concerne la segmentation et la modélisation préalable (dite "pré-opératoire") des différentes structures anatomiques et pathologiques du patient à partir de son image médicale. En routine clinique, il s'avère nécessaire que cette segmentation soit réalisée de manière interactive, les méthodes automatiques n'étant souvent pas suffisamment fiables (i.e. nécessité d'au moins pouvoir corriger manuellement les segmentations automatiques). Se pose alors, pour l'utilisateur, le problème du choix des algorithmes et de leurs paramètres, souvent peu intuitifs et très hétérogènes d'un algorithme à l'autre (e.g. modèles déformables, filtrage, sélection de composantes connexes, seuillage,...). L'idée a été de privilégier l'intégration de connaissances a priori plus intuitives et plus simples à exprimer et à manipuler, afin de guider les algorithmes. Typiquement, ceci a concerné des informations structurelles qualitatives telles *"les tumeurs hépatiques sont incluses dans le foie"*, *"le foie et la rate ont des densités analogues"*, *"les tumeurs hépatiques hypodenses sont plus sombres que le parenchyme dans une image tomodensitométrique"*, ... Dans ce contexte, je me suis également intéressé à la procédure de segmentation, sur la base du constat suivant : l'utilisateur segmente les structures souvent séquentiellement, en fonction de leur "complexité" (en commençant par les plus "simples" à segmenter) et de la modélisation attendue (i.e. les structures à segmenter dépendent de l'acte chirurgical en aval). Ainsi, je me suis intéressé à exploiter les connaissances structurelles mentionnées précédemment, exprimées qualitativement, pour guider la procédure

de segmentation. Ces constats ont ainsi été le point de départ de mes contributions sur ce sujet, résumées et contextualisées dans cette première partie.

La seconde partie (section 2.3) du chapitre porte sur mes contributions autour des applications. Un premier volet, associé à la période 2002-2009, concerne la segmentation d'images médicales dans le contexte de la chirurgie abdominale guidée par ordinateur. Le second volet, correspondant à la période 2010-2014, porte sur la quantification du flux sanguin dans les artères carotides, à l'occasion du co-encadrement d'une thèse CIFRE. Enfin, le dernier volet traite de mon activité naissante autour des objets connectés en santé. Ces travaux appliqués pluridisciplinaires ont été l'occasion d'établir de multiples collaborations et d'élargir le spectre de mes connaissances, en abordant des problèmes concrets et variés.

2.2 Connaissances a priori structurelles et qualitatives pour l'interprétation d'images

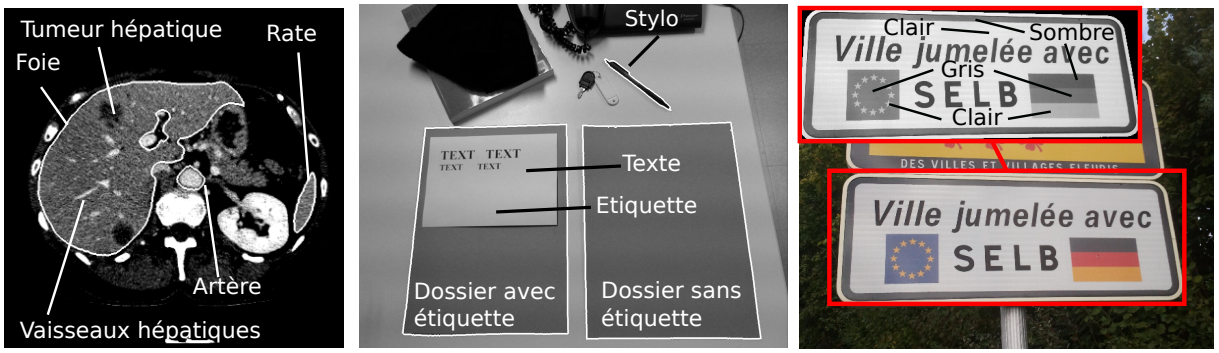

Figure 2.1 : Images dont on peut intuitivement déduire des relations entre les régions. A gauche : image médicale. Les vaisseaux hépatiques et la tumeur sont inclus dans le foie (relation d'inclusion). L'artère présente une densité (intensité moyenne) voisine de celle des vaisseaux hépatiques (relation photométrique). La rate et les tissus du foie présentent également des densités similaires, supérieures à celles des tumeurs hypodenses. Au centre : image en niveaux de gris. De la même manière que pour l'image médicale, on peut aisément établir des relations d'inclusion et photométriques entre les différentes entités indiquées, sur la base de l'observation du contenu. Il en est de même pour les positions relatives (e.g. le dossier sans étiquette est à droite du dossier avec étiquette). A droite : image couleur d'un panneau. Les relations d'inclusion entre les différentes composantes du panneau sont évidentes (e.g. texte inclus sur le fond blanc). On peut également établir des relations entre les régions, en terme des couleurs et les intensités de chacune d'elles (e.g. notion de "plus clair", "plus sombre", "plus coloré" ou "moins coloré").

La première section 2.2.1 introduit quelques éléments autour de la notion de connaissance qualitative et structurelle considérée dans nos travaux pour l'interprétation du contenu des images. A noter que la notion d'interprétation concerne la segmentation des régions composants l'image, ainsi que leur identification. La seconde section 2.2.2 se focalise sur l'exploitation de ces connaissances pour l'interprétation d'images, en présentant quelques éléments autour de l'appariement de graphes. La section suivante 2.2.3 traite de l'usage de ces connaissances dans

le cas d'une approche séquentielle (les régions sont traitées les unes après les autres, dans un ordre donné).

Ces sections présentent le contexte de nos contributions, tandis que les sections suivantes 2.2.4 et 2.2.5 les détaillent davantage.

2.2.1 Connaissances structurelles

En analyse d'images, on s'intéresse souvent à étudier le contenu d'une image, composée de différents objets ou régions. Comme récemment souligné par I. Bloch [21], les informations disponibles dans une image sont les propriétés des objets ou régions, ainsi que les relations entre ces objets ou régions. Ces relations constituent l'information dite structurelle d'une scène donnée. Les relations considérées en analyse d'images sont souvent les relations spatiales (e.g. *l'objet A est à droite de l'objet B*, ou *l'objet A est inclus dans l'objet B*), même si les relations photométriques peuvent également présenter un intérêt [50, 107]. Ceci est illustré par la figure 2.1 avec quelques images que nous avons utilisées dans nos travaux [45, 52, 50, 51].

Un intérêt de ce type de connaissance est qu'elle peut être aisément formulée de manière intuitive par des personnes non expertes en traitement d'images [40]. Une perspective est ainsi de faciliter la mise en œuvre de méthodes d'analyse supervisée, par une construction simplifiée des connaissances a priori (problématique de l'apprentissage), à utiliser par la suite pour guider l'interprétation. Une difficulté est de relier cette connaissance assez abstraite (niveau conceptuel) aux caractéristiques "bas-niveau" des images (niveau perceptuel), telles que par exemple l'intensité des pixels, la forme des régions [39, 70].

Concernant la nature de ces connaissances structures, on peut distinguer deux cas de figure : le cas où les connaissances sont purement qualitatives [39, 52, 51], et le cas où elles sont quantitatives (ou semi-quantitatives) [21]. Dans le cas des approches purement qualitatives, on s'intéresse à des relations non décrites numériquement. On peut, par exemple, citer la notion d'adjacence considérée par A. Deruyver [39], ou les notions d'intersection, d'inclusion et d'intensité relative que nous avons considérées [45, 52, 50, 51]. Les approches dites quantitatives ont notamment pour objectif et pour intérêt de réduire l'écart mentionné précédemment entre la représentation abstraite des connaissances et les informations bas-niveau de l'image [39]. On peut citer l'approche basée sur la logique floue [21], permettant par ailleurs d'intégrer l'imprécision dans les connaissances au moyen des fonctions d'appartenances [20] (e.g. notion de proximité - "proche" ou "loin" - dans une relation de position relative du type "l'objet B est à droite de l'objet A"), tout en préservant la capacité à combiner des informations structurelles hétérogènes (e.g. opérations de conjonction et de disjonction) [21]. La contre-partie de cette approche (semi-)quantitative est la nécessité de régler les paramètres en fonction de l'application considérée. Dans le cas de l'usage de la logique floue, il peut s'agir par exemple des paramètres des fonctions d'appartenance [107, 131].

Nous traitons ci-après de la modélisation de ces connaissances au moyen de graphes et l'interprétation d'images par appariement de graphes.

2.2.2 Interprétation d'images et appariement de graphes : quelques éléments

Nous allons considérer la situation où l'image à interpréter est préalablement sur-segmentée, conduisant à de multiples régions homogènes que l'on va essayer de mettre en correspondance avec les régions déclarées dans le modèle. Il s'agit d'une situation souvent considérée dans les

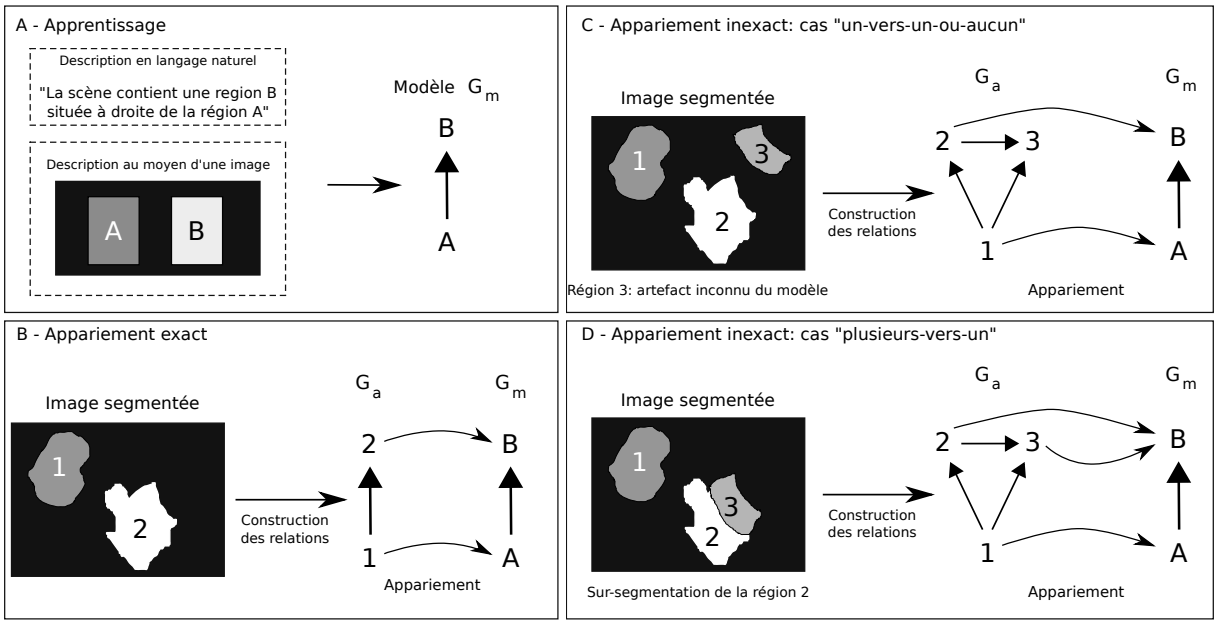

Figure 2.2 : Connaissances a priori structurelles et qualitatives, et appariement de graphes : exemple de la relation de position relative. A : les connaissances a priori sont modélisées au moyen d'un graphe orienté. Ce modèle peut-être construit (apprentissage) par exemple à partir d'une description en langage naturelle ou à partir d'un exemple. B : les deux régions segmentées correspondent exactement aux deux régions du modèle a priori. On peut ainsi aisément appareiller le graphe construit à partir de la segmentation avec celui du modèle. C : l'artéfact (région 3) ne doit être appareillé à aucun nœud (appariement de type "un-vers-un-ou-aucun"). D : la sur-segmentation conduit à deux régions correspondant à une région du modèle. L'appariement recherché, inexact, est de type "plusieurs-vers-un" (2 et 3 associés à B). En combinant les cas C et D, on peut faire face à un appariement inexact de type "plusieurs-vers-un-ou-aucun".

travaux connexes [39, 52, 109]. Cela permet notamment de réduire le nombre d'entités à manipuler, en considérant ces régions homogènes plutôt que les pixels (e.g. notion de "superpixel" [3]).

Sans chercher à être exhaustif, il apparaît que cette problématique d'intégration de ce type de connaissances pour l'interprétation a été formulée de deux manières principales : soit comme un problème de satisfaction de contraintes [39, 40, 131], soit comme un problème d'appariement de graphes [30, 98, 109, 110].

Dans le premier cas de figure, sur lequel je n'ai pas eu l'occasion de travailler mais qui mérite néanmoins d'être mentionné étant donné la production scientifique associée, on raisonne en considérant que les relations entre les régions à reconnaître correspondent à des contraintes qui doivent être satisfaites en fin de procédure. Il s'agit d'un type d'approche relevant de la recherche opérationnelle où les variables sont les régions inconnues. Les domaines associés aux variables correspondent aux identités que peuvent prendre les différentes régions inconnues que l'on cherche à identifier.

L'approche à laquelle je me suis davantage intéressé repose sur la théorie des graphes, et plus précisément sur la problématique d'appariement de graphes, que nous abordons ici.

Les graphes considérés sont le graphe associé à l'image à analyser et celui associé aux connais-

sances a priori. Les nœuds correspondent aux régions et les arêtes embarquent les informations structurelles (i.e. les relations entre les régions) [21]. Ceci est illustré par la figure 2.2-A. L'enjeu est de mettre en correspondance les nœuds des deux graphes, permettant ainsi d'identifier les régions initialement inconnues.

Pour la suite des explications, nous allons considérer qu'une image I est composée d'un ensemble V_m de régions a priori connues $X(u), u \in V_m$. L'indice m désigne la notion de *modèle* a priori. Les connaissances a priori sont modélisées par un graphe $G_m = (V_m, E_m)$, où les nœuds V_m désignent ces régions. Les arêtes E_m modélisent les relations a priori connues entre ces régions (voir l'exemple donné par la figure 2.2).

On considère que l'image à traiter est composée, après sur-segmentation préalable, de régions associées à l'ensemble V_a, à partir de laquelle on établit les relations (de même nature que celles du modèle a priori), conduisant à un graphe $G_a = (V_a, E_a)$. L'indice a désigne l'image à analyser. On peut également considérer des graphes valués, un graphe valué G pouvant être noté $G = (V, E, \alpha, \beta)$, où α et β sont l'ensemble des étiquettes respectivement associées aux nœuds et arêtes [2, 62, 85]. Les étiquettes peuvent correspondre à des vecteurs numériques caractérisant les nœuds et arêtes :

$$\alpha : V \rightarrow L_N \tag{2.1}$$

$$\beta : E \rightarrow L_E \tag{2.2}$$

où $L_N = \mathbb{R}^p$ et $L_E = \mathbb{R}^q$, p, q $\in \mathbb{N} \cup \{0\}$. Dans notre cas, considérer des graphes valués se traduit par $G_a = (V_a, E_a, \alpha_a, \beta_a)$ et $G_m = (V_m, E_m, \alpha_m, \beta_m)$. L_N peut par exemple porter l'information photométrique de la région associée (e.g. intensité moyenne). L_E peut renseigner les relations entre les régions (e.g. contraste, distance).

Comme ceci a été dit précédemment, le problème consiste à déterminer à quel nœud de G_m correspond chacun des nœuds de G_a. Ce problème d'appariement peut être classé en deux catégories principales[34] : un problème d'appariement exact de graphes ou un problème d'appariement inexact.

Nous donnons ci-après quelques détails concernant ces deux catégories, sachant qu'en pratique, notre problématique s'apparente à la seconde.

2.2.2.1 Appariement exact

L'appariement exact de graphes repose sur la notion importante d'isomorphisme de graphes [2, 62, 85]. Un isomorphisme entre G_a en G_m est une fonction bijective $f : V_a \rightarrow V_m$ telle que :

$$\forall (u_i, u_j) \in V_a \times V_a,\ (u_i, u_j) \in E_a \Leftrightarrow (f(u_i), f(u_j)) \in E_m \tag{2.3}$$

Dans le cas de graphes valués, les attributs associés aux nœuds et aux arêtes sont supposés égaux :

$$\forall (u_i) \in V_a,\ \alpha_a(u_i) = \alpha_m(f(u_i)) \tag{2.4}$$

$$\forall e(u_i, u_j) \in E_a,\ \beta_a(e(u_i, u_j)) = \beta_m(e(f(u_i), f(u_j))) \tag{2.5}$$

Ce cas de figure sous-entend que les deux graphes sont de même taille ($|V_a| = |V_m|$). Ceci est illustré par la figure 2.2-B ($f(1) = A$ et $f(2) = B$). Dans le cas de deux graphes de différentes tailles (e.g. sur-segmentation conduisant à $|V_a| > |V_m|$), on pourra s'intéresser à la notion d'isomorphisme de sous-graphe. On cherche alors un isomorphisme entre un sous-graphe $G_a^{'}$ de G_a

et le graphe *modèle* G_m, où $G_a^{'} = (V_a^{'}, E_a^{'}, \alpha_a^{'}, \beta_a^{'})$ est défini par :

$$V_a^{'} \subseteq V_a, E_a^{'} \subseteq E_a \tag{2.6}$$

$$\alpha_a^{'}(v) = \alpha_a(v), \forall v \in V_a^{'}, \beta_a{'}(e) = \beta_a(v), \forall e \in E_a^{'} \tag{2.7}$$

Cette situation d'appariement exact est malheureusement peu, voire pas, rencontrée en pratique, car elle suppose une stricte égalité entre les deux (sous-)graphes. On s'intéresse alors à chercher l'appariement le plus "acceptable", même s'il est imparfait. On parle alors d'appariement inexact de graphe ou d'appariement tolérant à l'erreur (*error-tolerant graph matching*).

2.2.2.2 Appariement inexact

L'appariement inexact peut être formalisé de différentes manières [58], selon que l'on considère ou non la préservation de la structure du graphe, et selon l'importance que l'on donne à la ressemblance entre les attributs portés par les nœuds et les arêtes mis en correspondance dans le cas de graphes valués.

Un point néanmoins important à souligner concerne la nature de l'appariement recherché. Il peut s'agir de rechercher une seule correspondance par nœud, dans le cas d'un appariement de type "un-vers-un" ("one-to-one"), ou plusieurs correspondances par nœud [135]. Dans ce dernier cas, on peut s'intéresser à différents types d'appariement : "plusieurs-vers-un" ("many-to-one"), "un-vers-plusieurs" ("one-to-many") ou plus généralement "plusieurs-vers-plusieurs" ("many-to-many"). Dans la suite, nous présentons succinctement le cas de l'appariement du type "un-vers-un", avant d'aborder celui impliquant plusieurs correspondances par nœud, ce dernier étant plus couramment rencontré en traitement d'images [21, 39, 40, 52, 98, 109].

L'appariement inexact de type "one-to-one" probablement le plus similaire au cas exact est l'isomorphisme de (sous-)graphes tolérant à la substitution (*"substitution-tolerant subgraph isomorphism"* [22]). Ce cas s'applique aux graphes valués et l'appariement est vu comme une substitution des nœuds et des arêtes du graphe G_a par ceux du graphe G_m, guidée par la ressemblance ou similarité entre les attributs, que l'on cherche à maximiser. On retrouve la propriété définie par l'équation 2.3, et seules les relations formulées par les équations 2.4 et 2.5 sont modifiées : $\alpha_a(u_i) \approx \alpha_m(f(u_i))$ et $\beta_a(e(u_i, u_j)) \approx \beta_m(e(f(u_i), f(u_j)))$. Une caractéristique de cette approche est que l'on s'interdit de modifier la structure du graphe.

Il existe cependant beaucoup de travaux permettant de gérer cette modification de structure. Une approche couramment considérée pour aborder ce problème est la notion de distance d'édition ("graph edit distance" ou GED) [58, 62, 85]. Il s'agit d'exprimer la mise en correspondance entre deux graphes au moyen d'opérations permettant de transformer le premier graphe afin d'obtenir le second. Les opérations considérées sont les opérations (potentiellement pondérées) de substitution, d'ajout et d'élimination de nœuds et d'arêtes [12, 58]. Cette situation est illustrée par la figure 2.2-C : l'appariement considère l'élimination du nœud 3. L'enchaînement de ces opérations d'édition est vu comme un chemin d'édition auquel on associe une distance, que l'on cherchera à minimiser. On peut également exprimer cet appariement au moyen d'une fonction de coût que l'on cherchera à minimiser (sans explicitement parler de substitution, d'ajout et d'élimination de nœuds et d'arêtes). On peut citer l'exemple de la formulation considérée par S. Berretti [15], qui suppose des graphes complets (i.e. chaque couple de nœud est relié par une arête). Dans ce cas, on considère l'appariement f minimisant le coût ou l'énergie $E(f)$ (i.e. $f = argmin_f(E(f))$), où E peut être formulée de la manière suivante (notation inspirée de la formulation utilisée par T. Bärecke [12]) :

$$E(f) = \alpha \sum_{v \in V_m} d_v(\alpha_m(v), \alpha_a(f^{-1}(v))) + (1-\alpha) \sum_{(u,v) \in E_m} d_e(\beta_m(u,v), \beta_a(f^{-1}(u), f^{-1}(v))), \tag{2.8}$$

où d_v et d_e matérialisent respectivement les distances (ou les dissimilarités) entre les nœuds et arêtes (à valeur dans [0,1]). Le coefficient α pondère la contribution de l'information portée par les nœuds vis à vis de celle associée aux arêtes (information structurelle). A noter que l'on utilise G_m comme référence (somme sur V_m et E_m) en supposant que chaque élément du modèle est associé à un élément du graphe de plus grande taille G_a (G_m est vu comme un sous-graphe de G_a). En traitement d'images, comme ceci a été souligné par E. Bengoetxea [14], cette manière de formuler le problème pourrait s'appliquer au cas où des objets "inconnus" du modèle sont présents dans l'image (e.g. le fond, s'il n'est pas modélisé) : les objets inconnus ne sont alors appareillés à aucun nœud du modèle (région ou nœud 3 dans la figure 2.2-C).

L'appariement inexact avec plusieurs correspondances par nœud est probablement plus adapté en traitement d'images. En effet, il intègre par exemple la possibilité d'associer plusieurs régions de l'image analysée à une région du modèle (cas rencontré notamment si l'image est sur-segmentée) [14, 135]. Cette situation est illustrée par la figure 2.2-D : l'appariement considère la fusion des nœuds (régions) 2 et 3. Ce cas de figure correspond au cas "plusieurs-vers-un", que l'on pourrait étendre au cas "plusieurs-vers-aucun-ou-un" dans le cas de régions inconnues du modèle (combinaison des situations C et D présentées dans la figure 2.2). A noter que certains travaux s'intéressent au cas plus général d'un appariement de type "plusieurs-vers-plusieurs" [135] que nous n'abordons pas ici. En se limitant au cas de figure "plusieurs-vers-un", avec l'hypothèse que $|V_m| \leq |V_a|$, on peut formuler l'appariement au moyen d'une fonction $f : V_a \rightarrow V_m$, caractérisée par :

$$\forall u_i \in V_a, \exists u_j \in V_m \mid u_j = f(u_i)$$
$$\forall u_j \in V_m, f^{-1}(u_j) = \{u_i \in V_a \mid u_j = f(u_i)\}$$

Cette problématique d'appariement intégrant plusieurs mises en correspondance par nœud a fait l'objet de beaucoup de travaux, avec différentes formulations [8, 23, 31, 123, 135]. Par exemple, R. Ambauen [8] propose une extension de la distance d'édition en intégrant de nouvelles opérations : la fusion de nœuds ("plusieurs-vers-un") et la division de nœuds ("un-vers-plusieurs"). Sans parler de distance d'édition, on peut également formuler la recherche de l'appariement (cas "plusieurs-vers-un") sous forme d'une fonction de coût à minimiser [14, 109] :

$$E(f) = \alpha D_v(f) + (1 - \alpha) D_e(f) \tag{2.9}$$

où α ($\in [0, 1]$) permet de régler le poids donné, pour un appariement f, à la dissimilarité entre les nœuds (D_v) et entre les arêtes (D_e) mis en correspondance. Cette fonction de coût peut être ajustée en fonction des éléments que l'on souhaite prendre en compte pour évaluer la pertinence d'une mise en correspondance [14]. Par exemple, M. C. Boeres [23] propose de tenir compte des nœuds non mis en correspondance bien que très similaires, afin de pénaliser une mise en correspondance qui ignorerait des nœuds et arêtes similaires. Cette fonction de coût proposée par M. C. Boeres [23] s'exprime sous la forme (formulation inspirée de S. Sorlin[123], en considérant des dissimilarités et non des similarités) :

$$D_v(f) = \frac{1}{|V_a|.|V_m|} \left(\sum_{(v_a, v_m) \in V_a \times V_m | v_m = f(v_a)} d_v(v_a, v_m) + \sum_{(v_a, v_m) \in V_a \times V_m | v_m \neq f(v_a)} 1 - d_v(v_a, v_m)) \right) \tag{2.10}$$

$$D_e(f) = \frac{1}{|E_a|.|E_m|} \left(\sum_{(e_a, e_m) \in E_a \times E_m | e_m = f(e_a)} d_e(e_a, e_m) + \sum_{(e_a, e_m) \in E_a \times E_m | e_m \neq f(e_a)} 1 - d_e(e_a, e_m) \right) \tag{2.11}$$

où $\forall e_a = (u_a, v_a) \in E_a,\ f(e_a) = f((u_a, v_a)) = (f(u_a), f(v_a))$. Les fonctions d_v et d_e sont les mesures de dissimilarité relatives aux nœuds et aux arêtes, à valeur dans $[0, 1]$.

Dans le cas de cette formulation (équation 2.9), une difficulté est le temps de calcul induit par le caractère hautement combinatoire de la recherche de la solution optimale (e.g. si l'on évalue toutes les mises en correspondance possibles) [109]. Même s'il s'agit d'un aspect que je n'ai pas abordé au cours de mes travaux, cette problématique de la recherche de la solution optimale est un enjeu majeur en appariement de graphes [30, 119].

Afin de contourner cette difficulté, au moins partiellement, une alternative consiste à tout d'abord chercher une mise en correspondance de type "un-vers-un" à partir de laquelle la relation "plusieurs-vers-un" est ensuite construite [109, 52, 110].

Par exemple, de récents travaux [109] dédiés à la segmentation interactive d'images proposent d'initialiser cet appariement grâce aux pixels sélectionnés manuellement par l'utilisateur (i.e. régions issues d'une sur-segmentation). Les auteurs proposent ensuite d'utiliser la notion de graphe déformable pour obtenir l'appariement final [110]. Le même type d'approche a été considéré, pour une autre application en traitement d'images [109]. Le lecteur pourra consulter ces deux articles pour de plus amples détails [109, 110]. Nous avons récemment proposé une approche (détaillée dans la section 2.2.4) suivant cette stratégie [52], où l'appariement de type "plusieurs-vers-un" est initialisé par un appariement de type "un-vers-un", correspondant à un isomorphisme de sous-graphe.

2.2.3 Interprétation séquentielle guidée par les connaissance structurelles

Une stratégie classique pour la segmentation d'images est de considérer une approche séquentielle (voir l'exemple récent [86] de procédure de segmentation d'une image médicale). Dans ce cas, il s'agit de concevoir une chaîne de traitement composée de différents algorithmes souvent hétérogènes de nature, chacun se focalisant sur une structure ou une région particulière. On choisit ces algorithmes en fonction des caractéristiques des régions (e.g. morphologiques, photométriques et texturales), ceci pouvant parfois s'avérer plus efficace que de considérer un seul algorithme qui aurait pour vocation de segmenter simultanément l'intégralité des régions (e.g. situation précédemment considérée, à partir d'une sur-segmentation initiale). Dans ce cas de figure, la question est de savoir comment exploiter (et pour quels bénéfices) les connaissances a priori que nous considérons ici pour guider ce processus d'interprétation séquentiel. L'idée est de s'appuyer, à une étape donnée du processus, sur les connaissances structurelles pour segmenter une nouvelle région, en s'aidant simultanément de celles déjà segmentées. Ceci peut également s'appuyer, en plus du raisonnement sur les connaissances, sur les outils, mentionnés précédemment, relatifs à l'appariement de graphes, pour identifier les régions obtenues en fin d'étape. Ces propos sont illustrés par la figure 2.3 : le guidage consiste à déterminer la région d'intérêt pour l'étape 2, tandis que l'appariement permet d'identifier les régions restantes à la fin de cette même étape.

Plusieurs travaux ont été réalisés ces dernières années sur ce sujet. Par exemple, O. Camara [28] s'est focalisé sur la segmentation de plusieurs structures thoraciques et abdominales à partir d'images médicales (CT et PET). Dans ce cas, une séquence particulière a été considérée : segmentation de la peau, puis des os, des poumons, des reins et enfin du foie. Cette procédure s'est appuyée sur des opérateurs élémentaires de traitement d'images (e.g. seuillage, morphologie mathématique binaire, analyse des composantes connexes). Les connaissances structurelles spatiales a priori connues ont été exploitées pour identifier la région d'intérêt (ROI) d'une cible donnée, en s'appuyant conjointement sur ces connaissances a priori et sur la connaissance des structures préalablement segmentées et identifiées. Une approche par modèles déformables a

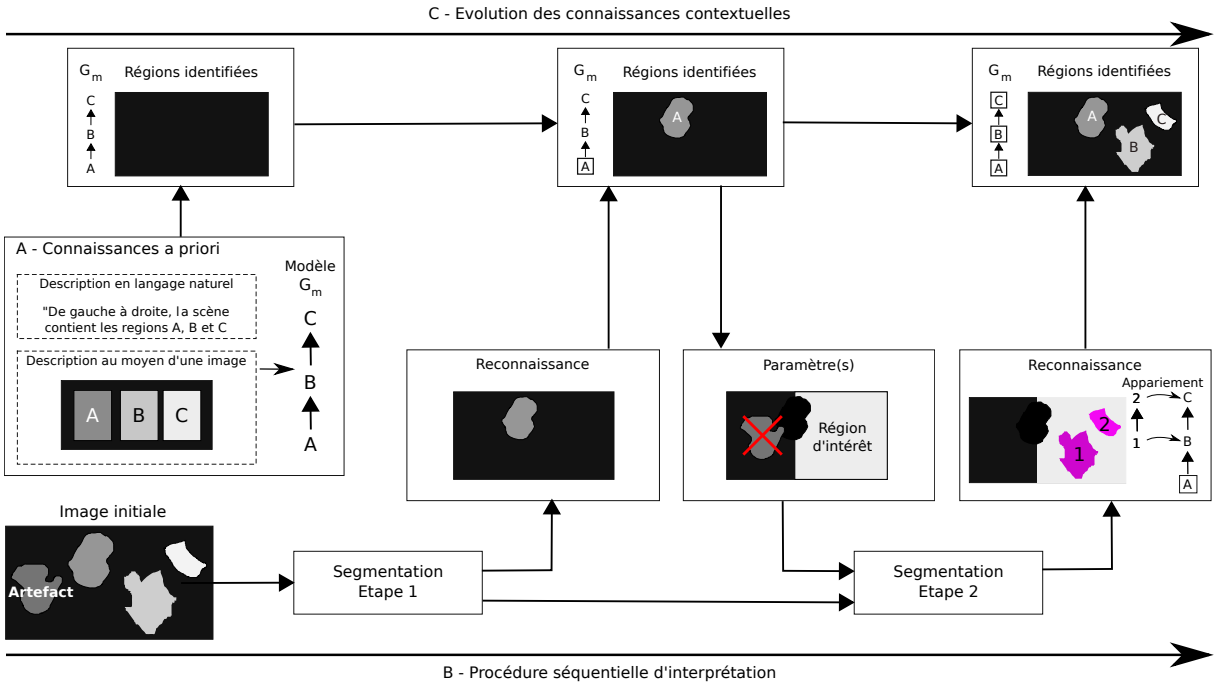

Figure 2.3 : Connaissances a priori structurelles et qualitatives, et interprétation séquentielle : principe par l'exemple. A : les connaissances a priori sont modélisées au moyen d'un graphe orienté (position relative). B : L'image initiale, comprenant un objet inconnu (artefact), est analysée en 2 étapes, chacune conduisant à l'identification de régions (région A, puis régions B et C). C : Evolution des connaissances, initialisées par le modèle a priori, en fonction du contexte ("connaissances contextuelles"). Etape 2 : les connaissances contextuelles permettent de guider la segmentation (*"paramètre(s)"*). Ainsi, pour obtenir les régions B et C sachant que A est identifiée, on contraint l'espace de recherche (région d'intérêt) en s'appuyant sur la connaissance a priori (B et C sont à droite de A). Comme ceci est illustré par l'appariement en fin d'étape 2, les connaissances peuvent être utilisées pour identifier les régions en fin d'étape. L'appariement fait intervenir un sous-graphe de G_m, associé au contexte.

également été intégrée à la procédure, en ajoutant un terme supplémentaire aux forces extérieures, attaché à la ROI (et donc aux connaissances structurelles a priori), afin d'assurer que le modèle reste à l'intérieur de celle-ci lors de la procédure de déformation. De manière similaire, les travaux de O. Colliot [33] ont porté sur l'intégration de relations spatiales floues sous forme de forces extérieures additionnelles pour guider le processus de segmentation à base de modèles déformables, dans le cas de la segmentation de structures cérébrales à partir d'images IRM. Une approche analogue a été proposée pour la segmentation du cœur [105], en s'appuyant sur la segmentation préalable des poumons, et les relations spatiales a priori connues. Dans le même esprit, nous avons proposé une approche permettant de déterminer la région d'intérêt au sein de laquelle une séquence de traitement doit être restreinte. Cette contribution, détaillée dans la section 2.2.5, a été réalisée dans le cadre d'un système interactif de segmentation et l'identification des structures anatomiques et pathologiques. Plus récemment, nous avons proposé une approche séquentielle où l'enjeu est de guider l'algorithme de segmentation par *k-means*, algorithme couramment utilisé en vision, et d'identifier les régions produites par cet algorithme en se basant sur un appariement exact de graphes. Cette contribution est détaillée en fin de section 2.2.5.

Dans ce cas de figure, nous combinons à la fois le guidage d'un algorithme et l'identification par appariement avec le graphe associé aux connaissances structurelles qualitatives a priori.

Il est important de citer les travaux de G. Fouquier [59], où l'accent a été mis sur la problématique de l'ordre dans lequel les structures doivent être segmentées et sur le problème de la propagation de l'erreur de segmentation. La méthode proposée pour guider la séquence repose sur les cartes de saillance combinée à la localisation des régions cibles candidates, par rapport aux structures déjà segmentées. Ainsi, à une étape donnée du processus, la région cible choisie est la plus saillante, supposée plus facile à segmenter. Le problème de la propagation de l'erreur de segmentation est géré en vérifiant la cohérence d'une structure nouvellement segmentée par rapport aux structures préalablement segmentées (sur la base des relations structurelles spatiales a priori connues). Ceci offre la possibilité de revenir en arrière en cas d'incohérence pour ensuite choisir une autre région cible.

2.2.4 Contribution : analyse globale, relations d'inclusion et relations photométriques

Cette contribution concerne l'interprétation de l'image selon une approche globale : on cherche à segmenter et identifier simultanément toutes les régions de la scène, à partir d'un algorithme de sur-segmentation préalable.

Ces travaux reposent sur un appariement de type "plusieurs-vers-un" initialisé par un appariement de type "un-vers-un", correspondant à un isomorphisme de sous-graphe [52]. Notre approche fait intervenir deux informations structurelles (représentées par un modèle à deux graphes) : les relations d'inclusion et les relations photométriques a priori.

En considérant une image composée de $|V_m|$ régions (chacune étant identifiée par une étiquette $n \in V_m$), les relations topologiques d'inclusion (t) et de photométrie (p) sont représentées par les graphes orientés $G_{m,t} = (V_m, F_{m,t})$, $G_{m,p} = (V_m, F_{m,p})$:

$$\forall \{k,l\} \in V_m^2,\ k \xrightarrow{t} l \Leftrightarrow X(k) \subsetneq X(l), \tag{2.12}$$

$$\forall \{k,l\} \in V_m^2,\ k \xrightarrow{p} l \Leftrightarrow \bar{X}(k) < \bar{X}(l), \tag{2.13}$$

où $\bar{X}(.)$ est l'intensité moyenne de la région X de l'image. Cette modélisation est illustrée par la figure 2.4 (*Image 1* et *Image 2* pour les cas simples).

Notre approche intègre la notion de similarité photométrique :

$$\forall \{k,l\} \in V_m^2,\ k \xrightarrow{p} l \Leftrightarrow \bar{X}(k) \simeq \bar{X}(l), \tag{2.14}$$

Cette notion de similarité (donnant lieu à une relation d'ordre incertaine) est illustrée par la figure 2.4-*Image 3*). Par exemple les régions "rouges" sont similaires mais non strictement identiques avec une relation d'ordre qui peut varier d'une pose à l'autre : $\bar{X}(2H) < \bar{X}(1G)$ pour la pose 1, mais $\bar{X}(2H) > \bar{X}(1G)$ pour la pose 6.

L'enjeu est de mettre en correspondance simultanément les deux graphes $G_{a,t}$ et $G_{a,p}$, construits à partir de l'image à analyser (initialement sur-segmentée), avec les deux graphes du modèle $G_{m,t}$ et $G_{m,p}$. La figure 2.6 donne quelques exemples d'images sur-segmentées (les résultats sont commentés en fin de section).

La méthode d'appariement proposée est résumée par la figure 2.5. Comme précisé précédemment, plutôt que de considérer une approche globale intégrant une fonction de coût à optimiser sur un grand nombre d'appariements possibles, nous optons ici pour une méthode en deux étapes (voir figure 2.5-gauche).

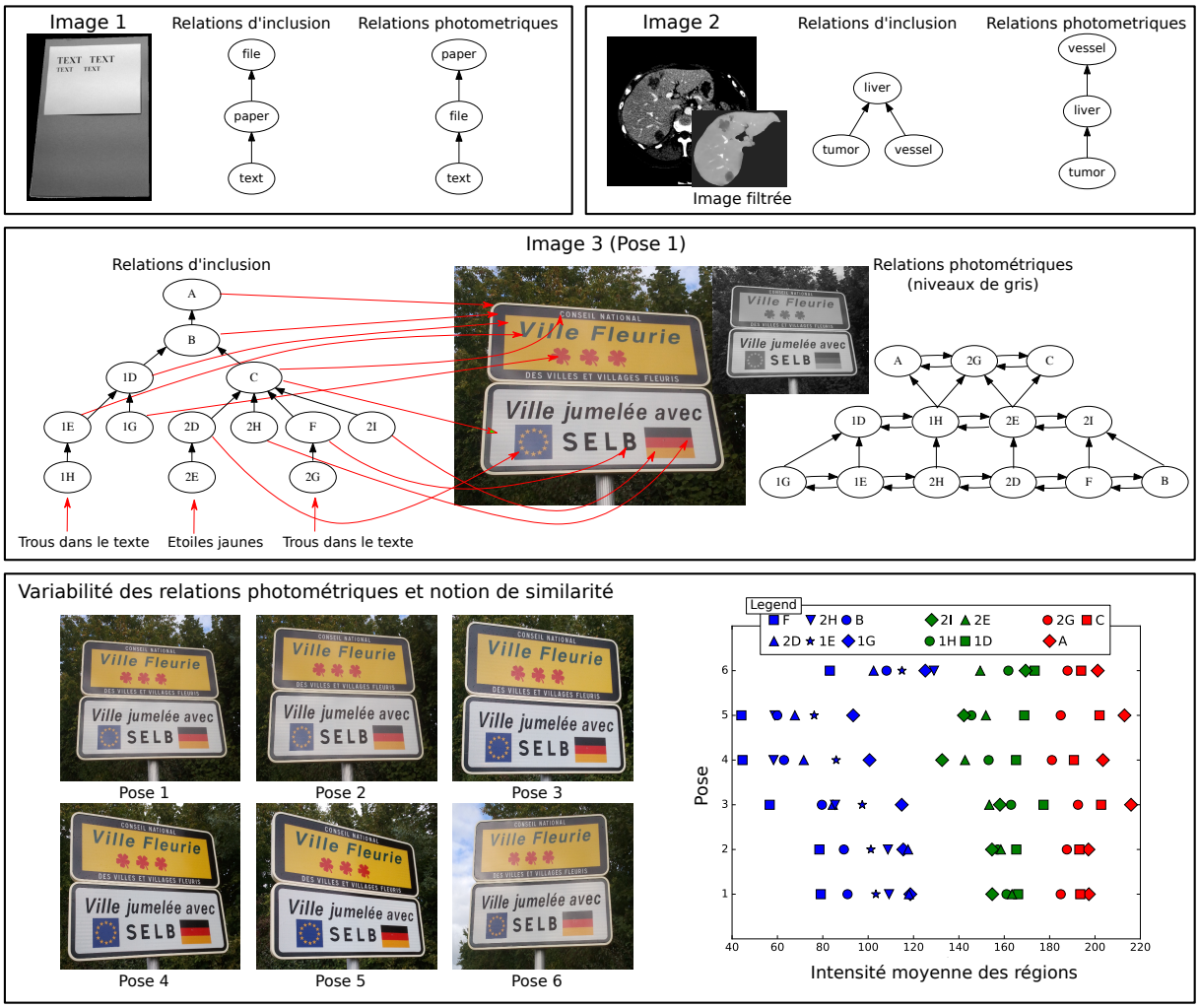

Figure 2.4 : Relations d'inclusion et de photométrie. La notion de similarité (matérialisée par les cycles) permet de gérer des relations d'ordre variables d'une image à l'autre. Cette figure est associée à nos travaux [52].

La première étape consiste à sélectionner un isomorphisme de sous-graphe commun aux deux informations (flèches rouges de la figure 2.5-gauche), parmi l'ensemble H des candidats possibles :

$$H = H_t \bigcap \left(\bigcup_i H_{p^i} \right) \tag{2.15}$$

où $\{H_t\}$ désigne l'ensemble des isomorphismes de sous-graphe entre un sous-graphe de $G_{a,t}$ et le graphe $G_{m,t}$. On peut visuellement déduire que plusieurs mises en correspondance sont possibles. Le terme $\{H_{p^i}\}$ fait intervenir les différents graphes photométriques induits en cas d'incertitude photométrique a priori (voir figure 2.5-droite pour une relation d'incertitude). A noter que l'on considère la fermeture transitive des graphes $G_{a,.}$, ce qui permet davantage d'appariement (i.e. $(i \rightarrow j) \ \wedge \ (j \rightarrow k) \Rightarrow (i \rightarrow k)$). Le meilleur isomorphisme est ensuite retenu. Le critère utilisé, non détaillé ici (voir [52]), cherche à maximiser la dispersion des intensités moyennes des régions du domaine de définition de l'isomorphisme. En cas d'incertitude photométrique, on cherche

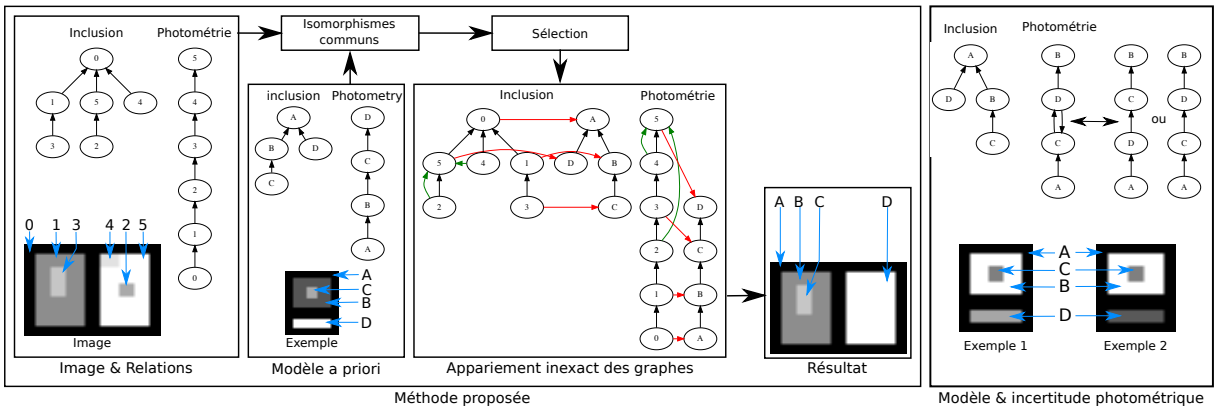

Figure 2.5 : Méthode d'interprétation basée sur de relations d'inclusion et de photométrie, associée à la publication [52]. Gauche : Principe de la méthode proposée. Droite : Modélisation des régions photométriquement similaires (incertitude sur la relation d'ordre).

simultanément à minimiser la dispersion des régions déclarées a priori similaires.

La seconde étape consiste à fusionner les régions restantes (flèches vertes de la figure 2.5-gauche). Les régions sont progressivement fusionnées avec les régions participant à l'isomorphisme retenu, en commençant par les fusions minimisant les différences en intensité moyenne, sous réserve qu'elles ne remettent pas en cause l'isomorphisme (et donc les relations attendues).

La figure 2.6 donne des exemples de résultats obtenus pour différents algorithmes de sur-segmentation initiale (on pourra se référer à l'article publié [52] pour l'évaluation expérimentale complète et détaillée). L'exemple 1 concerne une sur-segmentation initiale au sein d'une région d'intérêt (ROI) obtenue par level-sets. Indépendamment de l'identification des régions, l'appariement permet également de corriger la segmentation initiale. L'exemple 2 concerne une sur-segmentation initiale globale par meanshift suivie d'une fusion de quelques régions connexes (graphe d'adjacence des régions - RAG). Cette fusion est contrôlée en analysant les composantes connexes du graphe d'inclusion. La fusion s'arrête lorsqu'il y au moins une composante connexe de "taille" au moins égale à celle du graphe d'inclusion du modèle et présentant un isomorphisme avec le modèle (la composante finalement retenue étant celle présentant le plus grand nombre d'isomorphismes communs avec le modèle). Notre méthode d'appariement s'applique finalement sur la composante connexe présentant le plus grand nombre d'isomorphismes, permettant ainsi d'éliminer le fond (ou une partie). Les exemples 3a et 3b correspondent à deux résultats obtenus par une approche analogue sur une image présentant plus de régions à identifier (voir figure 2.4 pour le modèle initial). Enfin, le dernier exemple concerne une image médicale sur-segmentée par k-means (traitements contraints au sein du foie préalablement segmenté). Malgré l'imperfection de la sur-segmentation, les régions sont finalement correctement fusionnées et identifiées, en ajustant le modèle par ajout de l'entité "artefact" correspondant à l'effet de volume partiel couramment rencontré en imagerie médicale.

Le programme informatique et les données expérimentales liés à cette contribution sont mis librement à disposition de la communauté [1].

1. http ://perso-laris.univ-angers.fr/∼fasquel/scikit-gtimage.html

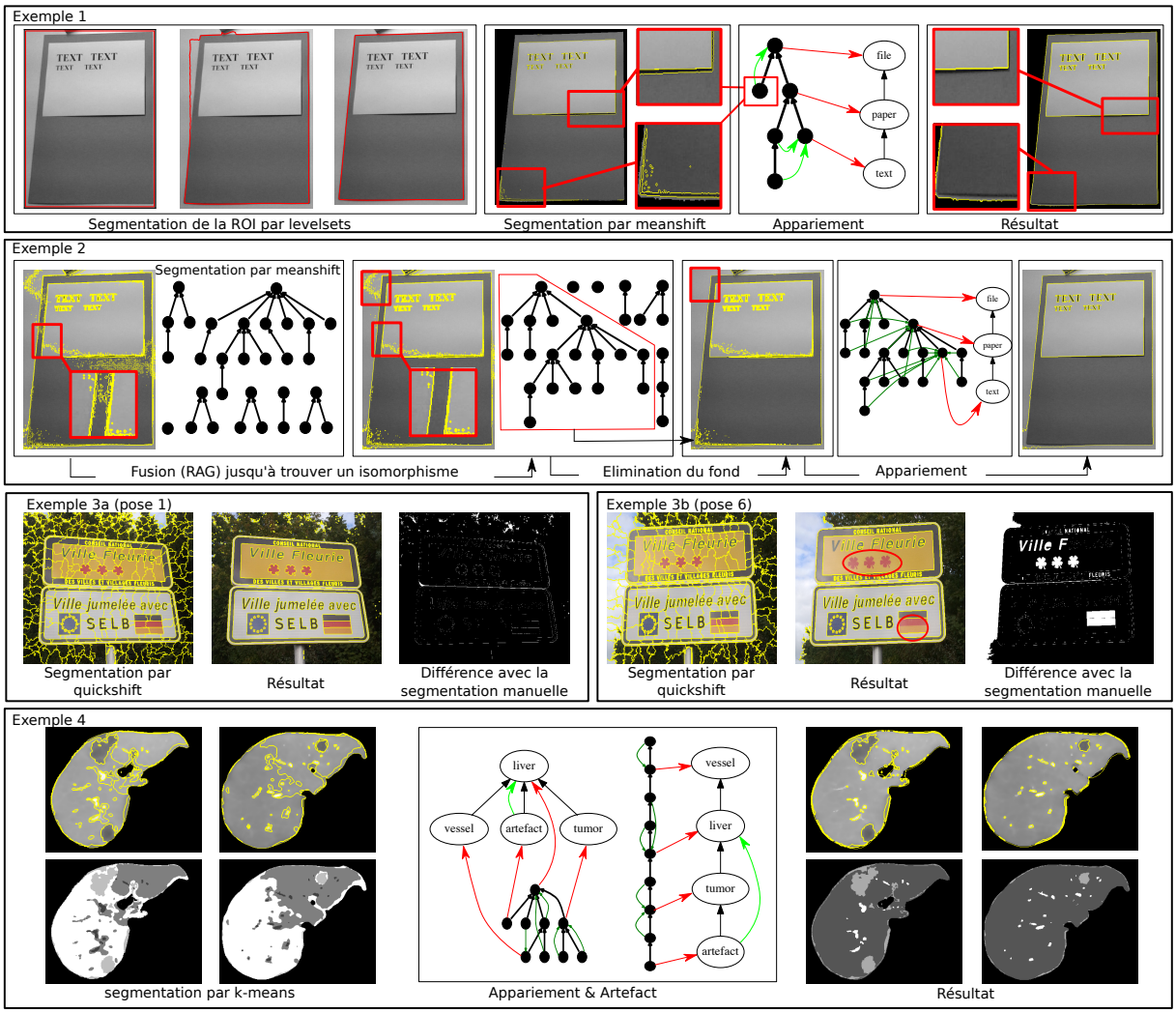

Figure 2.6 : Méthode d'appariement inexact de graphes, dans le cas de relations d'inclusion et de photométrie : exemples de résultats, pour différents algorithmes de sur-segmentation initiale. Cette figure est associée à la publication [52].

2.2.5 Contributions : analyse séquentielle, relations d'intersection, d'inclusion et photométriques

Cette section présente deux contributions à l'interprétation d'images, guidées par des connaissances structurelles qualitatives, dans le cas d'une approche de traitement séquentielle, cet aspect ayant été contextualisé dans la section 2.2.3.

2.2.5.1 Analyse séquentielle, relations d'intersection et région d'intérêt

Dans [45], nous avons proposé une approche analogue à [28], mais dans le cas d'une procédure de segmentation interactive, où l'on ne connaît pas a priori l'enchaînement des segmentations (l'ordre est défini par l'utilisateur). Notre approche vise à déterminer la région d'intérêt à considérer pour segmenter une nouvelle région, en fonction des relations d'intersection a priori connues et des structures segmentées. Les relations d'intersection sont représentées par le graphe orienté

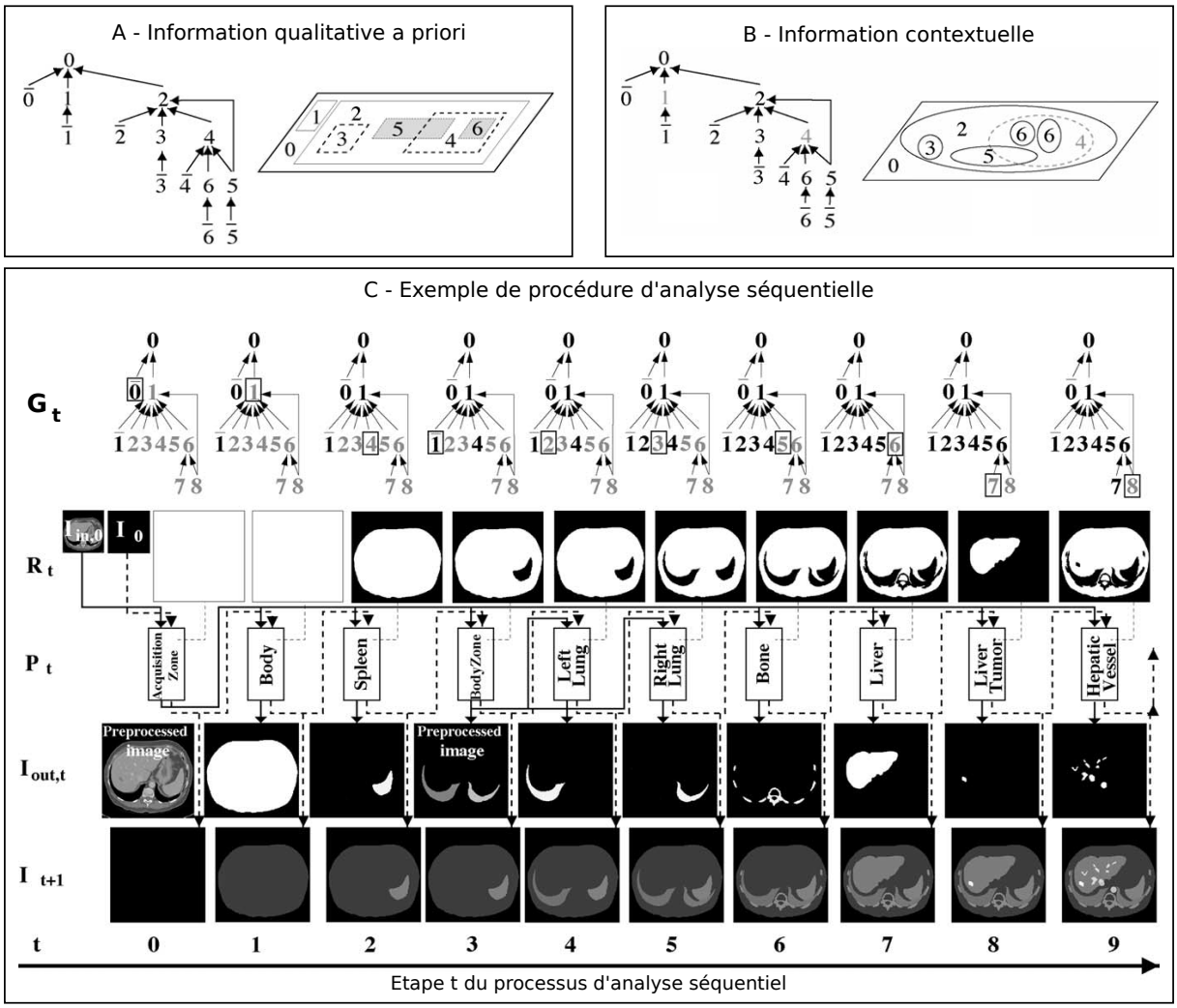

Figure 2.7 : Exploitation des connaissances structurelles qualitatives d'intersection dans un processus d'analyse séquentiel. Cette figure est associée à la publication [45].

$G = (V, F)$:

$$\forall \{k, l\} \in V^2,\ k \xrightarrow{F} l \Rightarrow X(k) \cap X(l) \neq \emptyset, \tag{2.16}$$

où $X(i)$ est la région $i \in V$ de l'image. Ceci est illustré par la figure 2.7-A, que l'on peut associer à des structures anatomiques et pathologiques de l'abdomen. Dans ce cas, les éléments de l'ensemble $V = \{0, 1, 2, 3, 4, 5, 6\}$ correspondent respectivement à la scène complète, à la table (sur laquelle est placé le patient), au corps du patient, à la rate, au foie, aux vaisseaux hépatiques (partie interne et externe au foie) et enfin aux tumeurs hépatiques. L'entité $\bar{i}$ ($i \in V$) est associée au résidu d'une région donnée (i.e. la région dont on enlevé toutes les parties incluses). Par exemple, $\bar{4}$ correspond aux tissus sains du foie (hors réseau vasculaire). Le résidu de 2 ($\bar{2}$) est le corps du patient, à l'exclusion des sous-régions déclarées dans le modèle a priori (i.e. l'ensemble des structures anatomiques et pathologiques considérée dans la modélisation). Ces résidus sont associés à des cibles ne correspondant pas à des structures anatomiques ou pathologiques particulières, mais à des zones de l'image que l'utilisateur peut souhaiter traiter, comme étape préalable à la segmentation ultérieure des entités déclarées dans le modèle. Cette

étape préalable peut, par exemple, consister en une réduction du bruit ou une recherche de composantes connexes.

La procédure d'analyse séquentielle conduit à une information contextuelle que l'on associe à un graphe $G_t = (V_t, F_t)$, où t désigne l'étape dans le processus de segmentation. V_t représente l'ensemble de régions déjà segmentées à l'instant t ($V_0 = \{0\}$ initialement, et $V_\infty = V$ en fin de procédure). La figure 2.7-B donne un exemple d'information contextuelle, matérialisée par le nœud en gras. Dans cet exemple, toutes les structures considérées ont été segmentées, à l'exception du foie (région 4) et de la table (région 1). La figure 2.7-C donne un exemple de procédure, dédiée à la segmentation de différentes structures. Dans cet exemple de séquence, on commence par un prétraitement de la scène complète ("acquisition zone"), le corps du patient ("body"), la rate ("spleen") et ainsi de suite. En fonction de la cible choisie par l'utilisateur (nœud entouré, $\in V \cup \bar{V}$), une chaîne de traitement est initialisée (entité P_t), prenant en entrée l'image I_t et la région d'intérêt R_t automatiquement calculée en fonction du contexte et de la cible. L'entité P_t comprend l'insertion d'opérateurs interactifs ainsi que leur gestion (ceci étant abordé dans les travaux applicatifs section 2.3.1), et peut produire une région identifiée (cas d'un cible $\in V$), qui vient alors enrichir la base des régions identifiées (matérialisée par l'entité I_t, et les nœuds en gras correspondant à G_t).

Cette contribution a porté sur l'établissement des relations (voir [45] pour le détails) permettant d'obtenir la région d'intérêt optimale, en exploitant conjointement les relations a priori connues (graphe G) et l'information contextuelle (graphe G_t), en fonction de la cible désignée par l'utilisateur (cible a priori connu ou résidu).

Les bénéfices concernent le temps de calcul et l'amélioration de l'efficacité des segmentations car les données externes à la ROI sont ignorées (ces données "polluantes" pouvant altérer l'efficacité). L'évaluation expérimentale est détaillée dans [45]. Indépendamment des aspects présentés ici, cette contribution intègre également la problématique de l'encodage des masques des régions segmentées (dans l'entité I_t dans la figure 2.7-C), afin de réduire la consommation mémoire (voir [45]).

2.2.5.2 Analyse séquentielle, relations d'inclusion et photométriques

Au cours de récents travaux [51], nous nous sommes intéressés à l'utilisation des relations d'inclusion et photométriques pour définir, pour une cible donnée, les paramètres d'un algorithme de segmentation particulier, à savoir les *k-means*. Ceci est illustré par la figure 2.8. Les paramètres des k-means sont : la région d'interêt à considérer, le nombre de classes (ou clusters) ainsi que des contraintes pour les graines (centroïdes). Pour les graines, il peut s'agir, comme ceci est illustré, d'une graine dite "statique" : cas où une région cible est déclarée photométriquement similaire a une autre région déjà segmentée. La graine est alors l'intensité moyenne de la région segmentée. Il peut également s'agir d'une graine choisie aléatoirement dans un intervalle restreint, déduit de la connaissance a priori et des structures segmentée. La ROI est déduite selon une approche similaire à la contribution précédente. La composition, a priori connue, de la ROI permet de déduire le nombre de classes, ceci étant requis pour initialiser les k-means.

Dans cette approche, les relations photométriques a priori sont utilisées pour identifier chacune des régions segmentées au sein de la ROI, par mise en correspondance du graphe "modèle" et du graphe issu de la segmentation (appariement exact de graphes). Le principe de cette approche est illustré par la figure 2.8, où les entité inconnues A et B sont logiquement identifiées comme étant respectivement 0 et 1, où A et B sont les étiquettes associées aux deux régions produites par les *k-means*.

A noter que l'approche proposée aborde également le cas d'images en couleur. Les relations

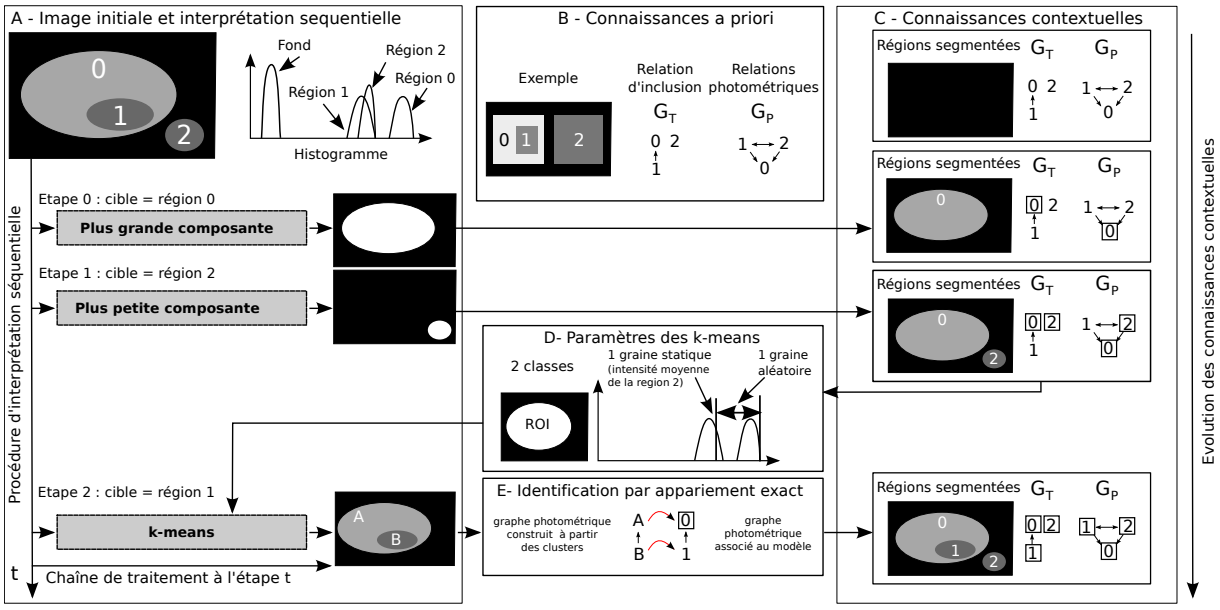

Figure 2.8 : Exploitation des connaissances structurelles qualitatives d'inclusion et de photométrie dans un processus d'analyse séquentiel. Cette figure est associée aux publications [51, 50]. A : Procédure d'interprétation séquentielle. B : Connaissances a priori. C : Evolution des connaissances contextuelles (les nœuds entourés correspondent aux structures progressivement segmentées et identifiées). D : Exploitation des connaissances contextuelles pour estimer les paramètres des *k-means*. E : Identification des régions produites par les *k-means* par appariement exact avec le sous-graphe photométrique du modèle.

photométriques sont basées sur la perception de la teinte, de la saturation et de la valeur des régions dans l'espace de représentation HSV [63]. Ces relations se traduisent par 3 graphes orientés (un graphe par composante) : $G_{P_H}, G_{P_S}, G_{P_V}$, comme ceci est illustré par la figure 2.9. Les relations sont construites en considérant les teintes, la saturation (coloré ou non) et la brillance (clair ou sombre). Dans cet exemple, on suppose que les régions 0 et 1 sont segmentées et identifiées : le contenu de la région 1 est alors segmenté par k-means, produisant 5 régions (A, B, C, D et E) que l'on cherche ensuite à identifier. L'identification par appariement se fait en prenant en compte les 3 graphes $G_{P_{H_C}}, G_{P_{S_C}}, G_{P_{V_C}}$ relatifs à la segmentation (C pour "cluster"), avec les 3 sous-graphes $G_{P_{H_R}}, G_{P_{S_R}}, G_{P_{V_R}}$ issus du modèle, et correspondant à la ROI (R pour "ROI"). Une difficulté concerne le cycle observé selon la composante H, induisant plusieurs isomorphismes possibles (3 dans ce cas). L'ambiguité est levée en retenant l'isomorphisme h vérifiant $h(C) = 3$, cette contrainte étant déduite du contexte ($0 \leftrightarrow 3$ avec 0 déjà identifié) : la région associée à 3 sera celle dont la teinte moyenne est la plus proche de la teinte moyenne de la région 0 (C dans notre cas). Pour plus d'informations, en particulier sur la procédure d'appariement, on pourra se référer à l'article publié sur ce sujet [51].

La figure 2.10 fournit un exemple de chaîne de traitement, ainsi que quelques exemples de résultats obtenus avec la méthode proposée (sans détailler les procédures séquentielles). On observe que notre approche surpasse les k-means standard et conduit, pour les exemples considérés, à de bons résultats, comparativement à des méthodes telles que par exemple le "meanshift" et le "spectral clustering", réputées plus efficaces. Les bénéfices observés concernent également les temps de calcul. L'ensemble des expériences et résultats peuvent être consultés en se référant à

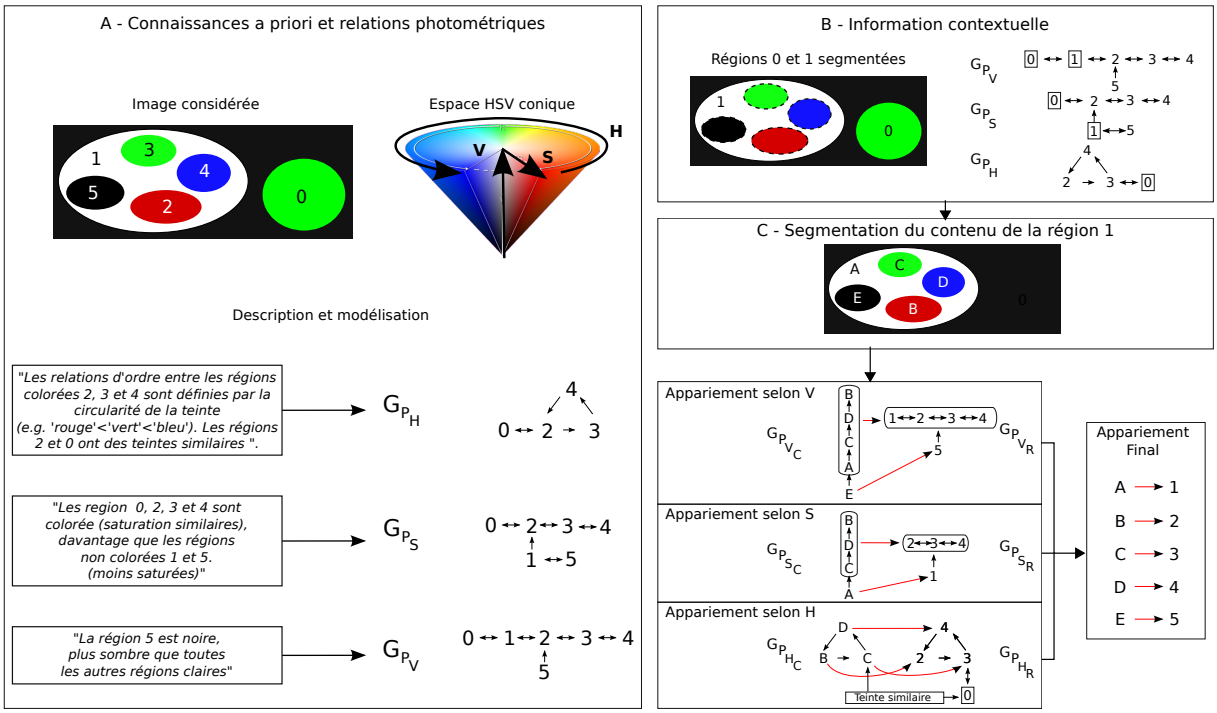

Figure 2.9 : Cas d'images couleur (figure associée à nos travaux [51, 50]). A : Connaissances a priori photométriques et image étudiée. B : Information contextuelle (les nœuds entourés correspondent aux structures segmentées et identifiées). C : Segmentation (*k-means*) conduisant aux régions A à E ("clusters"). L'identification par appariement se fait d'abord selon V, puis S et enfin H.

la publication associée à cette contribution [51].

A noter que ces travaux ont débuté par l'encadrement du Master recherche de Christophe Rigaud [118]. Ils se sont ensuite poursuivis en collaboration avec Nicolas Delanoue (enseignant chercheur à l'Université d'Angers) [50, 51].

2.3 Travaux appliqués

Cette section est consacrée à des contributions appliquées au domaine médical. Cela concerne essentiellement le traitement d'images médicales, et plus récemment l'analyse de données "non-image", par exemple de séries temporelles. Ce dernier point correspond à une activité naissante autour des objets connectés pour la santé.

Les travaux de recherche appliqués présentent trois intérêts majeurs. Tout d'abord, cela permet de lever des verrous scientifiques concernant d'autres disciplines, par exemple, dans le cas présent, en médecine, et ainsi de décloisonner les disciplines. Il peut s'agir d'améliorer la prise en charge de patient par exemple en développant des outils d'aide au diagnostic, au suivi et au geste thérapeutique. Le second intérêt majeur concerne le fait que les applications peuvent permettre d'identifier des verrous dont la portée dépasse le cadre applicatif considéré, ou encore de servir de support à l'élaboration de méthodes plus génériques. Ceci a été mon cas pour l'exploitation de connaissances a priori qualitatives. Enfin, les travaux appliqués sont souvent l'occasion, par exemple lors de l'état de l'art, de découvrir de nouvelles méthodes, ou de

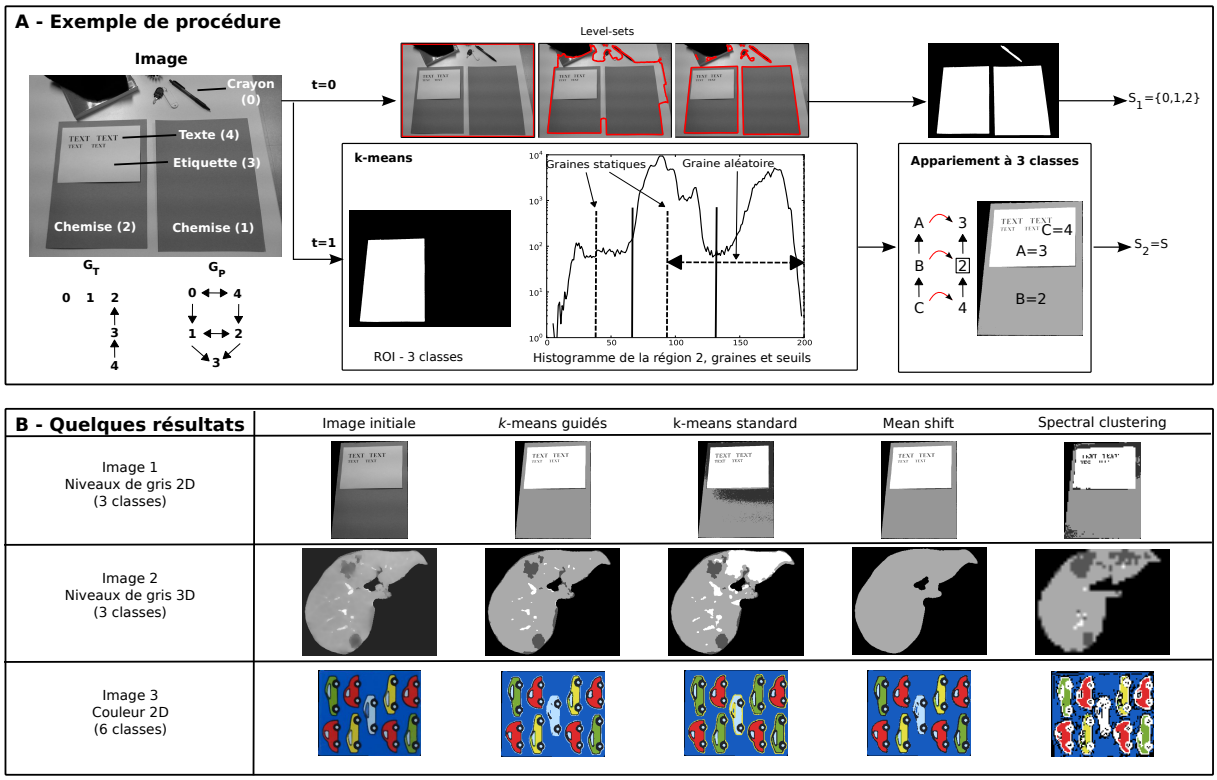

Figure 2.10 : Application de la méthode associée à la figure 2.8 : exemple d'une chaîne de traitement (A), et quelques résultats obtenus (B). Cette figure est associée à la publication [50].

nouvelles variantes, sources d'enrichissement de la culture scientifique. C'est également l'occasion d'observer les tendances, dans un cadre applicatif donné. Ceci a été mon cas, dans ce domaine de santé, avec les objets connectés [134].

Cette recherche appliquée a constitué une part importante de mon activité dès le début de ma thèse, avec les applications dédiées à la détection de tumeurs hépatiques [47] et à la reconnaissance de signatures manuscrites [48]. Depuis 2002, elle s'est poursuivie sur différents thèmes présentés dans les sections suivantes : modélisation du patient pour la chirurgie assistée par ordinateur (section 2.3.1), quantification du flux sanguin dans les carotides (section 2.3.2) et, plus récemment, objets connectés appliqués à la santé (section 2.3.3).

2.3.1 Chirurgie assistée par ordinateur

Ce domaine d'application a concerné la période 2002-2009, passée au sein de l'équipe de recherche et développement en informatique de l'IRCAD[2] (Institut de Recherche contre les cancers de l'appareil digestif).

L'objectif de cet institut, pour la partie concernant les travaux en informatique, est de développer des outils et des procédures pour l'assistance au geste chirurgical guidé par l'image, principalement en chirurgie digestive. Comme ceci est souligné dans un chapitre d'ouvrage publié par notre équipe en 2010 [122], cette problématique a été décomposée en plusieurs axes.

2. IRCAD : https ://www.ircad.fr/fr/

Le premier concerne la modélisation 3D du patient à partir de l'image médicale, la difficulté sous-jacente étant principalement la segmentation et l'identification des différentes structures anatomiques et pathologiques. Un second axe a concerné la planification chirurgicale, à partir de la modélisation 3D du patient. Un troisième axe s'est focalisé sur la simulation chirurgicale pré-opératoire. Enfin, le dernier axe est dédié à l'intra-opératoire, avec le développement de méthodes et d'outils pour la réalité augmentée d'une part et pour la robotique chirurgicale d'autre part (axe porté conjointement par l'IRCAD et par l'équipe Automatique Vision et Robotique du laboratoire ICube[3]).

Cette activité de recherche de l'équipe s'est déroulée dans le cadre de multiples collaborations. On peut citer les collaborations locales avec des équipes du laboratoire ICube de l'Université de Strasbourg et des médecins du CHU de Strasbourg. On peut également mentionner les collaborations internationales, notamment établies dans le cadre de projets européens. On peut donner l'exemple, pour les périodes me concernant, du projet européen ODYSSEUS de 2004 à 2007 et du projet européen PASSPORT de 2008 à 2011.

Dans ce contexte, mon activité a porté sur deux aspects, l'aspect matériel/logiciel d'une part et les méthodes de traitement d'images pour la modélisation 3D du patient d'autre part. L'aspect matériel/logiciel est davantage détaillé dans le chapitre suivant.

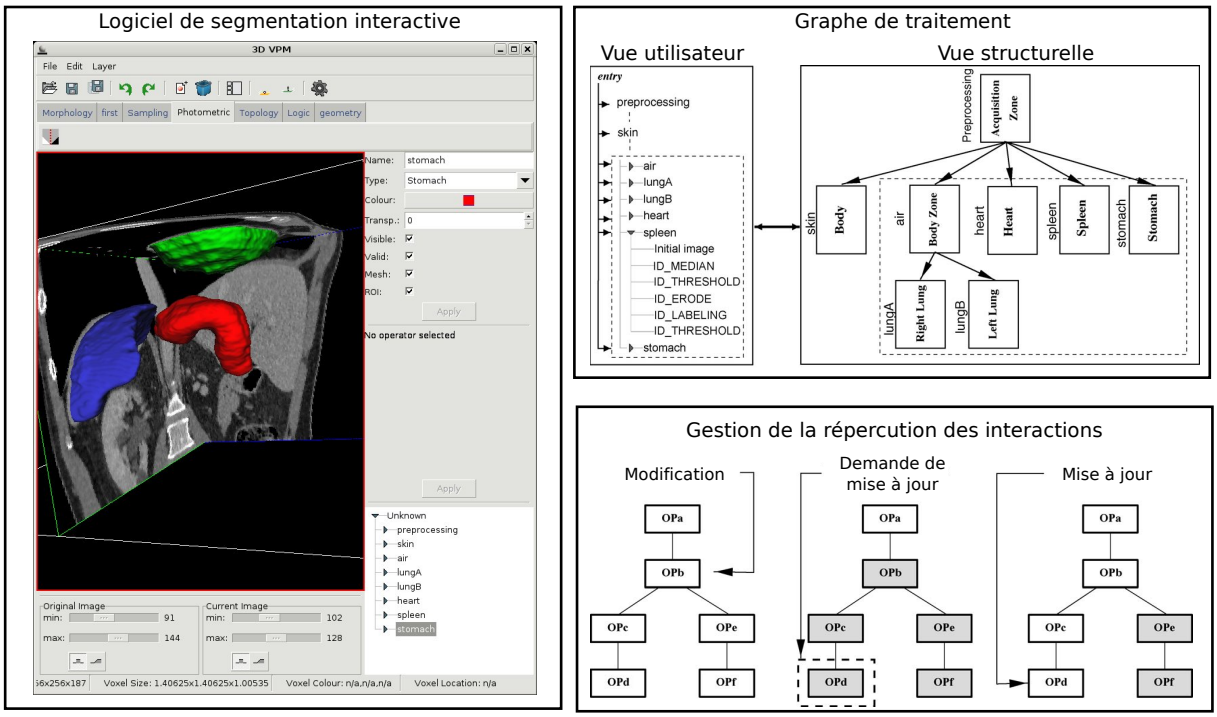

Figure 2.11 : Segmentation interactive des structures anatomiques et pathologiques, avec gestion des interactions au niveau de la chaîne de traitement (graphe matérialisant les dépendances entre les opérateurs). Cette figure est associée aux publications [45, 46].

Concernant les méthodes d'analyse d'images, dans ce contexte de la chirurgie guidée par l'image, mes recherches ont essentiellement porté sur la modélisation 3D du patient, principalement à partir d'images tomodensitométriques. Une approche souvent considérée consiste à combiner des opérateurs de segmentation automatiques avec des interactions du radiologue,

3. http ://icube-avr.unistra.fr

conduisant à un processus d'analyse interactif [45, 80, 81, 111]. Ceci permet de simplifier l'analyse (par rapport à une segmentation entièrement manuelle) tout en exploitant la connaissance de l'expert, qui intervient pour guider les algorithmes de segmentation et/ou pour corriger les défauts. Ceci permet ainsi d'obtenir un système exploitable en routine clinique. Ma principale contribution a concerné la procédure de segmentation séquentielle et interactive présentée précédemment dans la section 2.2.5, permettant de contrôler la région d'intérêt au sein de laquelle les traitements sont contraints [45]. Outre la gestion de cette connaissance a priori et des régions d'intérêt, cette procédure repose sur un enchaînement d'opérateurs de segmentation qui faut définir pour les différentes structures anatomiques et pathologiques. Cet aspect a fait l'objet de travaux spécifiques afin de déterminer ces enchainements appropriés (on pourra consulter nos publications sur le sujet [122, 45, 46]). Ceci sous-entend également la recherche et la mise au point d'opérateurs spécifiques pour certaines étapes de segmentation. Mon implication sur ce point a essentiellement concerné l'encadrement du master de différents étudiants. Par exemple, on peut citer une méthode interactive de détermination des paramètres de la diffusion anisotrope pour la réduction du bruit (Stéphan Kiss, Université de Bordeaux)[82], ou encore le développement de contours actifs interactifs [81] par Benoît Deville (Université de Reims Champagne-Ardenne) [41], Mickaël Hoarau (Université de Strasbourg) [67] et Viviane Leguy (Université d'Orléans) [84].

Indépendamment des algorithmes, une contribution, principalement en collaboration avec mon collègue Vincent Agnus, a concerné la gestion de la chaîne de traitement, afin de supporter les interactions au niveau de l'enchaînement des opérateurs (à la manière d'outils tels que ITK [18]). Ceci est illustré par la figure 2.11.

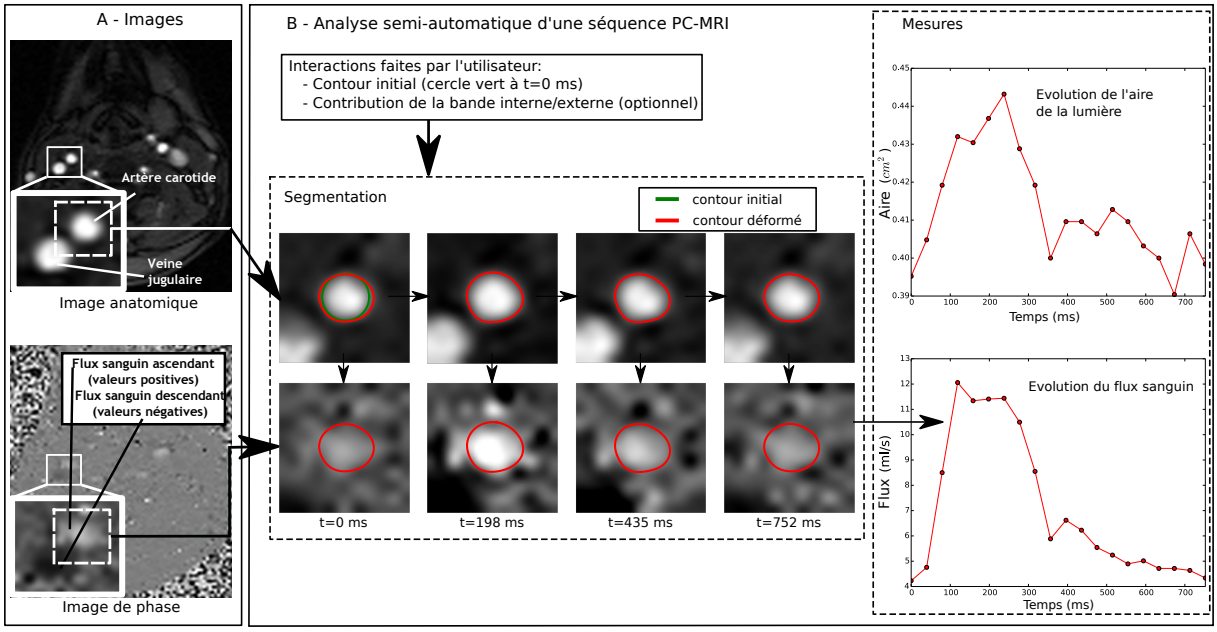

Figure 2.12 : Caractérisation du flux sanguin dans l'artère carotide à partir d'une séquence de vélocimétrie IRM par contraste de phase (PC-IRM). A : l'information acquise comprend une information anatomatique (haut), utilisée pour segmenter la lumière, et une information fonctionnelle, utilisée pour caractériser l'évolution du flux (bas). B : Segmentation de la séquence (gauche) et mesure du flux (droite). Cette figure est associée à la publication [54] et à la thèse de Guillaume Trébuchet [129].

2.3.2 Quantification du flux sanguin dans les carotides

Ce thème a concerné une part de mon activité de recherche de 2010 à 2015. Il s'agit d'un projet initié entre le LARIS (par l'intermédiaire de Christine Cavaro-Ménard) et le CHU d'Angers (par l'intermédiaire du Pr. Serge Willoteaux) auquel j'ai été intégré en 2010, suite à mon arrivée à l'Université d'Angers. Ce sujet a été traité dans le cadre de la thèse CIFRE de Guillaume Trébuchet [129], que j'ai co-encadré avec mes collègues C. Cavaro-Ménard et S. Willoteaux, en collaboration avec l'entreprise Siemens.

Dans le contexte des maladies cardiovasculaires, l'objectif de ces travaux était de quantifier, à des fins diagnostiques, la fonction cardiovasculaire et hémodynamique des patients, en s'appuyant sur l'imagerie par résonance magnétique (IRM), et plus particulièrement sur les séquences de vélocimétrie IRM par contraste de phase (PC-IRM) [124]. Ce type de séquence permet notamment l'acquisition conjointe d'informations anatomiques et fonctionnelles (voir figure 2.12-A). Malgré son réel potentiel par rapport aux autres modalités plus conventionnelles telles que l'imagerie échographie Doppler, cette technique apparaissait sous-utilisée dans la pratique clinique [29]. Afin de pouvoir quantifier le flux sanguin à partir de l'information fonctionnelle, l'objectif était de disposer d'un outil permettant la segmentation de la lumière de l'artère.

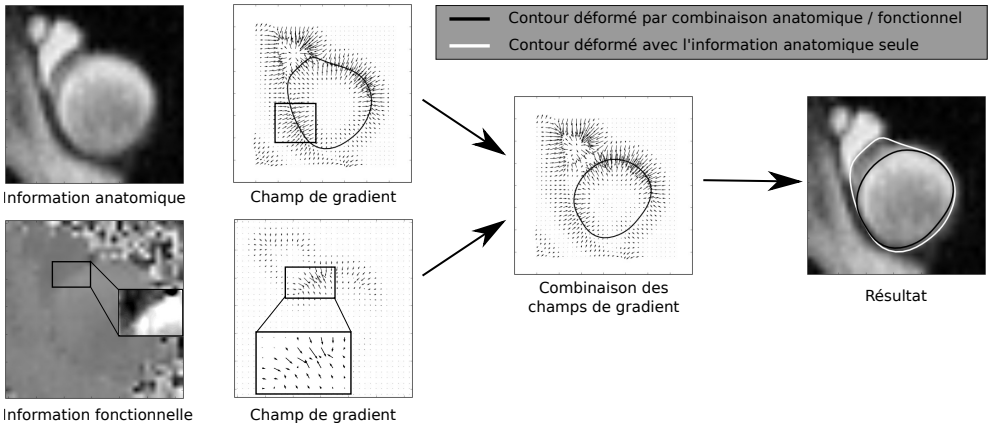

Figure 2.13 : Combinaison de l'information anatomique et de l'information fonctionnelle pour la segmentation de la lumière. Sur cet exemple, la combinaison de ces deux informations semble permettre de mieux délimiter le contour de la lumière, comparativement au seul usage de l'information anatomique. Cette figure est associée à la publication [128] et à la thèse de Guillaume Trébuchet [129].

Une première partie de l'étude a été consacrée à l'étude de l'apport de cette information fonctionnelle, combiner à l'information anatomique, afin de guider la procédure de segmentation. En s'inspirant de travaux analogues [77], ceci a été réalisé en considérant une méthode de segmentation par contours actifs et en ajoutant un second terme d'attache aux données pour guider la déformation, basé sur l'information fonctionnelle (voir figure 2.13). Ceci a fait l'objet d'une communication à un congrès international [128]. Cette approche a finalement été abandonnée au bénéfice d'une approche s'appuyant uniquement sur l'information anatomique, permettant ainsi notamment de s'affranchir de l'estimation des paramètres associés à l'information fonctionnelle.

La difficulté de la segmentation des artères carotides vient notamment du caractère mal contrasté de la lumière (voir figure 2.12-B). Pour cette raison, les travaux se sont orientés vers un guidage de la déformation par une information de type région et non contour [32, 102] (on pourra

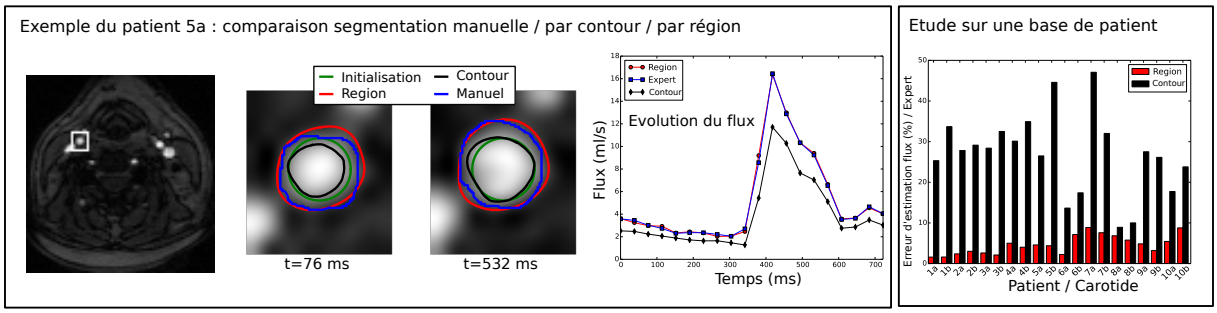

Figure 2.14 : Caractérisation du flux sanguin dans l'artère carotide à partir d'une séquence de vélocimétrie IRM par contraste de phase (PC-IRM). Comparaison entre approche par région et par contour. Cette figure est associée à la publication [54] et à la thèse de Guillaume Trébuchet [129].

consulter [54] pour les détails). Comme ceci est illustré par la figure 2.12-B, la segmentation est faite sur la base de l'information anatomique, permettant d'estimer l'évolution de la surface de la section de la lumière : l'information fonctionnelle (phase) est utilisée pour mesurer l'évolution du flux sanguin. Une part importante des travaux a été consacrée à l'évaluation de la méthode sur une base de séquences réelles, construite spécifiquement pour l'étude, et sur un fantôme, également réalisé pour cette évaluation (la figure 2.14 donne quelques éléments, limités à l'étude sur la base de patients). Ces travaux ont conduit à une communication internationale [127] et à une publication dans un journal international [54].

2.3.3 Objets connectés en santé

Les objets connectés font partie des sujets émergents suscitant un vif intérêt pour la communauté scientifique, en particulier celle dédiée à l'analyse des données, des signaux, voire des images. Comme ceci a été récemment souligné [72], l'internet des objets est un concept reflétant "a connected set of anyone, anything, anytime, anyplace, any service, and any network". Cette nouvelle thématique offre de nombreuses perspectives dans des domaines variés, dont la santé [25, 72, 134].

Une perspective intéressante est la capacité à avoir accès, à distance et de manière régulière, à de multiples informations. On peut par exemple citer les données physiologiques (saturation du sang en oxygène, température, activité cardiaque, poids,...) [9, 134], les informations de localisation (e.g. GPS) et/ou de mouvement (e.g. accéléromètre pour la détection de chute [108]), ou encore des images de l'environnement (e.g. identification et reconnaissance de l'état émotionnel [100]). Ce large panel de données fournies par les capteurs connectés associés présente un grand intérêt, par exemple pour le suivi, notamment à domicile, de personnes présentant des pathologies chroniques [72, 100, 134], de personnes âgées, ou encore de personnes suivants des programmes de rééducation [72].

Cette perspective d'accès à de multiples données hétérogènes de manière régulière (e.g. contrairement à des examens cliniques espacés) nécessite, en particulier, de développer des méthodes adaptées pour fusionner et traiter ces nouveaux jeux de données, en fonction des applications considérées. Les méthodes d'analyse en lien avec cette thématique ne sont pas limitées au traitement d'images (bien que l'image puisse être présente [100])), mais relèvent également du traitement du signal, et, plus généralement, du traitement de données. Il s'agit donc d'un axe fortement pluridisciplinaire, y compris au sein de la communauté scientifique spécialisée en

traitement du signal et des images.

Il s'agit d'une thématique à laquelle j'ai récemment commencé à m'intéresser sous un angle applicatif, notamment en lien avec le CHU d'Angers. La première contribution a porté sur le développement d'une architecture logicielle d'acquisition de données hétérogènes (bio-impédancemétrie, température et questionnaire de santé), à l'occasion de l'encadrement d'étudiants de l'ISTIA. Cette contribution orientée sur l'aspect logiciel est détaillée dans le chapitre suivant, et a fait l'objet de deux conférences internationales [75, 76]. Les relations établies lors de la présentation de ce travail a conduit à l'accueil de la 4ième édition d'une série de conférences internationales sur les objets connectés, à Angers en Octobre 2017. Cela a été également l'occasion de co-éditer les actes de cette conférence [5].

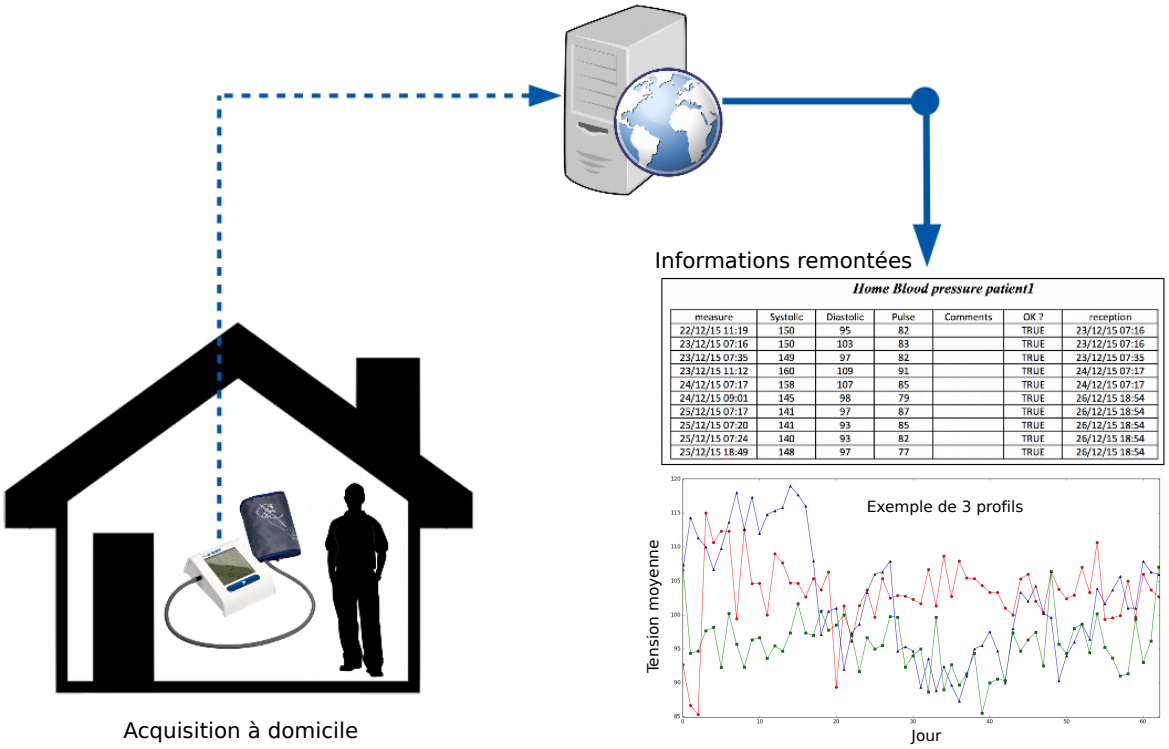

Figure 2.15 : Tensiomètre connecté : les données sont remontées depuis le domicile du patient. L'évolution de la tension moyenne est indiquée (sur environ 60 jours) pour 3 patients présentant 3 profils particuliers. Cette figure est associée à une communication internationale [74] et au master d'Antoine Jamin [73].

Ces travaux se sont récemment poursuivis par l'encadrement du Master recherche d'Antoine Jamin [73], et en collaboration avec Anne Humeau-Heurtier du LARIS, ainsi qu'avec Samir Henni (CHU d'Angers), Pierre Abraham (CHU d'Angers) et Georges Leftheriotis (CHU de Nice). Le travail d'Antoine Jamin a porté sur l'exploitation de données issues d'un tensiomètre connecté pour le suivi à distance de patients atteints d'un cancer et traités par chimiothérapie (voir la figure 2.15). Cette étude a consisté à trouver des descripteurs permettant de distinguer différents profils de patients, sur la base de l'analyse de l'évolution quotidienne de leur pression artérielle, en évaluant le potentiel de différentes méthodes d'analyse de séries temporelles [4]. On peut citer les descripteurs statistiques et l'alignement temporelle dynamique ("Dynamic Time Warping") pour des mesures de dissimilarité entre les séries, ceci ayant été testé avec et sans

approximation linéaire par morceaux [137, 10]. Une autre approche a été étudiée : l'évaluation de la complexité des signaux par mesure de leur entropie multi-échelle [133, 117]. L'usage de cette dernière méthode a permis de relier la complexité de la tension à la prise d'anti-hypertenseurs. Ceci a fait l'objet d'une communication internationale [74].

Plus récemment, des travaux ont débuté sur le thème de l'usage d'un vélo connecté dans le cadre de la maladie d'Alzheimer, à l'occasion de l'encadrement du master de Martin Schmidt. Il s'agit notamment de détecter les troubles cognitifs en étudiant les données captées par le vélo (e.g. vitesse de pédalage et angle du guidon). Par ailleurs, la thèse de Franck Mouney [106] débute actuellement sur le thème de la prévention des accidents cardio-vasculaires à l'aide d'un bracelet connecté. Il s'agit d'une thèse que j'ai initiée avec l'entreprise Axelife, et que je co-encadre avec Mickaël Dinomais (LARIS et CHU d'Angers) et Teodor Tiplica (LARIS).

2.4 Conclusion et perspectives

La première partie de ce chapitre a porté sur des contributions autour de l'exploitation de connaissances qualitatives a priori pour l'interprétation d'images, dans le cadre d'une approche globale puis séquentielle. Les travaux présentés utilisent des techniques basées sur la théorie des graphes et, en particulier, sur l'appariement de graphes. Comme ceci est souligné dans la littérature, cette problématique d'actualité suscite un grand intérêt pour la communauté scientifique et semble demeurer largement ouverte (e.g. absence de formulation unique encore récemment souligné [58]). Les perspectives qu'il semble raisonnable de dégager de ces travaux concerne l'intégration d'autres connaissances qualitatives a priori, aisément formulables par l'utilisateur d'un système de vision (e.g. texture, taille, position, ...). Un point important abordé dans cette partie concerne la stratégie de segmentation, et en particulier l'approche séquentielle. Sur ce point, les perspectives de recherche correspondent à la paramétrisation des algorithmes de segmentation combinés au sein d'une chaîne de traitement (e.g. illustré pour les modèles déformables, le regroupement de points ("clustering") et recherche de la région d'intérêt). Le raisonnement sur les informations structurelles qualitatives pour le guidage semble offrir un large champ d'investigation compte tenu de la multitude d'algorithmes et de chaînes de traitement pouvant être concernés. Sur ce point, d'autres verrous sous-jacents ont été également soulignés, tels que l'optimisation de l'ordre de segmentation et la problématique de la propagation de l'erreur.

La seconde partie a porté sur des contributions et des encadrements dans le cas de travaux tournés vers les applications. La recherche appliquée en traitement d'images, en particulier médicale, est toujours largement d'actualité et est un sujet d'étude important pour la communauté. Mon activité sur ce point s'est réduite au bénéfice des objets connectés appliqués à la santé. Ce thème des objets connectés, en particulier en santé, est un thème émergent qui pose de multiples problèmes (capteurs, analyse de données hétérogènes, communication, ...) et offre de nouvelles perspectives, typiquement le suivi "continu" de personnes à domicile. Il s'agit d'un thème pluridisciplinaire que j'ai récemment abordé et que je compte développer, dans le cadre de projets de recherche appliquée. Compte tenu du caractère fortement pluridisciplinaire de ce thème, cette activité est et sera portée par des collaborations avec des chercheurs des différentes communautés impliquées (e.g. médecine, signal, image, analyse de données).

Chapitre 3

Implémentation matérielle et logicielle

3.1 Introduction

Nous nous intéressons, dans ce chapitre, à la problématique de l'implémentation de méthodes d'analyse d'images (voire, plus récemment, de données). Cette thématique occupe une part importante de mon activité depuis 2002, en particulier pendant la période 2002-2011. Ce sujet continue à être omniprésente dans mon activité actuelle, même si mon investissement a été considérablement réduit depuis 2011.

Dans ce chapitre, cette problématique de l'implémentation est abordée en considérant deux enjeux importants : la performance et la généricité. La performance concerne essentiellement le temps de calcul, tandis que, dans notre contexte, la généricité se rapporte à la notion de réutilisation du code au sein de différents programmes. Par exemple, ceci peut concerner la réutilisation d'un algorithme de traitement d'images dans une application de réalité augmentée alors qu'il a initialement été conçu pour un logiciel de segmentation interactive d'images. Il en va de même pour d'autres fonctionnalités telles que la visualisation 3D et les interfaces graphiques par exemple.

Je présente ci-après mes contributions sur le sujet en considérant deux types d'implémentation : l'implémentation matérielle et l'implémentation logicielle (supposée abstraire la couche matérielle).

L'aspect matériel est traité dans la section 3.2, l'enjeu étant principalement celui de la performance. Cet aspect est abordé de manière assez succincte car mes contributions sur le sujet concerne essentiellement la période 2003-2006. Il est néanmoins intéressant d'aborder ce point qui est complémentaire à l'autre volet et demeure un sujet d'étude important pour la communauté s'intéressant au calcul scientifique (et au traitement d'images en particulier). Le volet relatif à l'implémentation logicielle est traité de manière plus détaillée dans la section 3.3. Dans ce cas, les enjeux sont simultanément la recherche de performance et de généricité.

3.2 Aspect matériel

Les travaux de la communauté sur ce sujet portent sur l'exploitation des propriétés des dispositifs de traitement spécialisés pour reduire les calculs, par exemple en exploitant le paral-

lélisme. Actuellement, les principaux dispositifs de traitement spécialisés sont numériques. On peut par exemple citer les processeurs multicœurs, les processeurs des cartes graphiques (GPU pour "graphical processing unit") [43], les processeurs numériques spécialisés pour le traitement du signal (DSP pour "Digital Signal Processor"), ou encore les FPGAs ("Field Programmable Gate Array") [71, 87]. Ces dispositifs permettent d'effectuer les opérations de manière massivement parallèle, augmentant ainsi les performances. Une difficulté concerne l'adaptation des algorithmes à ces architectures, ceci constituant un sujet d'étude pour la communauté concernée. A ce sujet, on pourra par exemple consulter le récent état de l'art sur l'implémentation sur GPU d'algorithmes de traitement d'images médicales [43].

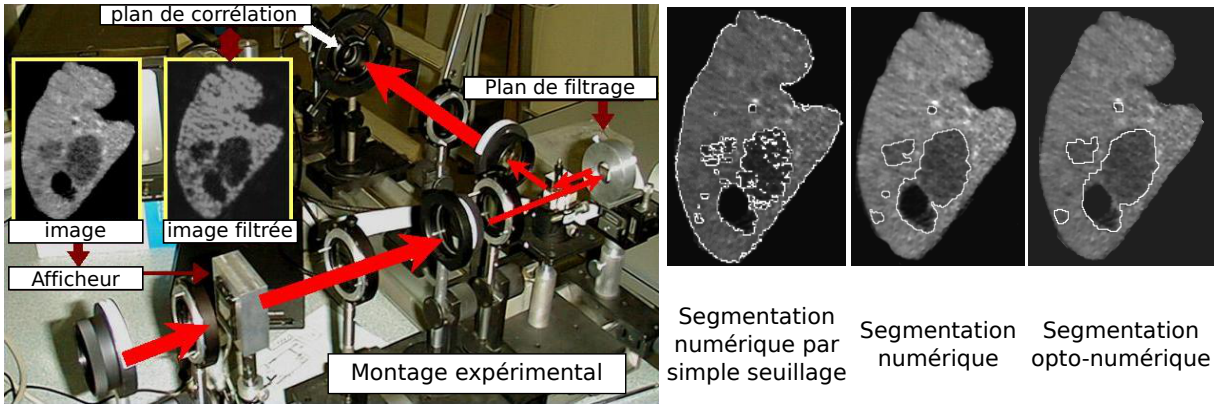

Figure 3.1 : Exemple de processeur opto-électronique (corrélateur de Vander Lugt) développé durant la période de thèse, et associé à la publication [47].

Bien que probablement moins courants, il existe également des travaux concernant les traitements par voie optique [7, 89, 103, 104], avec des applications telles que, pour citer quelques travaux récents, la reconnaissance faciale [26] et la reconnaissance de l'iris [112]. Le traitement par voie optique permet de profiter du parallélisme inhérent aux techniques optiques [79, 104], ainsi que des bandes passantes élevées des interconnections optiques. A ce sujet, on pourra consulter par exemple une récente étude sur l'implémentation opto-électronique de réseaux de neurones convolutifs pour l'apprentissage profond [89], avec par ailleurs l'avantage d'une réduction de la consommation en énergie. A noter que, comme récemment souligné [89, 104], l'optimisation des traitements peut se faire en considérant des architectures hétérogènes, couplant traitements optiques et dispositifs numériques spécialisés (e.g. FPGA, GPU). Le traitement par voie optique (ou opto-électronique) a constitué mon sujet de thèse, avec deux contributions principales concernant la détection de tumeurs cancéreuses hépatiques et la reconnaissance de signatures manuscrites, en s'appuyant sur un corrélateur optique [47, 48] (voir figure 3.1).

Suite à ces travaux, j'ai eu l'opportunité de monter le projet POEME[1]. Il s'agissait d'un projet collaboratif entre l'université de Strasbourg, le laboratoire ICube (anciennement Phase et InESS), l'IRCAD et l'entreprise MicroModule [2]. Cette collaboration s'est traduite par l'obtention du financement de la thèse de Morgan Madec (thèse CIFRE) [97], que j'ai co-encadré de 2003 à 2006 avec Yannick Hervé (directeur de thèse) et Wilfried Uhring (co-encadrant). Les travaux ont principalement porté sur la conception d'une architecture opto-électronique dédiée à la

1. POEME : Processeur Opto-Electronique pour le MEdical.
2. http ://www.micromodule.fr/

reconstruction d'images tomodensitométriques issues d'un scanner à rayons X. Un des enjeux concernait l'implémentation optique de la reconstruction, et son couplage avec les dispositifs opto-électroniques et numériques permettant de réduire les temps de calcul (voir figure 3.2 pour un modèle du cœur optique du système).

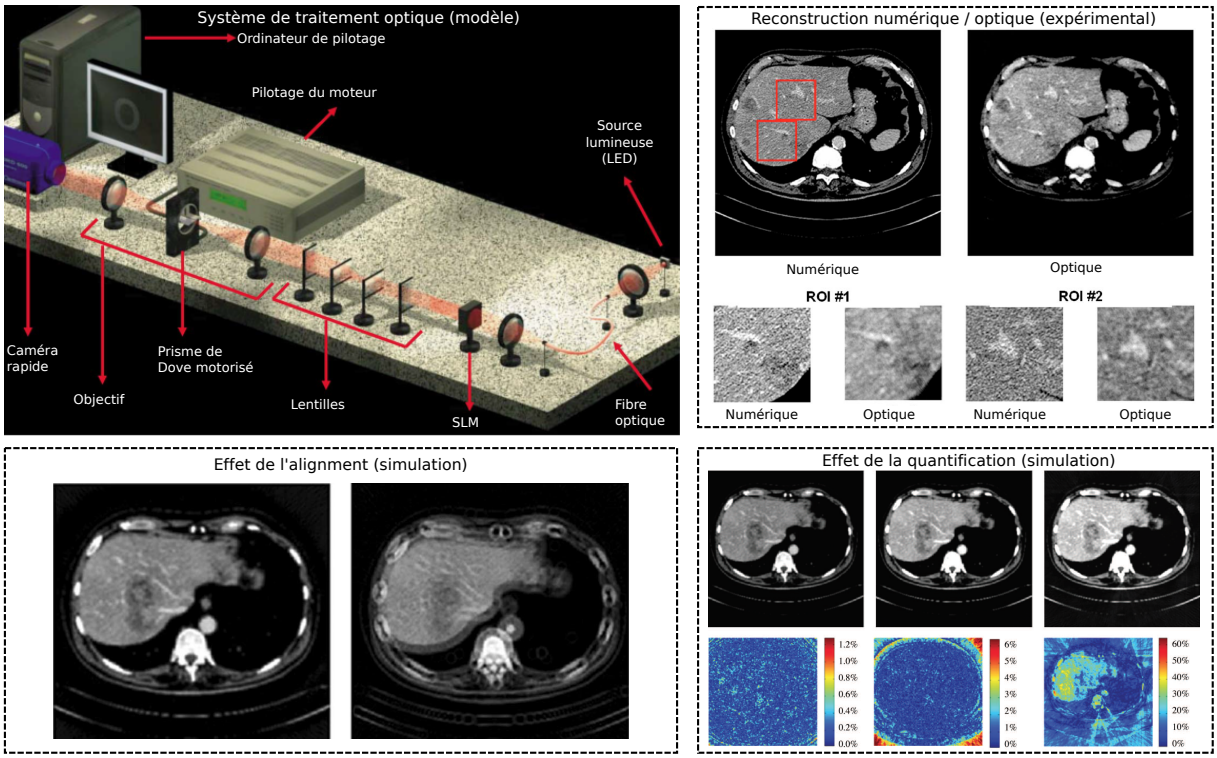

Figure 3.2 : Cœur opto-électronique de l'architecture proposée pour la reconstruction d'images tomodensitométriques (modèle). Un système expérimental a également été réalisé pour comparer les reconstructions numériques et optiques (haut-droite). Les effets de la dynamique limitée des interfaces (quantification) et de l'alignement ont été étudiés (simulation). Pour l'effet de la quantification (8 bits, 4 bits et 1 bit), l'erreur par rapport à l'image reconstruite numériquement est également rapportée. Ces figures sont extraites de la publication [92].

Dans ce cadre, des études spécifiques ont été conduites pour notamment évaluer l'impact de la dynamique limitée (quantification) inhérente aux interfaces opto-électroniques sous-jacentes, ainsi que des défauts d'alignement des composants (voir figure 3.2-bas). Ces travaux ont conduit à la publication de 4 articles dans des revues internationales [91, 92, 95, 96], des communications internationales [93, 94] et à un brevet [90].

Même si les travaux relatifs au traitement par voie optique (ou plutôt opto-électronique, voire opto-numérique car combinant des composants optiques et numériques) sont prometteurs comme le montre la littérature récente sur le sujet, les dispositifs purement numériques spécialisés restent plus répandus (exemple des GPU disponibles sur la plupart des ordinateurs). De plus, l'implémentation des algorithmes est simplifiée, en particulier grâce à des langages et librairies masquant la spécificité de l'architecture sous-jacente (e.g. OpenCL [37] pour GPU, CPU multicœur et FPGA [37, 43]).

3.3 Aspect logiciel

Au niveau de l'aspect logiciel, les enjeux relatifs à l'implémentation concernent certes les performances, mais également la généricité et la simplicité d'utilisation. Une part importante des travaux de recherche en analyse des images est consacrée à ce volet logiciel, ceci étant attesté par la production scientifique associée, à l'interface entre la communauté "signal-image" et génie logiciel.

Une part importante des travaux de recherche se focalise sur la performance en travaillant, en amont, au niveau algorithmique, sur la formulation (ou la reformulation) mathématique, des méthodes afin d'optimiser leur implémentation. On peut citer l'algorithme classique de la transformée de Fourier rapide [27, 35] (FFT pour "Fast Fourier Transform"), conçue en reformulant de manière pertinente le calcul de la transformée de Fourier discrète. Sous condition que le nombre d'échantillons du signal soit une puissance de 2, la FFT permet de réduire considérablement le nombre d'opérations. Sur ce thème de la représentation et de l'analyse fréquentielles des signaux et des images, on peut également mentioner la transformée rapide en ondelettes (FWT pour "Fast wavelet transform")[16]. En morphologie mathématique, en travaillant sur la décomposition des éléments structurants, et en utilisant les propriétés de symétrie, on peut considérablement réduire le nombre de calculs [38, 65]. Cette activité de recherche n'est pas limitée aux algorithmes élémentaires dont quelques exemples viennent d'être rappelés. Ceci concerne également des algorithmes plus sophistiqués. On peut par exemple mentionner, dans le domaine du calcul de descripteurs locaux invariants en rotation et en échelle (e.g. descripteurs SIFT proposé par D. Lowe [88]), les descripteurs SURF (*"Speed Up Robust Feature"*). Ces descripteurs ont été proposés avec le souci, en amont, des performances en temps de calcul [13].

Ces exemples se focalisent principalement sur la notion de performance. Un autre notion importante est celle, mentionnée précédemment, de la généricité, afin notamment de faciliter la réutilisation des algorithmes (et de fonctionnalités connexes telles que la visualisation par exemple). Cette section est ainsi principalement organisée autour de ces deux notions. Nous évoquons tout d'abord le problème du choix d'un langage et de l'écosystème associé, en soulignant quelques publications associées. Ensuite, nous nous focalisons sur l'intégration et l'adaptation de notions empruntées au monde du logiciel, avec les notions de patrons de conception, d'approche orientée "composant" et enfin la notion de "rôle" à laquelle j'ai eu l'occasion de contribuer. La dernière section portera succinctement sur cette problématique du logiciel dans le domaine émergent des objets connectés en santé.

3.3.1 Langages et écosystèmes associés

La question du choix du langage, et de l'écosystème associé (i.e. ensemble de librairies implémentées avec un langage donné) est probablement la manière la plus simple d'aborder cette problématique d'implémentation logicielle de méthodes de traitement d'images.

Dans ce cas, l'enjeu de la performance est souvent combiné à la recherche de généricité, à la richesse des fonctionnalités (écosystème) et mécanismes associés et à la simplicité d'utilisation. Par les exemples, les langages Fortran, C et C++ sont réputés plus performants, et pour des raisons historiques, plus riches en librairies de calcul scientifique que des langages tels que Java, Python ou Matlab. Ils sont cependant moins simples à utiliser. Sur ce point, on peut citer la récente publication autour de HOPE, une librairie Python dédiée au calcul scientifique en astrophysique [6], soulignant cette différence de performances entre le C++ et Python, ainsi que la simplicité d'utilisation de ce dernier et la richesse des fonctionnalités fournies par les librairies

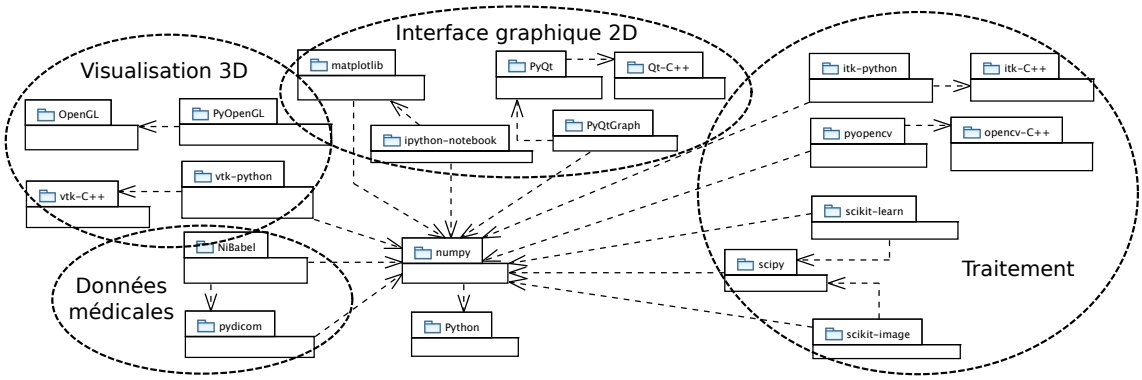

Figure 3.3 : Modélisation partielle (diagramme UML de package) de l'écosystème des librairies associées au langage Python pour des applications scientifiques. Cette figure est extraite de la publication [53].

associées. Concernant ce langage en particulier, on peut également citer quelques publications soulignant la pertinence de cet outil pour le calcul scientifique en général [114, 115], ou pour certaines fonctionnalités plus ciblées. On peut par exemple citer Osirix pour le médical [19], ou la librairie scikit-learn, devenue un standard en apprentissage automatique, et ayant fait l'objet d'un article dédié [113]. Nous avons nous même présenté au colloque GRETSI, avec Christian Jeanguillaume (MCU-PH au CHU d'Angers), certains éléments autour de ce langage et de son écosystème [53]. Les figures 3.3 et 3.4 montrent respectivement une modélisation partielle de l'écosystème associé au langage Python, ainsi que quelques exemples d'applications réalisées pour illustrer la richesse de ce dernier.

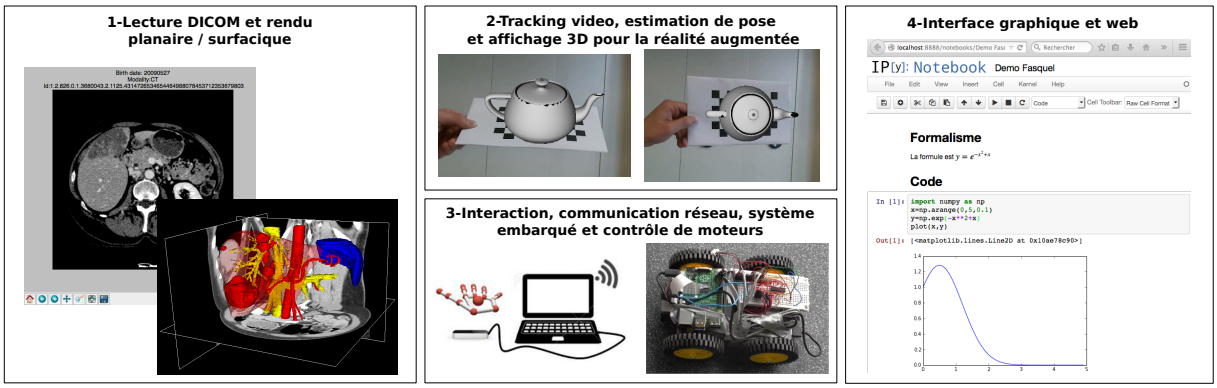

Figure 3.4 : Exemples d'applications basées sur l'écosystème Python. Cette figure est extraite de la publication [53].

Sur le même sujet, avec Alban Vergnaud, alors étudiant en thèse au LARIS et Laurent Autrique, nous avons discuté de la pertinence de cet outil, pas seulement pour la recherche mais également pour l'enseignement de l'automatique. Ceci a récemment donné lieu à une communication dans une conférence internationale [132].

3.3.2 Algorithme et patron de conception

Au délà des aspects soulignés précédemment autour des langages et écosystèmes, la performance et la généricité de l'implémentation logicielle d'algorithmes peuvent être abordées en exploitant des mécanismes et des concepts empruntés à la communauté du génie logiciel, ou relevant de celle-ci. Par exemple, on peut citer les mécanismes mis en œuvre dans les librairies ITK et VTK [78], qui sont des librairies de référence, écrites en C++, dédiées respectivement au traitement et à la visualisation d'images, principalement médicales. Du point de vue de la conception, ces frameworks reposent sur une architecture de type "pipe and filter" [18, 68], issue du monde du génie logiciel [116] (e.g. mécanisme intégré aux systèmes d'exploitation de la famille UNIX). De plus, ces librairies exploitent d'autres mécanismes tels que la programmation générique [36], permettant d'abstraire le code source du type des données, et en particulier de celui des images (dimension spatiale et dynamique de code). Un élément de conception important correspond à l'homogénéisation, dans le cas du paradigme objet, des interfaces (ou classes), afin de faciliter le chaînage des algorithmes, ainsi que l'intégration aisée de nouveaux. Ceci est le cas dans ITK [18], mais également dans d'autres librairies récentes telles que scikit-learn [113], avec l'abstraction de la notion d'estimateur. Les patrons de conception (ou "design pattern") [61], issus du monde du génie logiciel, font partie des concepts souvent incontournables pour une implémentation générique d'algorithmes. On peut citer l'exemple du patron de conception *itérateur* ("iterator") permettant d'abstraire la méthode de parcours d'une image, et qui est implémenté dans les librairies de traitement d'images.

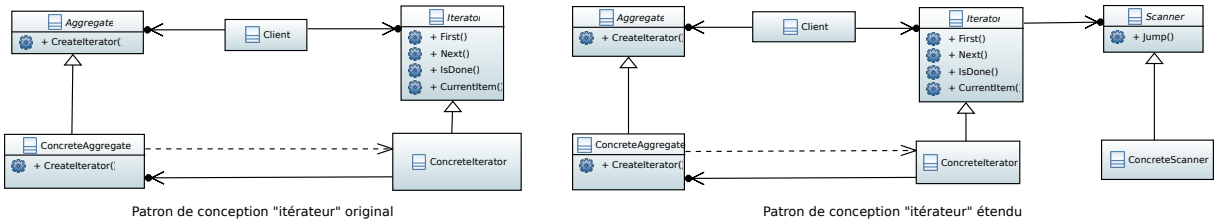

Figure 3.5 : Généricité du parcours d'une image et gestion de régions d'intérêt par extension du patron de conception "itérateur" (diagramme UML de droite). Cette figure est associée aux publications [44, 56].

Sur ce point, je peux citer une contribution, en collaboration avec Vincent Agnus et Julien Lamy (IRCAD), visant à étendre ce patron afin de contraindre les traitements au sein de régions d'intérêt (ROI) de formes quelconques. Cet aspect est associé aux méthodes présentées dans le chapitre 2 autour de l'usage des connaissances qualitatives a priori et des graphes pour déterminer la ROI au sein de laquelle on souhaite pouvoir restreindre les traitements. L'enjeu de cette contribution a été de préserver la généricité des algorithmes de traitement d'images tout en augmentant les performances, par réduction du nombre de points traités. Cette contribution a fait l'objet d'une publication et d'une communication [44, 56]. La figure 3.5 résume notre contribution au moyen de diagrammes UML de classes. Un algorithme traite les pixels d'une image en parcourant celle-ci. Dans les modèles associés à la figure 3.5, l'algorithme est matérialisé par l'entité *Client* et l'image par la l'entité *Aggregate* : lors du parcours, l'algorithme s'adresse à l'entité de parcours *Iterator* et non directement à l'image pour aller de point en point ou accéder à la valeur d'un point. L'abstraction de la région parcourue a été mise en place par l'intégration d'une nouvelle entité : le *Scanner*. Cette entité définit les sauts à opérer lorsque l'on passe d'un pixel à l'autre afin de ne pas traiter chaque pixel mais seulement ceux définis

par la région d'intérêt. Cette proposition a été évaluée en intégrant ce modèle à la librairie ITK. Plusieurs méthodes de parcours ont été testées. Il pouvait par exemple s'agir d'une méthode associée à une forme régulière (e.g. cube) ou à une forme quelconque (par exemple associée à un "masque" définissant la forme de la région à traiter). Cette évaluation a validé la pertinence de ce modèle, en permettant de changer le mode de parcours sans altérer l'algorithme initial, tout permettant de réduire considérablement les temps de calcul. Cela peut par exemple permettre d'utiliser directement le masque du foie plutôt qu'une simple "boîte englobante" pour traiter cet organe, comme ceci est illustré par la figure 3.6.

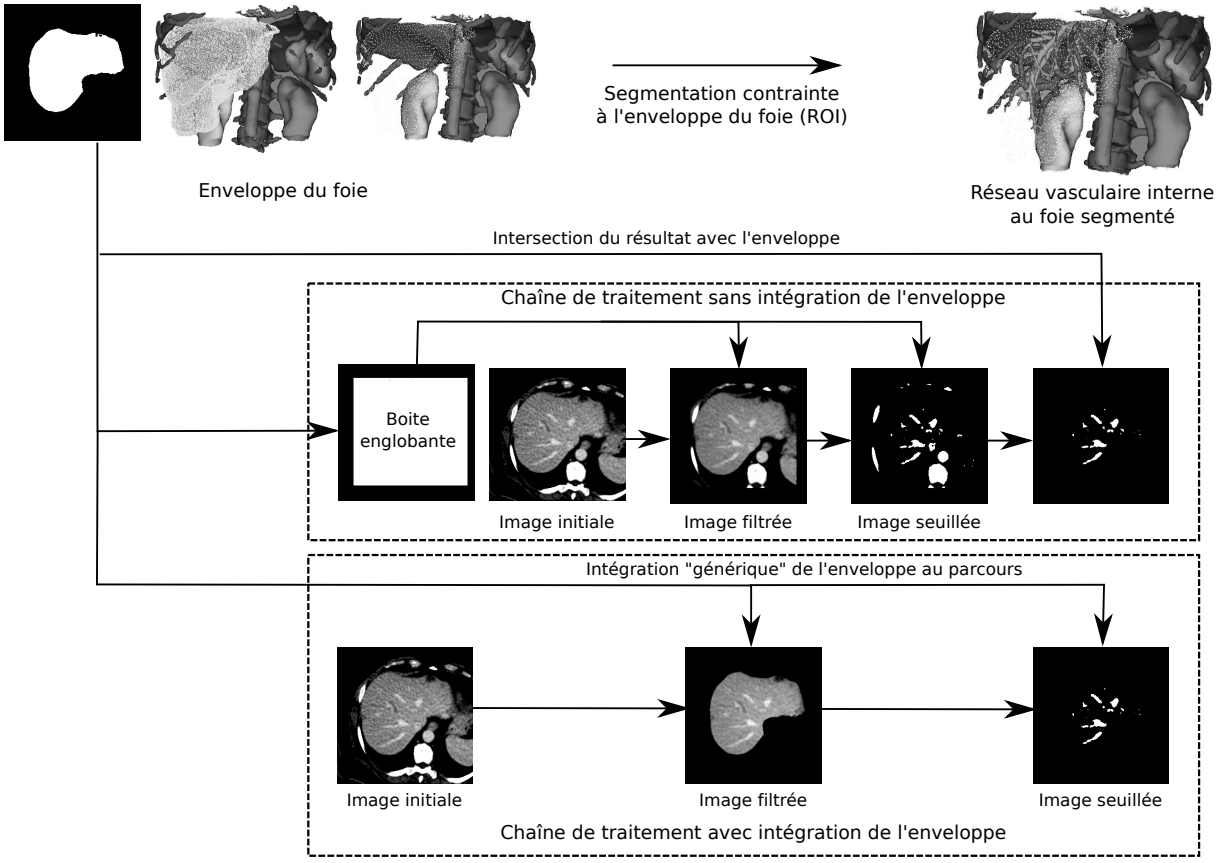

Figure 3.6 : Généricité du parcours d'une image et gestion de régions d'intérêt : exemple de la chaîne de traitement contraignant les calculs directement à l'enveloppe du foie (en bas) plutôt qu'à une boîte englobante (milieu). Pour cet exemple, la chaîne de traitement considérée est élémentaire : filtrage (diffusion anisotrope) suivi d'une simple seuillage. Cette figure est associée aux publications [44, 56].

3.3.3 Conception de logiciels

Cette partie porte sur la conception de logiciels. Nous nous intéressons ci-après à la combinaison des algorithmes et des fonctionnalités complémentaires, telles les interfaces graphiques et la visualisation (2D et 3D), la gestion des données [68] et l'interopérabilité avec des systèmes robotisés.

Interface du logiciel de modélisation

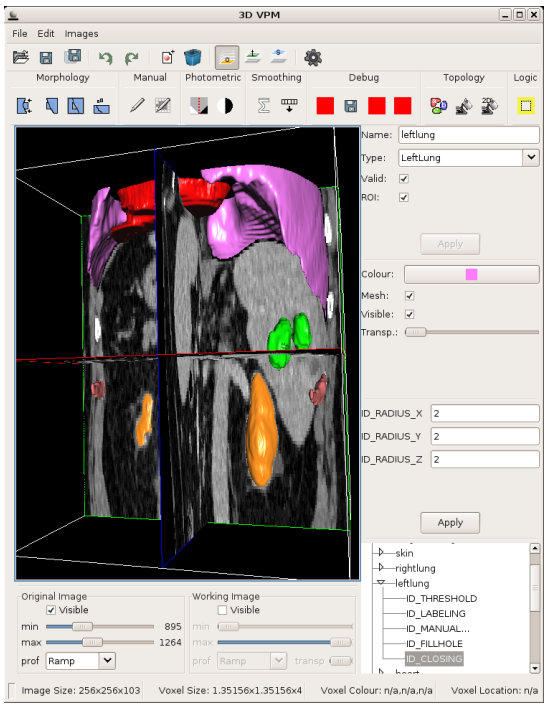

Diagramme de composants

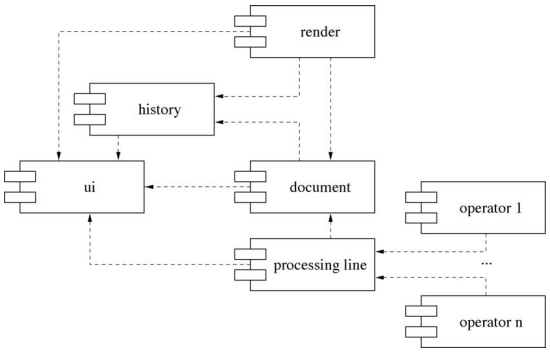

Diagramme de classes et patron de conception "Strategy"

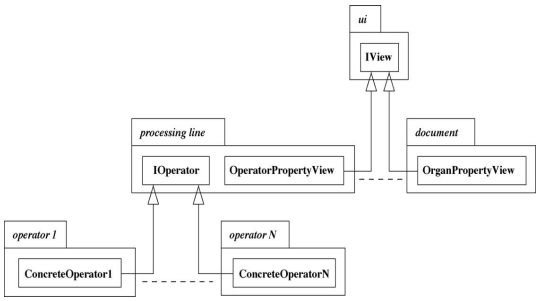

Figure 3.7 : Quelques éléments autour de l'architecture d'un logiciel de modélisation du patient par segmentation interactive de son image médicale 3D. Les notions relatives au génie logiciel utilisées pour son architecture sont : la notion de composant (haut-droite) et la notion de patron de conception (bas-droite, limité au patron "strategy"). Cette figure est associée à la publication [46].

Cette problématique du logiciel est omniprésente dans la discipline de la vision, et est souvent abordée par rapport à des domaines applicatifs précis. Ceci est attesté par les publications dont l'objet n'est pas seulement de proposer un algorithme ou une méthode de traitement spécifique, mais bien de proposer un logiciel, incluant une description de ses fonctionnalités et/ou de son architecture. En se limitant à quelques publications récentes, on peut citer le logiciel IJ-OpenCV pour l'analyse d'images biomédicales [42], un article autour de la conception d'un logiciel dédié à la radiothérapie [136], le logiciel NMSBuilder [130] pour la création de modèles musculo-squelettiques, un environnement pour le recalage et la visualisation d'images médicales [138], le logiciel Quartz pour l'analyse d'images rétiniennes [60], un logiciel modulaire pour la chirurgie dentaire guidée par l'image [17] et un logiciel pour les interventions chirurgicales guidées par ultrason [83].

Parmi les difficultés rencontrées, on peut également souligner l'hétéorégénéité de certaines architectures, telles que celles impliquant des systèmes embarqués (e.g. système embarqué de traitement d'images pour le comptage de poissons [66]) ou les applications web (e.g. traitement et visualisation d'images médicales par application web [99]), ou encore les systèmes intégrant des périphériques particuliers (e.g. intervention guidée par images ultrasonores,[83], cité précédemment).

Dans ce contexte, je me suis intéressé, durant la période 2002 à 2009 à la conception d'un logiciel dédié à la modélisation du patient à partir d'images 3D. Un enjeu de ce logiciel est de faci-

liter l'exploitation d'algorithmes en routine clinique [83], notamment en masquant la complexité sous-jacente de la configuration des algorithmes et des chaînes de traitement. Un point important soulevé, lors de ces travaux, a été l'usage, pour cette application spécifique, de concepts utilisés en génie logiciel, tels que les patrons de conception déjà mentionnés précédemment [101], ou encore la notion de composant logiciel [24, 61, 125, 121]. Ce travail, réalisé avec les membres de l'équipe au sein de laquelle je travaillais (en particulier Guillaume Brocker pour la conception), a fait l'objet d'une publication dans une revue internationale, à l'interface entre la communauté du traitement de l'image et du logiciel [46] : la contribution principale portait sur le lien entre ces notions classiques du génie logiciel et la problématique de l'analyse interactive d'images médicales, incluant leur visualisation. Ceci est illustré par la figure 3.7. Les composants désignent des entités (assimilable en partie à des librairies) avec des dépendances limitées et bien identifiées [125]. En particulier, le composant "processing line" (gérant l'enchaînement des algorithmes de traitement) déclare une interface dédiée aux algorithmes de traitement d'images. Cette interface sera implémentée par les composants "opérateur" (notion de "plugin") qui fourniront les algorithmes effectifs. Les dépendances sont limitées et bien identifiées : le cœur du logiciel ne dépend pas des opérateurs, qui peuvent être intégrés dynamiquement, permettant ainsi d'éviter d'avoir un logiciel très monolithique. Le patron de conception mis en place pour intégrer un nouvel opérateur de traitement d'image est le "Strategy" [24, 61]. A noter que d'autres patrons de conception couramment utilisés en génie logiciel ont été intégrés. On peut notamment citer "singleton" (pour l'unicité de l'entité "patient") et "observer" (pour la synchronisation des données) [24, 61]. Indépendamment des composants dédiés au traitement d'images, les autres composants considérés dans cette architecture sont dédiés à l'interface graphique ("ui"), la visualisation 3D ("render"), l'historisation des commandes ("history", permettant notamment de déjouer/rejouer des opérations) et enfin le "document". Le document gère l'entité "patient", intégrant l'ensemble des structures anatomiques et pathologiques segmentées, indépendamment des traitements réalisés pour les segmenter et les identifier.

Dans un souci de généricité, nous avons également travaillé à l'extension de cette application de modélisation du patient à une fabrique de logiciels ou suite logicielle ("software product line"[1, 120]). L'objectif a été de faciliter la réutilisation de fonctionnalités au sein d'une famille de logiciels. Par exemple, il s'agissait de réutiliser les composants de visualisation d'un logiciel de modélisation du patient dans un logiciel de planification chirurgicale. Pour répondre à cette problématique, nous avons proposé une architecture couplant le concept de "rôle" à celui de "composant" déjà considéré [55]. La notion de rôle a permis de faciliter la déclaration des différents logiciels considérés dans notre activité : visualisation 3D seule, modélisation du patient, réalité augmentée, ... Un logiciel donné était déclaré comme un objet auquel on pouvait dynamiquement, et de manière standardisée (pour la généricité), affecter des "rôles". Ces "rôles" sont assimilables à des fonctionnalités de traitement, d'affichage, de chargement de données... Le cœur de notre contribution sur ce point se résume essentiellement à un patron de conception proposant une implémentation de cette notion de "rôle" (illustré par la figure 3.8 haut-gauche). Cette notion a été couplée à la notion de composant (voir figure 3.8 haut-droite). La partie inférieure de cette figure 3.8 donne un exemple d'usage de ce modèle pour implémenter une application élémentaire. Cette application réalise une érosion morphologique sur une image médicale stockée au format Dicom. Les images avant et après traitement sont affichées en 3D au moyen d'un rendu multiplanaire (une coupe dans un volume 3D dans notre cas) et les paramètres d'érosion peuvent être saisis au moyen d'un élément graphique d'édition. Structurellement, cette application est un objet (entité `:Objet`) contenant différents champs (objets spécifiques telles des images). Chaque objet se voit affecter un "rôle" particulier : lecture DICOM (`:Dicom`), traitement (`:Erode` -

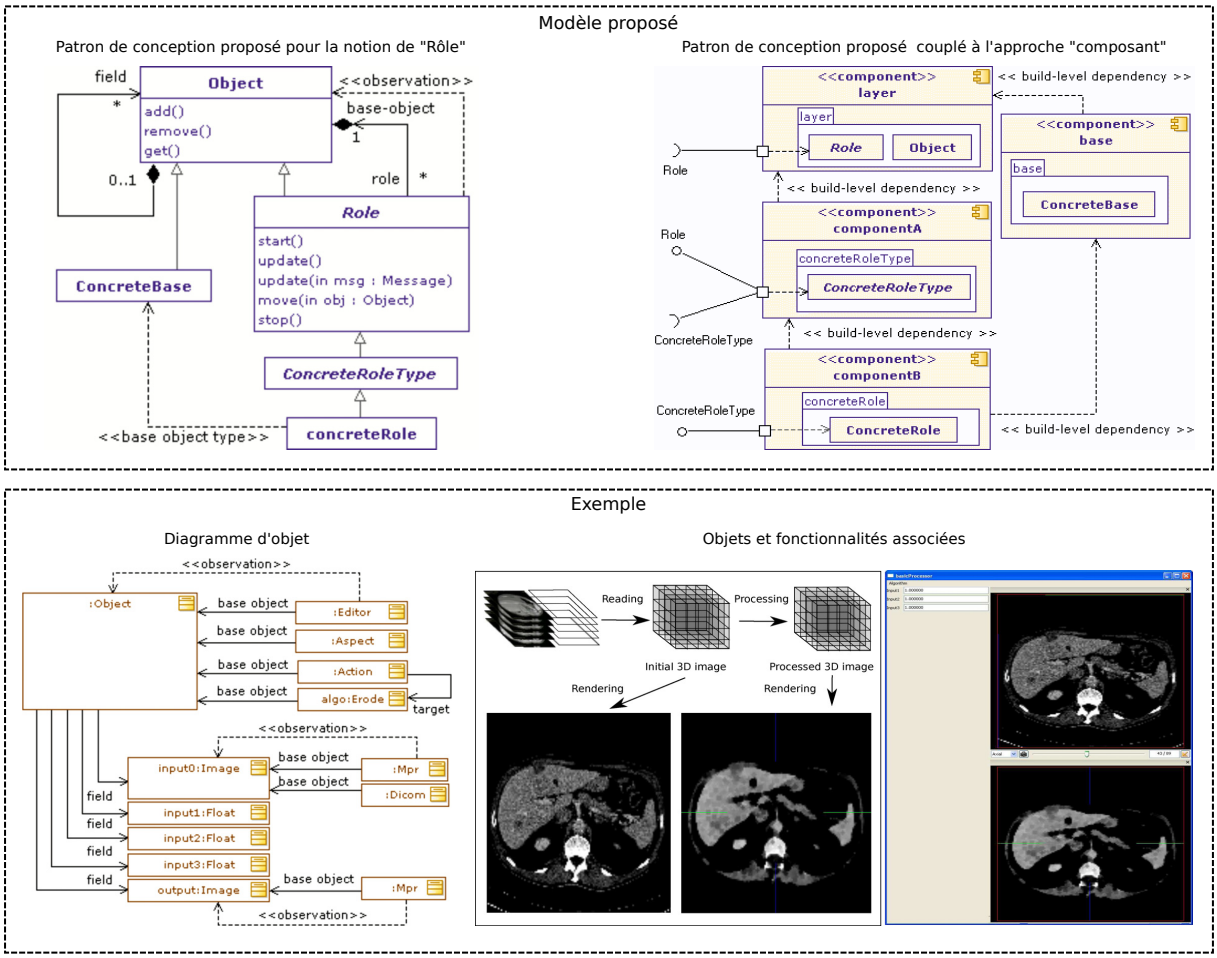

Figure 3.8 : Modèle proposé pour la fabrique de logiciels médicaux. Ce modèle intègre le paradigme de "rôle" (modélisation en haut à gauche), couplé à celui de composant (modélisation en haut à droite). Un exemple est donné, montrant le lien entre les entités et une fonctionnalité de traitement d'images, couplée à la lecture de fichiers DICOM, à une interface graphique et à de la visualisation 3D. Cette figure est associée à la publication [55].

pour érosion mathématique), visualisation 3D (`:Mpr` - pour rendu multiplanaire) et interface graphique pour l'édition (e.g. `:Editor` pour éditer les paramètres). Chaque fonctionnalité est fournie par un composant, afin de disposer de dépendances limitées et maîtrisées. Cette notion de rôle peut s'apparenter à de l'héritage (paradigme objet classique) que l'on pourrait déclarer dynamiquement. Dans l'exemple mentionné ci-dessus, au lieu d'avoir la fonctionnalité de rendu 3D déclarée par exemple par héritage statique (e.g. classe image qui implémenterait une méthode de rendu), cette fonctionnalité est attachée dynamiquement à l'objet de type `Image`. Au delà de cet aspect s'apparantant à de l'héritage dynamique, d'autres propriétés ont été considérées. Ces propriétés sont spécifiques au concept de rôle proposé dans la littérature. On peut par exemple citer les propriétés de multiplicité, de transitivité, de migration, d'activation, et de cycle de vie contraint (on pourra consulter pour les détails [55]). Ces propriétés sont notamment gérées au moyen de l'interface qu'un rôle doit implémenter (voir les méthodes de la classe `Role` modélisées dans la figure 3.8 haut-gauche).

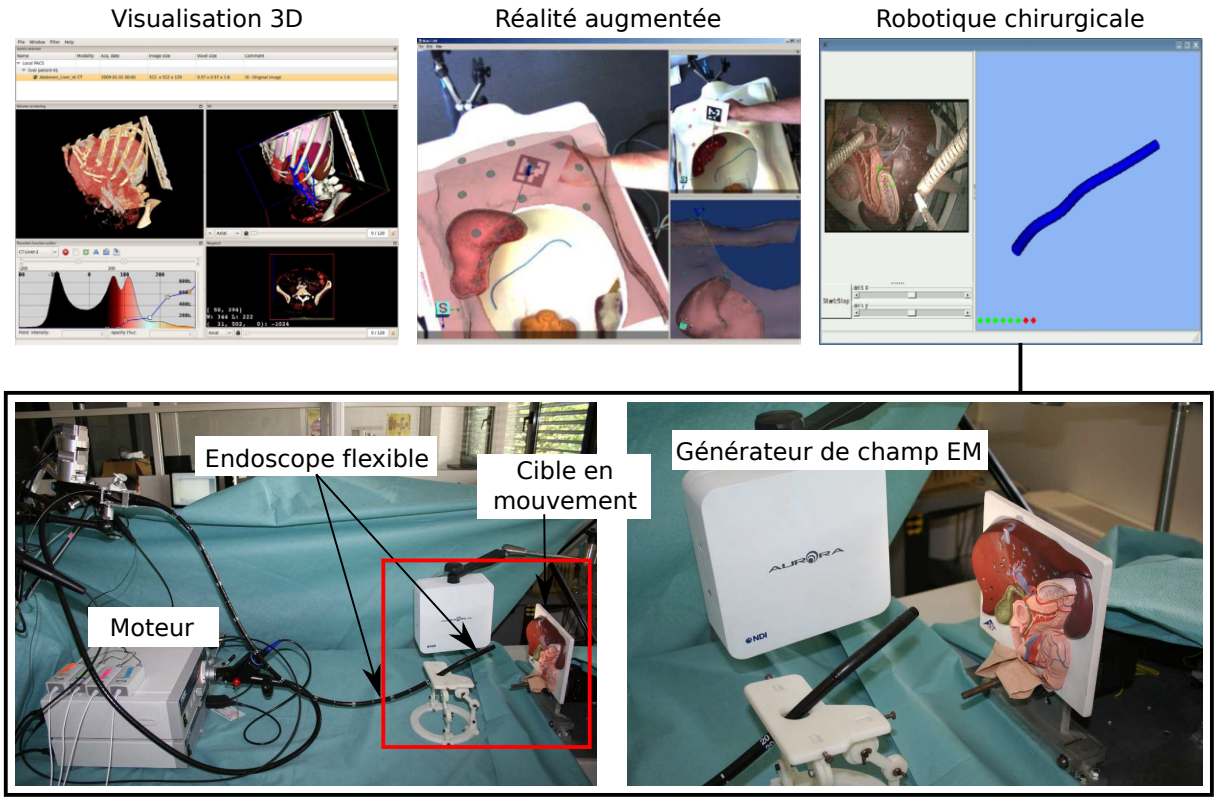

Figure 3.9 : Quelques exemples d'applications élémentaires pour évaluer la pertinence du modèle d'architecture proposé pour la famille de logiciels considérée. Cette figure est associée aux publications [55, 57, 49].

Ce travail a fait l'objet d'une publication dans un journal de la communauté dédiée au génie logiciel [55]. Il a également constitué le socle logiciel du projet opensource fw4spl[3], ouvert en 2009 et utilisé par la société Visible Patient (société issue de l'IRCAD). La pertinence de cette architecture a été validée en construisant des applications élémentaires autour du traitement d'images pour la modélisation du patient (exemple fourni par la figure 3.8-bas), de la visualisation d'images médicales (figure 3.9-gauche), de la réalité augmentée (figure 3.9-centre) et de la robotique chirurgicale (figure 3.9-droite). La preuve de concept autour de la réalité augmentée a été effectuée à l'occasion de l'encadrement d'un étudiant de Master, Julien Waechter, et a fait l'objet d'une communication internationale [57]. La preuve de concept dédiée à la robotique chirurgicale a été réalisée, en collaboration avec l'équipe robotique du laboratoire ICube, au cours de l'encadrement d'un autre étudiant de Master, Guillaume Chabre, et a également fait l'objet d'une communication internationale [49]. Il s'agissait de superviser une fonctionnalité de filtrage des mouvements dus aux battements cardiaques (on pourra consulter [49] pour les détails).

3.3.4 Objets connectés en santé

Cette problématique du logiciel est particulièrement d'actualité avec le développement de l'internet des objets ou des objets connectés [25, 64], en particulier en santé [134]. Dans ce

3. http ://fw4spl-org.github.io/

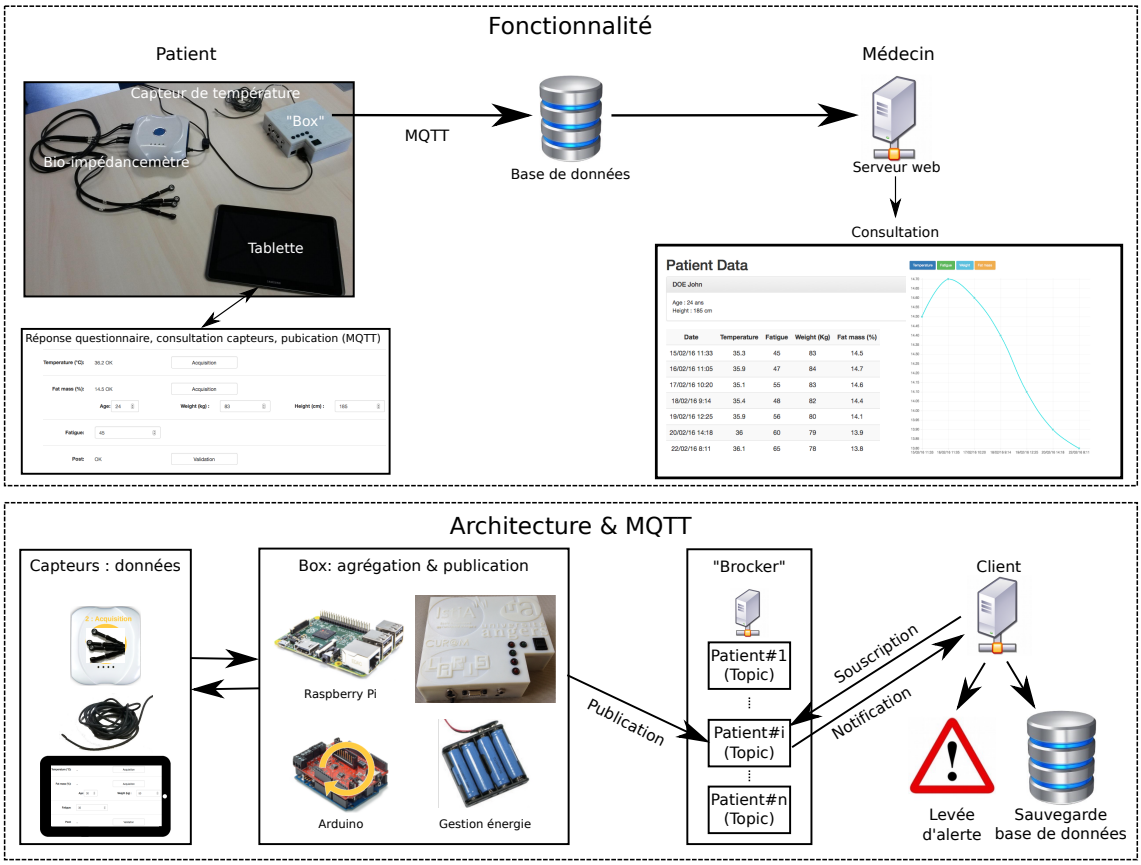

Figure 3.10 : Système "IoT" développé pour l'acquisition de données en utilisant le protocole de communication MQTT ("Message Queueing Telemetry Transport"), de type "publish/subscribe". Haut : le patient remonte des données (capteurs et questionnaire) consultable à distance par le médecin. Bas : le client souscrit un abonnement auprès du "brocker" (serveur intermédiaire) pour recevoir les données publiées par le patient identifié par $\#i$. A chaque publication par la "box", le client est notifié : il peut sauvergarder les données, consultable ultérieurement, et/ou lever une alerte. Cette figure est associée aux communications [75, 76].

contexte, l'enjeu est de concevoir des applications distribuées (e.g. entités d'acquisition, de stockage et de traitement [64]) intégrant des données et capteurs hétérogènes, par exemple pour le suivi de patients à distance [100, 134]. Comme ceci est souligné dans le chapitre, en santé, les informations et dispositifs associés peuvent être multiples : on peut par exemple citer les caméras [69, 100] et les dispositifs fournissant des données physiologiques (saturation du sang en oxygène, température, ECG, poids,...) [134], ou encore des informations telles que la géolocalisation de l'individu (e.g. GPS) ainsi que ses déplacements (e.g. accéléromètre pour la détection de chute [108]). En terme d'implémentation, une problématique sous-jacente importante concerne l'architecture logicielle relative à la remontée de cette information. Sur ce point, on peut citer quelques publications récentes. Par exemple, Barata [11] a récemment présenté un système d'acquisition de quelques données physiologiques (oximétrie et ECG), de transmission, de stockage et de visualisation de ces données. Cette proposition repose sur l'exploitation de différentes technologies telles que le smartphone, la communication bluetooth et le protocole MQTT ("Message Queueing Telemetry Transport"), ce protocole implémentant le modèle "publish/subscribe" cou-

ramment utilisé dans le domaine des objets connectés [126]. Une récente étude a été publiée dans le cas de la supervision d'exercices de rééducation respiratoire (capteurs de suivi des battements cardiaques et de la saturation du sang en oxygène). Indépendamment de l'application médicale visée, l'enjeu de cette étude était de comparer différents protocoles de communication suivant ce modèle "publish/subscribe" [126]. Dans le même esprit, nous avons réalisé un prototype de système distribué utilisant ce protocole de communication MQTT de type "publish/subscribe". Ce travail a fait l'objet de deux communications internationales [75, 76]. La contribution a porté sur la description détaillée du prototype réalisé et de l'architecture mise en œuvre, ainsi que sur les technologies sous-jacentes, permettant de remonter des données depuis le capteur jusqu'à la base de données en utilisant ce protocole. Les capteurs considérés pour cette preuve de concept ont été un capteur de température cutanée et un bio-impédancemètre. A ces données ont été ajoutées celles issues d'un formulaire renseigné par le patient au moyen d'une tablette, ces informations étant remontées simultanément aux données physiologiques précédentes (voir la figure 3.10-haut). Ceci a été l'occasion de montrer comment réaliser ce type de système distribué et hétérogène autour de l'IoT, incluant les différents éléments constitutifs de l'architecture ainsi que quelques outils permettant son implémentation. La figure 3.10-bas donne quelques éléments permettant d'interagir avec les capteurs, la tablette et de publier les données, avec la capacité à lever une alerte dès réception de celles-ci. L'objectif était davantage d'évaluer la faisabilité de la mise en œuvre de ce type de système, que de viser une application médicale particulière. Ce travail a été réalisé lors de l'encadrement de projets d'étudiants en école d'ingénieurs (niveau bac + 4 et 5), et également dans le cadre du master recherche d'Antoine Jamin [76]. Ce travail a été réalisé en collaboration avec Mehdi Lhommeau de l'université d'Angers, avec des médecins du CHU d'Angers et de l'ICO d'Angers (Institut de Cancérologie de l'Ouest), et enfin en partenariat avec l'entreprise Bioparhom (pour les calculs de bio-impédancemétrie à partir de leur dispositif).

3.4 Conclusion

En conclusion, on constate qu'il y a une activité importante, attestée par de récentes publications, autour de cette problématique de l'implémentation, à différentes échelles (i.e. du matériel au logiciel), y compris dans le domaine émergent des objets connectés. Comme ceci a été illustré dans ce chapitre, cette problématique peut-être abordée au niveau matériel, algorithmique ou encore logiciel selon différents angles : adaptation des algorithmes aux architectures, reformulation des méthodes pour réduire la complexité, exploitation des mécanismes et fonctionnalités fournis par les langages et librairies associées, ou encore appropriation et adaptation des concepts empruntés au monde du génie logiciel.

Les enjeux sous-jacents de performance, généricité et simplicité, sont des sujets importants et toujours d'actualité. Ces travaux peuvent apparaître comme étant du ressort de l'ingénierie davantage que de celui de la recherche académique. Ils reflètent néanmoins une problématique importante pour la communauté, comme l'attestent les publications. Ils permettent de faciliter le partage et le développement d'algorithmes, ainsi que la mise en œuvre de recherches tournées vers les applications, pour lesquelles des outils spécifiques doivent être conçus.

Il s'agit d'une problématique complémentaire aux aspects "méthodes" présentés dans le chapitre 2, et à laquelle j'ai eu l'occasion de contribuer soit directement, soit par l'intermédiaire d'encadrements d'étudiants en master recherche ou thèse.

Chapitre 4

Conclusion et perspectives

4.1 Conclusion

Le chapitre 1 synthétise mon activité d'enseignement et de recherche depuis l'obtention de mon doctorat en 2002. Une particularité de mon parcours concerne les deux environnements au sein desquels j'ai travaillé : tout d'abord dans une structure privée dédiée à la recherche appliquée et au développement puis dans une structure publique universitaire. Ce parcours atypique a été riche par la diversité des activités (enseignement, recherche, ingénierie, valorisation industrielle, ...) et des collaborations. Ceci s'est traduit par des contributions dans deux domaines relativement distincts mais aussi complémentaires sous certains aspects.

Le chapitre 2 est dédié à mon activité de recherche et d'encadrement appliquée au traitement d'images principalement. J'ai présenté mes différentes contributions en deux parties. La première partie porte sur un axe que l'on peut qualifier de "théorique", autour de l'exploitation de connaissances structurelles qualitatives a priori, en s'appuyant sur des méthodes basées sur la théorie des graphes. La seconde partie se focalise sur des travaux appliqués, principalement à la santé, dans le domaine de l'analyse d'images médicales pour l'aide au diagnostic, et, plus récemment, dans le domaine des objets connectés en santé.

Le chapitre 3 est dédié à la problématique de l'implémentation de méthodes de traitement et de fonctionnalités connexes, principalement autour de l'imagerie médicale. Ce chapitre souligne l'importance de cette problématique, y compris dans la communauté signal-image, abordée essentiellement autour des notions de performance et de généricité, avec les considérations technologiques sous-jacentes (e.g. outils dans le domaine des objets connectés). Mes contributions sur ce point sont étroitement liées à mon activité de développement et d'ingénierie sur la période 2002-2009.

4.2 Perspectives

Au moment de l'écriture de ce document, mon activité s'articule autour de deux axes que je compte privilégier. Le premier axe concerne un aspect que je qualifierais de "théorique", porte sur l'usage des connaissances structurelles qualitatives pour l'interprétation des images (section 4.2.1). Le second axe concerne une recherche plus appliquée, principalement autour des objets connectés en santé (section 4.2.2). Les perspectives autour de ces deux axes s'apparentent à un projet de recherche, sachant que sa mise en œuvre est nécessairement conditionnée par les opportunités et les contraintes notamment liées à l'environnement.

4.2.1 Connaissances structurelles qualitatives pour l'interprétation d'images

Cet axe se place dans la continuité de ce qui a été présenté dans le chapitre 2. L'objectif est d'apporter des éléments de réponse à la question générale suivante : Comment, et pour quels bénéfices, exploiter la description qualitative d'une image pour en guider son analyse ?

Sur la base de mes récentes contributions et de la littérature sur le sujet, plusieurs pistes d'investigation semblent pertinentes. Ces pistes sont associées à la figure 4.1, expliquée par la suite.

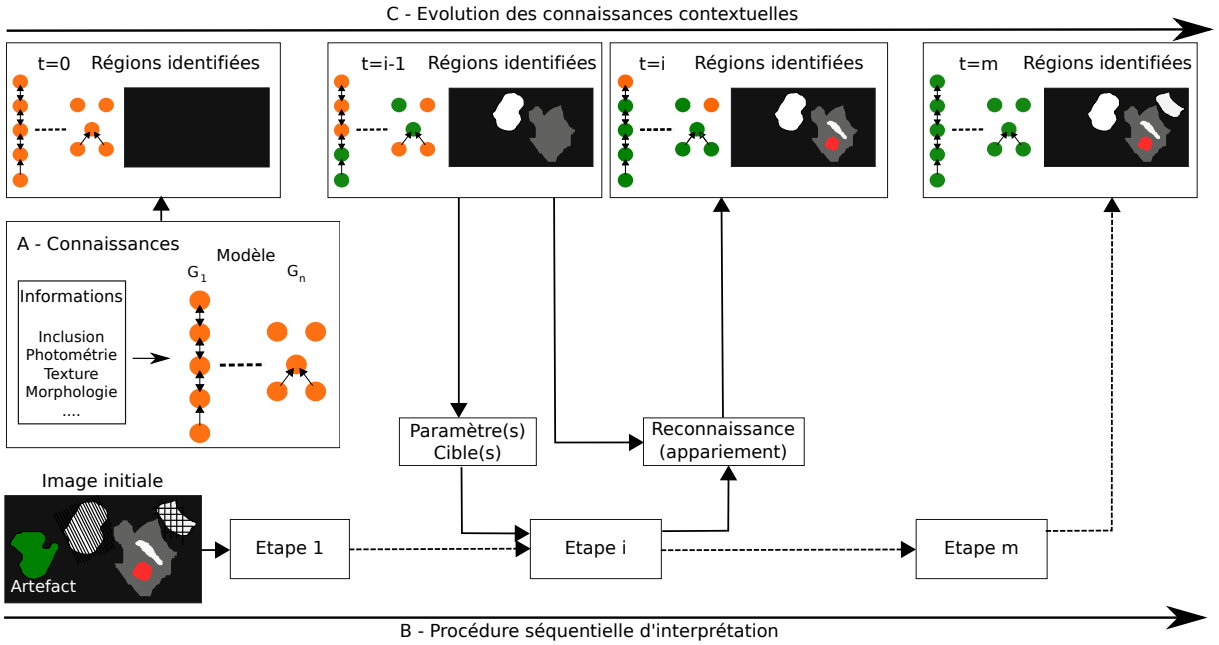

Figure 4.1 : Connaissances a priori qualitatives et perspectives : vue d'ensemble. **A** : les connaissances a priori peuvent correspondre à diverses relations entre les régions (graphes G_1 à G_n), d'inclusion, de photométrie, de texture, de morphologie... Dans cet exemple, G_1 correspond à la photométrie, et plus précisément à la luminosité ("valeur" dans le système HSV), avec plusieurs régions similaires. G_n modélise les relations d'inclusion. **B** : L'image initiale, composée d'un artefact, de deux régions texturées et d'une région colorée, est analysée en plus étapes. A une étape donnée (e.g. étape "i"), on peut s'appuyer sur les connaissances contextuelles pour décider de la cible (ou des cibles), des paramètres des algorithmes utilisés durant cette étape et enfin pour reconnaître la région cible (ou les régions cibles) en fin d'étape. **C** : Evolution des connaissances, initialisées par le modèle a priori : les régions sont progressivement identifiées (nœuds en vert). A noter qu'une approche globale peut être vue comme une approche séquentielle à une étape (e.g. étape "i" avec reconnaissance seule).

A court terme, une voie d'approfondissement que me semble importante concerne la contrainte forte imposée sur les relations d'inclusion, considérée dans mes travaux récents [52]. En effet, contrairement aux relations photométriques pour lesquelles la notion de similarité permet de gérer des relations d'ordre incertaines, notre approche ne tolère pas qu'une relation d'inclusion déclarée dans le modèle ne soit pas vérifiée dans l'image en cours d'analyse. Cette limitation

est partiellement contournée par fusion progressive des régions adjacentes jusqu'à ce que les relations d'inclusion soient vérifiées, permettant d'obtenir au moins un appariement. De manière similaire aux relations photométriques, une alternative consisterait à déclarer des relations d'inclusion incertaines au niveau du modèle. Une autre alternative serait de quantifier les relations d'inclusion construites à partir de l'image en cours d'analyse (e.g. degré d'inclusion d'une structure dans une autre). Comme pour l'information photométrique, ceci induirait la définition d'un critère permettant de choisir le "meilleur" isomorphisme en s'appuyant sur cette quantification.

A moyen terme, des pistes d'investigation qui semblent pertinentes et que je compte prioritairement étudier concernent la nature de l'information a priori, le guidage des algorithmes et la stratégie de segmentation.

Au sujet de la nature de l'information a priori, une perspective peut concerner l'intégration de nouvelles connaissances, en complément des relations photométriques et d'inclusion que j'ai déjà pu considérer, en privilégiant une description qualitative (graphes G_1 et G_n dans la figure 4.1-A). Ceci peut concerner des connaissances associées à des travaux connexes, par exemple autour des positions relatives. Il peut également s'agir d'informations associées à d'autres propriétés, par exemple la texture et la morphologie. Ces informations correspondent à l'ensemble des n graphes G_1 à G_n représentés dans la figure 4.1-A. Pour reconnaître les différentes régions en fin d'analyse ("reconnaissance" dans la figure 4.1), la difficulté sera de les intégrer de manière homogène avec celles déjà considérées (e.g. recherche des isomorphismes communs) et de définir des critères de décision adaptés (e.g. cas du critère, basé sur l'information photométrique, permettant de choisir un isomorphisme parmi l'ensemble des isomorphismes communs).

Concernant le guidage d'algorithmes ("paramètre(s)" dans la figure 4.1), il reste à ma connaissance, et compte tenu de la multitude d'algorithmes de segmentation existants, de nombreuses pistes à étudier, en complément des travaux mentionnés dans le chapitre 2. Par exemple, nous nous sommes récemment intéressés au guidage de l'algorithme des k-means [51]. Il serait intéressant d'étendre nos travaux à d'autres algorithmes de "clustering", tel que, pour ne citer que lui, l'algorithme "meanshift". Comme pour les k-means, l'enjeu serait de déterminer les relations entre la connaissance "haut-niveau" et les paramètres "bas-niveau" de l'algorithme.

Dans le contexte d'une approche de traitement séquentielle, la stratégie de segmentation est un point important, comme ceci a été souligné dans le chapitre 2 en citant les travaux de G. Fouquier [59]. Il s'agit d'un problème auquel je n'ai pas directement contribué, mais qui s'avère connexe à mes travaux sur le guidage d'algorithmes. Une piste d'investigation serait de s'appuyer sur les relations, non seulement spatiales mais également photométriques (voire, par exemple, texturales, selon la nature des relations a priori connues) pour guider, à une étape donnée, le choix de la structure à segmenter à l'étape suivante ("cible(s)" dans la figure 4.1), tout en vérifiant ensuite que la cohérence d'ensemble a bien été préservée.

Un point important que je souhaite développer concerne l'utilisation de cette approche pour différentes applications afin d'en évaluer sa portée, et sa capacité à résoudre des problèmes concrets.

Sur cette thématique, mon objectif est également de porter une attention particulière à

l'implémentation de ces méthodes, à l'instar de la librairie "scikit-gtimage" [1] associée à notre récente contribution [52], afin de fournir à la communauté des outils, conçus de manière générique, permettant de faciliter le partage, l'évaluation et l'extension de ces travaux.

4.2.2 Objets connectés en santé

L'axe "applicatif" que je privilégie actuellement et que j'envisage de développer concerne essentiellement les objets connectés pour la santé. Comme ceci a été souligné en fin de chapitre 2, il s'agit d'un thème en plein essor. S'agissant par ailleurs d'un thème pluridisciplinaire (e.g. médecine, traitement du signal, des images et plus généralement des données), mon activité sur ce domaine se fera en collaboration avec des personnels de santé et des collègues spécialisés dans les STIC (sciences et technologies de l'information et de la communication). Cela nécessite une part d'adaptation afin d'élargir mon socle de connaissances ainsi que de transposer à ce contexte les outils et méthodes que je pratique ou ai pratiqué en traitement d'images (e.g. outils relatifs à l'apprentissage automatique que l'on rencontre couramment en traitement des images). Comme ceci est souligné par la récente littérature sur le domaine, il est à noter que la composante "image" peut tout à fait avoir sa place, dans ce contexte de "l'internet des objets" où les informations sont de nature fortement hétérogènes. Cette activité vient en complément de l'axe plus "théorique" présenté précédemment, ce dernier présentant l'avantage, par rapport à cet axe applicatif, de réduire les risques de blocages inhérents à cette recherche pluridisciplinaire en santé (e.g. dépendance aux données, contraintes règlementaires).

La question générale est : Comment, et pour quels bénéfices, les objets connectés peuvent améliorer la santé des individus ? Le "comment" sous-entend le choix des informations remontées et la manière de les exploiter. La seconde partie de la question porte sur les bénéfices d'un tel suivi, par rapport à un suivi classique (e.g. consultations espacées). La figure 4.2 illustre, dans un cas particulier, ce lien entre "objets connectés" et médecine "classique". Cet exemple souligne également le besoin de relier les informations médicales acquises selon des procédures traditionnelles (une image médicale dans cet exemple) et les informations que l'on serait à même de pouvoir collecter de manière distante et plus régulière.

Comme ceci a été précisé dans le chapitre 2, hormis les travaux précédents autour d'un système d'agrégation de données [76] et d'un tensiomètre connecté [74], je participe actuellement à des travaux sur le sujet autour de deux applications particulières. Ces deux applications constituent mon projet de recherche appliqué à moyen terme.

L'application principale concerne la prévention des accidents cardiovasculaires à l'aide d'un bracelet connecté, suite à une collaboration que j'ai établie avec la société Axelife. Ce travail de recherche est mené par Franck Mouney, qui vient de débuter sa thèse au LARIS, thèse que je co-encadre avec Mickaël Dinomais (LARIS et CHU d'Angers) et Teodor Tiplica (LARIS).

La seconde application porte sur l'usage d'un vélo connecté pour la détection des troubles cognitifs dans le cadre de la maladie d'Alzheimer, à l'occasion d'un partenariat établi entre le LARIS (Anne Humeau-Heurtier), le CHU d'Angers et la société Cottos. Il s'agit d'un projet dans lequel je m'investis, à l'heure actuelle, dans une moindre mesure, à l'occasion de l'encadrement du master recherche de Martin Schmidt.

A noter que ces perspectives de contributions autour des objets connectés n'excluent pas

1. https ://github.com/jbfasquel/skgtimage

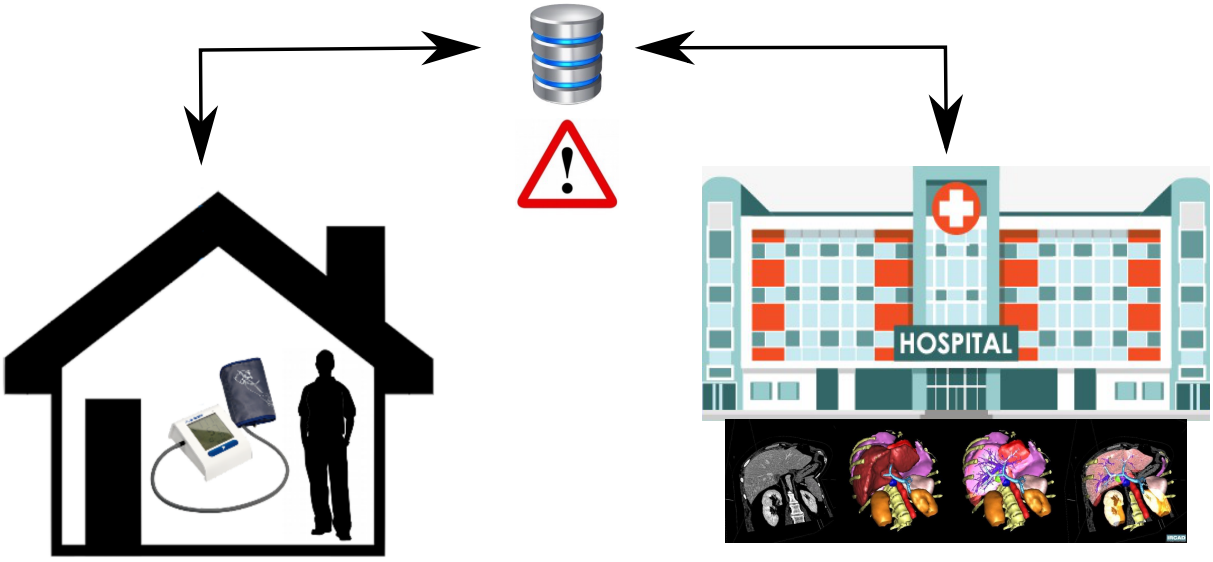

Figure 4.2 : Objet connecté (à gauche) et pratique médicale traditionnelle (à droite). Cet exemple implique un tensiomètre connecté permettant de remonter une information (entité de stockage au centre) complémentaire l'information que l'on pourrait obtenir par imagerie médicale. L'utilisation conjointe de ces informations pourrait permettre d'améliorer le suivi, notamment en permettant d'identifier des situations à risque (levée d'alerte au centre).

nécessairement des applications liées à l'imagerie médicale à l'occasion de collaborations ponctuelles, potentiellement en lien avec l'axe "théorique" précédent.

Par ailleurs, comme ceci est illustré par la figure 4.2, il est raisonnable de penser, qu'à terme, il sera possible et nécessaire de fusionner ces deux sources d'information (informations issues des objets connectés et des pratiques médicales classiques) afin de disposer d'un outil de suivi plus complet.

Comme pour l'axe précédent, une attention particulière sera apportée à la problématique de l'implémentation, ceci constituant un enjeu important faisant régulièrement l'objet de publications à destination de la communauté scientifique en STIC, visant à fournir des algorithmes fonctionnels et/ou à expliquer/contextualiser/proposer des architectures et des technologies. Il s'agit également d'une problématique souvent omniprésente pour répondre à divers besoins ponctuels (e.g. outil logiciel pour des médecins dans le cadre de travaux collaboratifs). Pour cette raison, même s'il s'agira de contributions marginales, je compte continuer à exploiter cette expérience au service de ce domaine d'application, dans l'esprit de notre dernière communication sur le sujet [76].

Bibliographie

[1] *Software Product Lines : Practices and Patterns*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[2] Z. Abu-Aisheh. *Approches anytime et distribuées pour l'appariement de graphes*. PhD thesis, 2016. Thèse de doctorat dirigée par Ramel, Jean-Yves Informatique Tours 2016.

[3] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11) :2274–2282, 2012.

[4] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah. Time-series clustering – a decade review. *Information Systems*, 53 :16 – 38, 2015.

[5] M. U. Ahmed, S. Begum, and J.-B. Fasquel, editors. *Internet of Things (IoT) Technologies for HealthCare, Proceedings of the 4th International Conference, HealthyIoT 2017, Angers, France, October 24-25, 2017*. Springer, 2018.

[6] J. Akeret, L. Gamper, A. Amara, and A. Refregier. Hope : A python just-in-time compiler for astrophysical computations. *Astronomy and Computing*, 10 :1 – 8, 2015.

[7] A. Alfalou and C. Brosseau. Chapter two - recent advances in optical image processing. volume 60 of *Progress in Optics*, pages 119 – 262. Elsevier, 2015.

[8] R. Ambauen, S. Fischer, and Horst Bunke. Graph edit distance with node splitting and merging, and its application to diatom identification. In *Proceedings of the 4th IAPR International Conference on Graph Based Representations in Pattern Recognition*, GbRPR'03, pages 95–106, Berlin, Heidelberg, 2003. Springer-Verlag.

[9] J. Andreu-Perez, D.R. Leff, H.M.D. Ip, and Guang-Zhong Yang. From wearable sensors to smart implants- toward pervasive and personalized healthcare. *Biomedical Engineering, IEEE Transactions on*, 62(12) :2750–2762, 2015.

[10] Z. Bankó and J. Abonyi. Mixed dissimilarity measure for piecewise linear approximation based time series applications. *Expert Systems with Applications*, 42(21) :7664 – 7675, 2015.

[11] D. Barata, G. Louzada, A. Carreiro, and A. Damasceno. System of acquisition, transmission, storage and visualization of pulse oximeter and ECG data using android and MQTT. *Procedia Technology*, 9 :1265 – 1272, 2013.

[12] T. Bärecke. *Isomorphisme Inexact de Graphes par Optimisation Évolutionnaire*. PhD thesis, Pierre and Marie Curie University, Paris, France, 2009.

[13] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3) :346 – 359, 2008.

[14] E. Bengoetxea. *Inexact Graph Matching Using Estimation of Distribution Algorithms*. PhD thesis, 2002. Ecole Nationale Superieure des Telécommunications (ENST).

[15] S. Berretti, A. Del Bimbo, and E. Vicario. Efficient matching and indexing of graph models in content-based retrieval. *IEEE Transactions on pattern analysis and machine intelligence*, 23(10) :1089 –1105, 2001.

[16] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms. *Communications on Pure and Applied Mathematics*, 44(2) :141–183, 1991.

[17] W. Birkfellner, K. Huber, A. Larson, D. Hanson, M. Diemling, P. Homolka, and H. Bergmann. A modular software system for computer-aided surgery and its first application in oral implantology. *IEEE Transactions on Medical Imaging*, 19(6) :616–620, 2000.

[18] I. Bitter, R. Van Uitert, I. Wolf, L. Ibanez, and J. M. Kuhnigk. Comparison of four freely available frameworks for image processing and visualization that use ITK. *IEEE Transactions on Visualization and Computer Graphics*, 13(3) :483–493, 2007.

[19] M. D. Blackledge, D. J. Collins, D.-M. Koh, and M. O. Leach. Rapid development of image analysis research tools : Bridging the gap between researcher and clinician with pyOsiriX. *Computers in Biology and Medicine*, 69 :203 – 212, 2016.

[20] I. Bloch. Spatial reasoning under imprecision using fuzzy set theory, formal logics and mathematical morphology. *International Journal of Approximate Reasoning*, 41(2) :77–95, 2006.

[21] I. Bloch. Fuzzy sets for image processing and understanding. *Fuzzy Sets and Systems*, 281 :280–291, 2015.

[22] P. Le Bodic, P. Héroux, S. Adam, and Y. Lecourtier. An integer linear program for substitution-tolerant subgraph isomorphism and its use for symbol spotting in technical drawings. *Pattern Recognition*, 45(12) :4214 – 4224, 2012.

[23] M. C. Boeres, C. C. Ribeiro, and I. Bloch. *A Randomized Heuristic for Scene Recognition by Graph Matching*, pages 100–113. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[24] G. Booch. Handbook of software architecture. *http ://handbookofsoftwarearchitecture.com/*, 2006.

[25] E. Borgia. The internet of things vision : Key features, applications and open issues. *Computer Communications*, 54 :1 – 31, 2014.

[26] F. Bouzidi, E. Charfi, F. Ghozzi, and A. Fakhfakh. Implementation of optical correlator for face recognition applications. In *International Conference on Computer Vision and Image Analysis Applications*, pages 1–5, 2015.

[27] E. O. Brigham and R. E. Morrow. The fast fourier transform. *IEEE Spectrum*, 4(12) :63–70, 1967.

[28] O. Camara, O. Colliot, and I. Bloch. Computational modeling of thoracic and abdominal anatomy using spatial relationships for image segmentation. *Real-Time imaging*, 10 :263–273, 2004.

[29] J. Caroff, L. Bière, G. Trebuchet, C. Nedelcu, E. Sibileau, J.-P. Beregi, C. Aubé, A. Furber, and S. Willoteaux. Applications des séquences de vélocimétrie par contraste de phase en imagerie cardiovasculaire. *Journal de Radiologie Diagnostique et Interventionnelle*, 93(3) :170 – 182, 2012.

[30] R.M. Cesar, E. Bengoetxea, I. Bloch, and P. Larrañaga. Inexact graph matching for model-based recognition : Evaluation and comparison of optimization algorithms. *Pattern Recognition*, 38(11) :2099 – 2113, 2005.

[31] P.-A. Champin and C. Solnon. Measuring the similarity of labeled graphs. In *Proceedings of the 5th International Conference on Case-based Reasoning : Research and Development*, ICCBR'03, pages 80–95, Berlin, Heidelberg, 2003. Springer-Verlag.

[32] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2) :266–277, 2001.

[33] O. Colliot, O. Camara, and I. Bloch. Integration of fuzzy spatial relations in deformable models - application to brain MRI segmentation. *Pattern Recognition*, 39 :1401–1414, 2006.

[34] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty Years Of Graph Matching In Pattern Recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(3) :265–298, 2004.

[35] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19 :297–301, 1965.

[36] K Czarnecki and U. Eisenecker. *Generative programming : methods, techniques and applications*. Addison-Wesley, 1999.

[37] D. Coimbra de Andrade and L. G. Trabasso. An opencl framework for high performance extraction of image features. *Journal of Parallel and Distributed Computing*, 109 :75 – 88, 2017.

[38] O. Déforges, N. Normand, and M. Babel. Fast recursive grayscale morphology operators : from the algorithm to the pipeline architecture. *Journal of Real-Time Image Processing*, 8(2) :143–152, 2013.

[39] A. Deruyver and Y. Hodé. Qualitative spatial relationships for image interpretation by using a conceptual graph. *Image and Vision Computing*, 27 :876–886, 2009.

[40] A. Deruyver, Y. Hodé, and L. Brun. Image interpretation with a conceptual graph : Labeling over-segmented images and detection of unexpected objects. *Artificial Intelligence*, 173 :1245–1265, 2009.

[41] B. Deville. Création d'un prototype de segmentation interactive 3D par modèles déformables. Master's thesis, Master in Computer Science, University of Reims, `PDF:http://perso-laris.univ-angers.fr/~fasquel/research.html`, 2007.

[42] C. Domínguez, J. Heras, and V. Pascual. IJ-OpenCV : Combining ImageJ and OpenCV for processing images in biomedicine. *Computers in Biology and Medicine*, 84 :189 – 194, 2017.

[43] A. Eklund, P. Dufort, D. Forsberg, and S. M. LaConte. Medical image processing on the GPU – past, present and future. *Medical Image Analysis*, 17(8) :1073 – 1094, 2013.

[44] J.-B. Fasquel, V. Agnus, and J. Lamy. An efficient and generic extension to ITK to process arbitrary shaped regions of interest. *Computer Methods and Programs in Biomedicine*, 81(1), 2006.

[45] J.-B. Fasquel, V. Agnus, J. Moreau, L. Soler, and J. Marescaux. An interactive medical image segmentation system based on the optimal management of regions of interest using topological medical knowledge. *Computer Methods and Programs in Biomedicine*, 82 :216–230, 2006.

[46] J.-B. Fasquel, G. Brocker, J. Moreau, N. Papier, V. Agnus, C. Koehl, L. Soler, and J. Marescaux. A modular and evolutive component oriented software architecture for patient modeling. *Computer Methods and Programs in Biomedicine*, 83(3) :222 – 233, 2006.

[47] J.-B. Fasquel and M. Bruynooghe. New hybrid opto-electronic method for fast and unsupervised object detection. *Optical Engineering*, 42 :42 – 42 – 13, 2003.

[48] J.-B. Fasquel and M. Bruynooghe. A hybrid opto-electronic method for fast off-line handwritten signature verification. *Document Analysis and Recognition*, 7(1) :56–68, 2004.

[49] J.-B. Fasquel, G. Chabre, P. Zanne, S. Nicolau, V. Agnus, L. Soler, M. De Mathelin, and J. Marescaux. A role-based component architecture for computer assisted interventions : illustration for electromagnetic tracking and robotized motion rejection in flexible endoscopy. In *Workshop on Systems Architecture for Computer Assisted Intervention, the 12th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), London, UK*, 2009.

[50] J.-B. Fasquel and N. Delanoue. Exploitation des relations qualitatives topologiques et photométriques a priori pour le seuillage d'images en niveaux de gris par k-means. In *Colloque Gretsi*, 2013.

[51] J.-B. Fasquel and N. Delanoue. An approach for sequential image interpretation using a priori binary perceptual topological and photometric knowledge and k-means based segmentation. *Journal of the Optical Society of America A. To appear*, 2018.

[52] J.-B. Fasquel and N. Delanoue. A graph based image interpretation method using a priori qualitative inclusion and photometric relationships. *IEEE Transactions on Pattern Analysis and Machine Intelligence. To appear*, 2018.

[53] J.-B. Fasquel and C. Jeanguillaume. Python en signal-image : éléments sur le langage, les outils, l'architecture, les fonctionnalités et la communauté. In *Colloque Gretsi*, 2015.

[54] J.-B. Fasquel, A. Lécluse, C. Cavaro-Ménard, and S. Willoteaux. A semi-automated method for measuring the evolution of both lumen area and blood flow in carotid from phase contrast MRI. *Computers In Biology And Medicine*, 66, 2015.

[55] J.-B. Fasquel and J. Moreau. A design pattern coupling role and component concepts : Application to medical software. *The Journal of Systems and Software*, 84 :847–863, 2011.

[56] J.B. Fasquel and V. Agnus. Improving genericity and performances of medical systems and softwares based on image analysis. In *The 18th IEEE International Symposium on Computer-Based Medical Systems, Dublin, Ireland*, 2005.

[57] J.B. Fasquel, J. Waechter, S. Nicolau, V. Agnus, L. Soler, and J. Marescaux. A XML based component oriented architecture for image guided surgery : illustration for the video based tracking of a surgical tool. In *Workshop on Systems Architecture for Computer Assisted Intervention, the 11th International Conference on Medical Image Computing and Computer Assisted Intervention, New York, USA*, 2008.

[58] P. Foggia, G. Percannella, and M. Vento. Graph matching and learning in pattern recognition in the last 10 years. *International Journal of Pattern Recognition and Artificial Intelligence*, 28(01) :1450001, 2014.

[59] G. Fouquier, J Atif, and I. Bloch. Sequential model-based segmentation and recognition of image structures driven by visual features and spatial relations. *Computer Vision and Image Understanding*, 116 :146–165, 2012.

[60] M.M. Fraz, R.A. Welikala, A.R. Rudnicka, C.G. Owen, D.P. Strachan, and S.A. Barman. Quartz : Quantitative analysis of retinal vessel topology and size – an automated system for quantification of retinal vessels morphology. *Expert Systems with Applications*, 42(20) :7221 – 7234, 2015.

[61] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns : Elements of Reusable Object-oriented Software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.

[62] X. Gao, B. Xiao, D. Tao, and X. Li. A survey of graph edit distance. *Pattern Analysis and Applications*, 13(1) :113–129, 2010.

[63] Farid Garcia-Lamont, Jair Cervantes, Asdrúbal López, and Lisbeth Rodriguez. Segmentation of images by color features : A survey. *Neurocomputing*, 292 :1–27, 2018.

[64] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (iot) : A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7) :1645 – 1660, 2013.

[65] M. Van Herk. A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels. *Pattern Recognition Letters*, 13(7) :517 – 521, 1992.

[66] J.M. Hernández-Ontiveros, E. Inzunza-González, E.E. García-Guerrero, O.R. López-Bonilla, S.O. Infante-Prieto, J.R. Cárdenas-Valdez, and E. Tlelo-Cuautle. Development and implementation of a fish counter by using an embedded system. *Computers and Electronics in Agriculture*, 145 :53 – 62, 2018.

[67] M. Hoarau. Segmentation d'images médicales 3D par contours actifs interactifs. Master's thesis, Telecom-Physique Strasbourg, University of Strasbourg, `PDF:http://perso-laris.univ-angers.fr/~fasquel/research.html`, 2007.

[68] T. Hoegg, C. Koehler, and A. Kolb. Component based data and image processing systems - a conceptual and practical approach. In *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 66–69, 2015.

[69] Y. Z. Hsieh and Y. L. Jeng. Development of home intelligent fall detection iot system based on feedback optical flow convolutional neural network. *IEEE Access*, PP(99) :1–1, 2017.

[70] C. Hudelot, J. Atif, and I. Bloch. Fuzzy spatial relation ontology for image interpretation. *Fuzzy Sets and Systems*, 159 :1929–1951, 2008.

[71] T. Hussain. ViPS : A novel visual processing system architecture for medical imaging. *Biomedical Signal Processing and Control*, 38 :293 – 301, 2017.

[72] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak. The internet of things for health care : A comprehensive survey. *IEEE Access*, 3 :678–708, 2015.

[73] A. Jamin. Objets connectés et suivi à domicile de patients traités pour un cancer. Master's thesis, Master Systèmes Dynamiques et Signaux, Université d'Angers, `PDF:http://perso-laris.univ-angers.fr/~fasquel/research.html`, 2017.

[74] A. Jamin, J.-B. Fasquel, A. Humeau-Heurtier, P. Abraham, G. Leftheriotis, and S. Henni. Characterization of home-acquired blood pressure time series using multiscale entropy for patients treated against kidney cancer. In *Fourth International Conference on IoT Technologies for HealthCare, Angers, France*, 2017.

[75] A. Jamin, J.B. Fasquel, M. Lhommeau, S. Abadie-Lacourtoisie, and G. Leftheriotis S. Henni. An example of homemade box for iot-based in-home health monitoring. In *World Congress of the IFAC, Toulouse, France*, 2017.

[76] A. Jamin, J.B. Fasquel, M. Lhommeau, E. Cornet, S. Abadie-Lacourtoisie, S. Henni, and G. Leftheriotis. An aggregation plateform for iot-based healthcare : Illustration for bioimpedancemetry, temperature and fatigue level monitoring. In *Third International Conference on IoT Technologies for HealthCare, Vasteras, Sweden*, 2016.

[77] R. L. Janiczek, F.H. Epstein, and S.T. Acton. Velocity guided segmentation of phase contrast magnetic resonance angiography. In *Proceedings of the 15th IEEE International Conference on Image Processing*, pages 2264 – 2267, 2008.

[78] H. J. Johnson, M. M. McCormick, and L. Ibanez. *The ITK Software Guide : Introduction and Development Guidelines*. Kitware, Inc., USA, 2015.

[79] P. Junjie, F. Youyi, Z. Xiaofeng, K. Shuai, and W. Xinyu. Implementation of DFT application on ternary optical computer. *Optics Communications*, 410 :424 – 430, 2018.

[80] P. Kalshetti, M. Bundele, P. Rahangdale, D. Jangra, C. Chattopadhyay, G. Harit, and A. Elhence. An interactive medical image segmentation framework using iterative refinement. *Computers in Biology and Medicine*, 83 :22 – 33, 2017.

[81] P. Karasev, I. Kolesov, K. Fritscher, P. Vela, P. Mitchell, and A. Tannenbaum. Interactive medical image segmentation using pde control of active contours. *IEEE Transactions on Medical Imaging*, 32(11) :2127–2139, 2013.

[82] S. Kiss. Débruitage d'images médicales par diffusion anisotrope : un outil interactif de détermination automatique des paramètres. Master's thesis, University of Bordeaux, `PDF:http://perso-laris.univ-angers.fr/~fasquel/research.html`, 2006.

[83] A. Lasso, T. Heffter, A. Rankin, C. Pinter, T. Ungi, and G. Fichtinger. Plus : Open-source toolkit for ultrasound-guided intervention systems. *IEEE Transactions on Biomedical Engineering*, 61(10) :2527–2537, 2014.

[84] V. Leguy. Segmentation interactive d'images médicales par contours interactifs 2D répercutés en 3D. Master's thesis, Master in Mathematics, University of Orléans, `PDF:http://perso-laris.univ-angers.fr/~fasquel/research.html`, 2007.

[85] O. Lézoray and L. Grady. *Image processing and Analysis with Graphs : theory and practice*. CRC Press, 2012.

[86] B. Liu, H. Zhang, S. Hua, Q. Jiang, R. Huang, W. Liu, S. Zhang, B. Zhang, and Z. Yue. An automatic segmentation system of acetabulum in sequential CT images for the personalized artificial femoral head design. *Computer Methods and Programs in Biomedicine*, 127 :318 – 335, 2016.

[87] M. Lopez-Ramirez, L.M. Ledesma-Carrillo, E. Cabal-Yepez, G. Botella, C. Rodriguez-Donate, and Sergio Ledesma. FPGA-based methodology for depth-of-field extension in a single image. *Digital Signal Processing*, 70 :14 – 23, 2017.

[88] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110, 2004.

[89] T. Luo, S. Liu, L. Li, Y. Wang, S. Zhang, T. Chen, Z. Xu, O. Temam, and Y. Chen. Dadiannao : A neural network supercomputer. *IEEE Transactions on Computers*, 66(1) :73–88, 2017.

[90] M. Madec, J-B. Fasquel, and W. Uhring. Processeur opto-électronique de reconstruction de données tomographiques. Dépôt de brevet français FR0608361, 2006. Abandonné.

[91] M. Madec, J.-B. Fasquel, W. Uhring, P. Joffre, and Y. Hervé. Optical implementation of the filtered backprojection algorithm. *Optical Engineering*, 46, 2007.

[92] M. Madec, J.B. Fasquel, W. Uhring, P. Joffre, and Y. Hervé. Optoelectronic implementation of helical cone-beam computed tomography algorithms. *Optical Engineering*, 47, 2008.

[93] M. Madec., E. Hueber, W. Uhring, J.B. Fasquel, and Y. Hervé. Procedures for SLM image quality improvement. In *European Optical Society Annual Meeting, Paris, France*, 2008.

[94] M. Madec., W. Uhring, J.B. Fasquel, P. Joffre, and Y. Hervé. FLC-SLM dynamic improvement with temporal multiplexing : Application to optical image processing. In *Proc. SPIE, Photonics Europe 2006, Strasbourg, France*, pages 390–399, 2006.

[95] M. Madec, W. Uhring, J.B. Fasquel, P. Joffre, and Y. Hervé. Compatibility of temporal multiplexed spatial light modulator with optical image processing. *Optics Communications*, 275 :27–37, 2007.

[96] M. Madec, W. Uhring, E. Hueber, J.-B. Fasquel, J. Bartinger, and Y. Hervé. Methods for improvement of spatial light modulator image rendering. *Optical Engineering*, 48, 2009.

[97] Morgan Madec. *Conception, simulation et réalisation d'un processeur optoélectronique pour la reconstruction d'images médicales*. PhD thesis, 2006. Thèse de doctorat de l'université de Strasbourg.

[98] K. Madi, H. Seba, H. Kheddouci, and O. Barge. A graph-based approach for kite recognition. *Pattern Recognition Letters*, pages –, 2016.

[99] S. E. Mahmoudi, A. Akhondi-Asl, R. Rahmani, S. Faghih-Roohi, V. Taimouri, A. Sabouri, and H. Soltanian-Zadeh. Web-based interactive 2D/3D medical image processing and visualization software. *Computer Methods and Programs in Biomedicine*, 98(2) :172 – 182, 2010.

[100] L. Y. Mano, B. S. Faiçal, L. H.V. Nakamura, P. H. Gomes, G. L. Libralon, R. I. Meneguete, G. P.R. Filho, G. T. Giancristofaro, G. Pessin, B. Krishnamachari, and J. Ueyama. Exploiting IoT technologies for enhancing health smart homes through patient identification and emotion recognition. *Computer Communications*, 89-90 :178 – 190, 2016.

[101] B.B. Mayvan, A. Rasoolzadegan, and Z.G. Yazdi. The state of the art on design patterns : A systematic mapping of the literature. *Journal of Systems and Software*, 125(Supplement C) :93 – 118, 2017.

[102] J. Mille. Narrow band region-based active contours and surfaces for 2D and 3D segmentation. *Computer Vision and Image Understanding*, 113(9) :946 – 965, 2009.

[103] M. S. Monjur, M. F. Fouda, and S. M. Shahriar. All optical three dimensional spatio-temporal correlator for automatic event recognition using a multiphoton atomic system. *Optics Communications*, 381 :418 – 432, 2016.

[104] M. S. Monjur, S. Tseng, R. Tripathi, J.J. Donoghue, and M. S. Shahriar. Hybrid optoelectronic correlator architecture for shift-invariant target recognition. *Journal of the Optical Society of America A*, 31(1) :41–47, 2014.

[105] A. Moreno, C.M. Takemura, O. Colliot, O. Camara, and I. Bloch. Using anatomical knowledge expressed as fuzzy constraints to segment the heart in CT images. *Pattern Recognition*, 41(8) :2525 – 2540, 2008.

[106] F. Mouney. *Analyse et classification de données biomédicales issues de capteurs connectés*. PhD thesis, 2018-2021. Thèse de doctorat de l'université d'Angers en préparation.

[107] O. Nempont, J. Atif, and I. Bloch. A constraint propagation approach to structural model based image segmentation and recognition. *Information Sciences*, 246(0) :1–27, 2013.

[108] L. P. Nguyen, M. Saleh, and R. Le Bouquin Jeannès. An efficient design of a machine learning-based elderly fall detector. In *Fourth International Conference on IoT Technologies for HealthCare, Angers, France - To appear in springer proceedings*, 2017.

[109] A. Noma, A.B.V. Graciano, R.M. Cesar Jr, L.A. Consularo, and I. Bloch. Interactive image segmentation by matching attributed relational graphs. *Pattern Recognition*, 45(3) :1159 – 1179, 2012.

[110] A. Noma, A. Pardo, and R. M. Cesar. Structural matching of 2D electrophoresis gels using deformed graphs. *Pattern Recognition Letters*, 32(1) :3 – 11, 2011.

[111] S.D. Olabarriaga and A.W.M. Smeulders. Interaction in the segmentation of medical images : a survey. *Medical Image Analysis*, 5 :127–142, 2001.

[112] E. Ouabida, A. Essadike, and A. Bouzid. Optical approach for iris segmentation and tracking. In *2016 4th IEEE International Colloquium on Information Science and Technology*, pages 476–480, 2016.

[113] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, 12 :2825–2830, 2011.

[114] F. Perez, B. E. Granger, and J. D. Hunter. Python : An ecosystem for scientific computing. *Computing in Science and Eng.*, 13(2) :13–21, 2011.

[115] J. M. Perkel. Programming : pick up python. *Nature*, 518(7537) :125–126, 2015.

[116] Shari Lawrence Pfleeger. *Software Engineering : Theory and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 4nd edition, 2012.

[117] S M Pincus. Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences*, 88(6) :2297–2301, 1991.

[118] C. Rigaud. Interprétation d'images et graphe conceptuel intégrant des connaissances topologiques et photométriques. Master's thesis, Master Systèmes Dynamiques et Signaux, Université d'Angers, `PDF:http://perso-laris.univ-angers.fr/~fasquel/research.html`, 2011.

[119] F. Serratosa and X. Cortés. Graph edit distance : Moving from global to local structure to solve the graph-matching problem. *Pattern Recognition Letters*, 65 :204 – 210, 2015.

[120] A. Shatnawi, A.-D. Seriai, and H. Sahraoui. Recovering software product line architecture of a family of object-oriented product variants. *Journal of Systems and Software*, 131 :325 – 346, 2017.

[121] A. Shatnawi, A.-D. Seriai, H. Sahraoui, and Z. Alshara. Reverse engineering reusable software components from object-oriented APIs. *Journal of Systems and Software*, 131 :442 – 460, 2017.

[122] L. Soler, S. Nicolau, A. Hostettler, J.-B. Fasquel, V. Agnus, A. Charnoz, J. Moreau, B. Dallemagne, D. Mutter, and J. Marescaux. *Computational Surgery and Dual Training*, part III, ch. I, pages 139–154. Springer, first edition, 2010.

[123] S. Sorlin, C. Solnon, and J.-M. Jolion. A generic graph distance measure based on multivalent matchings. In Abraham Kandel, Horst Bunke, and Mark Last, editors, *Applied Graph Theory in Computer Vision and Pattern Recognition*, volume 52 of *Studies in Computational Intelligence*, pages 151–181. Springer, 2007.

[124] M. B. Srichai, R. P. Lim, S. Wong, and V. S. Lee. Cardiovascular applications of phase-contrast MRI. *American Journal of Roentgenology*, 192, 2009.

[125] C. Szyperski. *Component Software : Beyond object-oriented programming*. Addison-Wesley, 2nd edition, 2002.

[126] A. Talaminos-Barroso, M. A. Estudillo-Valderrama, L. M. Roa, J. Reina-Tosina, and F. Ortega-Ruiz. A machine-to-machine protocol benchmark for ehealth applications – use case : Respiratory rehabilitation. *Computer Methods and Programs in Biomedicine*, 129 :1 – 11, 2016.

[127] G. Trébuchet, J.-B. Fasquel, A. Lecluse, C. Cavaro Ménard, and S. Willoteaux. Region-based active contours for computer-aided analysis of carotid phase contrast MRI. In *The 26th IEEE International Symposium on Computer-Based Medical Systems, Porto, Portugal*, 2013.

[128] G. Trébuchet, J.-B. Fasquel, C. Cavaro Ménard, and S. Willoteaux. Coupling anatomical and functional information for the computer-aided delineation of phase-contrast MRI images using active contours. In *International Conference on Image Processing Theory, Tools and Applications, Istanbul, Turkey*, 2012.

[129] G. Trébuchet. *Segmentation par contours actifs de séquences de vélocimétrie IRM : Application aux artères carotides.* PhD thesis, 2013. Thèse de doctorat de l'université d'Angers.

[130] G. Valente, G. Crimi, N. Vanella, E. Schileo, and F. Taddei. nmsbuilder : Freeware to create subject-specific musculoskeletal models for opensim. *Computer Methods and Programs in Biomedicine*, 152 :85 – 92, 2017.

[131] M. C. Vanegas, I. Bloch, and J. Inglada. Fuzzy constraint satisfaction problem for model-based image interpretation. *Fuzzy Sets and Systems*, 286(Supplement C) :1 – 29, 2016. Theme : Images and Clustering.

[132] A. Vergnaud, J.-B. Fasquel, and L. Autrique. Python-based internet tools in control education. In *IFAC Workshop on Internet Based Control Education, Brescia, Italy*, 2015.

[133] S.-D. Wu, C.-W. Wu, S.-G. Lin, K.-Y. Lee, and C.-K. Peng. Analysis of complex time series using refined composite multiscale entropy. *Physics Letters A*, 378(20) :1369 – 1374, 2014.

[134] Y. Yin, Y. Zeng, X. Chen, and Y. Fan. The internet of things in healthcare : An overview. *Journal of Industrial Information Integration*, 1 :3 – 13, 2016.

[135] M. Zaslavskiy, F. Bach, and J.-P. Vert. *Many-to-Many Graph Matching : A Continuous Relaxation Approach*, pages 515–530. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[136] L. Zhang, M. Hub, S. Mang, C. Thieke, O. Nix, C. P. Karger, and R. O. Floca. Software for quantitative analysis of radiotherapy : Overview, requirement analysis and design solutions. *Computer Methods and Programs in Biomedicine*, 110(3) :528 – 537, 2013.

[137] J. Zhao and L. Itti. shape DTW : Shape dynamic time warping. *Pattern Recognition*, 74 :171 – 184, 2018.

[138] Y.-M. Zhu and S. M. Cochoff. An object-oriented framework for medical image registration, fusion, and visualization. *Computer Methods and Programs in Biomedicine*, 82(3) :258 – 267, 2006.

Annexes

1. **Pages 75-88 :** J.-B. Fasquel and N. Delanoue. A graph based image interpretation method using a priori qualitative inclusion and photometric relationships. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2018.
2. **Pages 89-98 :** J.-B. Fasquel and N. Delanoue. An approach for sequential image interpretation using a priori binary perceptual topological and photometric knowledge and k-means based segmentation. Journal of the Optical Society of America A. 2018.
3. **Pages 99-113 :** J.-B. Fasquel, V. Agnus, J. Moreau, L. Soler, and J. Marescaux. An interactive medical image segmentation system based on the optimal management of regions of interest using topological medical knowledge. Computer Methods and Programs in Biomedicine, 2006.
4. **Pages 115-120 :** A. Jamin, J.-B. Fasquel, A. Humeau-Heurtier, P. Abraham, G. Leftheriotis, and S. Henni. Characterization of home-acquired blood pressure time series using multiscale entropy for patients treated against kidney cancer. In Fourth International Conference on IoT Technologies for HealthCare, Angers, France, 2017.
5. **Pages 121-129 :** J.-B. Fasquel, A. Lécluse, C. Cavaro-Ménard, and S. Willoteaux. A semi-automated method for measuring the evolution of both lumen area and blood flow in carotid from phase contrast MRI. Computers in Biology And Medicine, 2015.
6. **Pages 131-136 :** A. Jamin, J.B. Fasquel, M. Lhommeau, E. Cornet, S. Abadie-Lacourtoisie, S. Henni, and G. Leftheriotis. An aggregation plateform for iot-based healthcare : Illustration for bioimpedancemetry, temperature and fatigue level monitoring. In Third International Conference on IoT Technologies for HealthCare, Vasteras, Sweden, 2016.
7. **Pages 137-143 :** J.-B. Fasquel, V. Agnus, and J. Lamy. An efficient and generic extension to itk to process arbitrary shaped regions of interest. Computer Methods and Programs in Biomedicine, 2006.
8. **Pages 145-156 :** J.-B. Fasquel, G. Brocker, J. Moreau, N. Papier, V. Agnus, C. Koehl, L. Soler, and J. Marescaux. A modular and evolutive component oriented software architecture for patient modeling. Computer Methods and Programs in Biomedicine, 2006.
9. **Pages 157-173 :** J.-B. Fasquel and J. Moreau. A design pattern coupling role and component concepts : Application to medical software. The Journal of Systems and Software, 2011.
10. **Pages 175-190 :** M. Madec, J.B. Fasquel, W. Uhring, P. Joffre, and Y. Hervé. Optoelectronic implementation of helical cone-beam computed tomography algorithms. Optical Engineering, 2008
11. **Pages 191-205 :** M. Madec, J.-B. Fasquel, W. Uhring, P. Joffre, and Y. Hervé. Optical implementation of the filtered backprojection algorithm. Optical Engineering, 2007.

A graph based image interpretation method using a priori qualitative inclusion and photometric relationships.

Jean-Baptiste Fasquel and Nicolas Delanoue

Abstract—This paper presents a method for recovering and identifying image regions from an initial oversegmentation using qualitative knowledge of its content. Compared to recent works favoring spatial information and quantitative techniques, our approach focuses on simple a priori qualitative inclusion and photometric relationships such as *"region A is included in region B"*, *"the intensity of region A is lower than the one of region B"* or *"regions A and B depict similar intensities"* (photometric uncertainty). The proposed method is based on a two steps' inexact graph matching approach. The first step searches for the best subgraph isomorphism candidate between expected regions and a subset of regions resulting from the initial oversegmentation. Then, remaining segmented regions are progressively merged with appropriate already matched regions, while preserving the coherence with a priori declared relationships. Strengths and weaknesses of the method are studied on various images (grayscale and color), with various intial oversegmentation algorithms (k-means, meanshift, quickshift). Results show the potential of the method to recover, in a reasonable runtime, expected regions, a priori described in a qualitative manner. For further evaluation and comparison purposes, a Python opensource package implementing the method is provided, together with the specifically built experimental database.

Index Terms—Image interpretation, inexact graph matching, subgraph isomorphism, qualitative knowledge, inclusion relationships, photometric relationships

1 INTRODUCTION

Graph theory plays an important role in modern image analysis [1], [2], [3]. In this context, graph matching is widely used [3], for various applications such shape analysis, 3D-recognition, video and image database indexing [2], as underlined by recent papers involving graph matching [4], [5], [6].

Graph matching techniques are based on an input graph from the image (usually oversegmented), where nodes represent regions and edges relations between regions (e.g. relative position [6]), to be matched with a model graph. The difficulty regards the fact that several nodes of the input graph must be matched with one node of the model graph (many-to-one mapping), this kind of situation being called inexact graph matching [2], [7]. The mapping is usually computed by optimizing an objective function [2], [6], [8] or, for instance, by satisfying constraints (constraint satisfaction problem) [2], [9], [10].

In this paper, we propose an inexact graph matching method exploiting inclusion and photometric relationships between regions in a qualitative manner. This idea is to exploit a priori knowledge resulting from the visual perception on the scene content (*qualitative* knowledge [10]). As illustrated by figure 1, the qualitative knowledge we consider in this study regards binary (i.e. compared to quantified) relationships between regions in terms of inclusion (e.g. liver tumors belong to the liver) and photometry. Considered photometric relationships are limited to the perceived contrasts (i.e. related to differences between mean intensities). For instance, the notion *"region A is darker than region B"* is assumed to be translated into *"the mean intensity of region A is lower than the mean intensity of region B"* (we consider the grayscale-converted image in case of color images). Such a qualitative knowledge is assumed to be easier to acquire and less sensitive to fluctuations (i.e. from an image to another), compared to *quantitative* one, often related to statistical theory (e.g. statistics related to the photometry of some image regions) [10], [11]. As recently underlined [12], a major limitation of such quantitative approaches regards the requirement of a representative, and often large, training data set allowing to model a priori knowledge. For instance, in case of examples reported on figure 1, it is expected that a hypodense liver tumor remains darker than liver tissues whatever the patient, even if its mean intensity may vary from one patient to another or from one acquisition protocol or device to another. As illustrated by figure 1, our proposal aims at also integrating the notion of similar intensities to manage uncertainty (sign entities of similar tones).

• *J.-B. Fasquel and N. Delanoue are with the LARIS laboratory, EA4094, University of Angers, 62 avenue Notre Dame du Lac, 49000 Angers, France.*
E-mail: Jean-Baptiste.Fasquel@univ-angers.fr

Fig. 1. Example of qualitative inclusion and photometric relationships. Left : CT Scanner image of the abdominal area. The liver area depicts hypodense tumors, included within the liver area, and darker than liver tissues, being, in turns darker than liver vessels. Right: natural color image. The sign depicts clear inclusion relationships (e.g. stars included within a square, itself belonging to a white background). Photometric relationships can be deduced from the perception of tones (grayscale-converted image). Yellow stars and strip depict a similar tone (relatively bright, compared to dark elements and medium tones), being less bright than white elements.

Many graph-based techniques involving spatial and/or photometric information have been proposed, as illustrated by recent works [6], [9], [10]. Compared to the proposed approach, some of the related works do not combine both inclusion and photometric information. For instance, [10] is limited to spatial information only, while [6] only considers relative position of regions (and not inclusion) and photometry (appearence). Other related works integrate both information but in a quantitative manner. For instance, [9] requires a learning step for parameterizing relationships (quantitative model), [6] weights both photometric and spatial information.

The main contribution of this paper regards the use of this qualitative a priori information and the proposal of a non parametric inexact graph matching technique. A part of the contribution concerns the implementation of the proposed method as a freely available Python package [1], provided together with the considered experimental database. Section 2 aims at describing the proposed approach, while section 3 concerns its experimental evaluation. Before concluding, section 4 discusses about this method.

2 Proposed method

Figure 2 provides an overview of the proposed approach. Graphs from model and images are respectively described in sections 2.1 and 2.2.

From these two graphs, the proposed method consists in three steps. First, one computes common subgraph isomorphisms (detailed in section 2.3). The second step aims at selecting the best subgraph isomorphism among the set of common ones (detailed in section 2.4). Finally, in a third step, regions are finally recovered and identified by merging unmatched nodes (detailed in section 2.5). Last section 2.6 focuses on the a priori model validity.

1. http://perso-laris.univ-angers.fr/~fasquel/scikit-gtimage.html

2.1 Graphs from model

As illustrated by figure 2-model, we consider that an image I is composed of a set M (region labels) of a priori known regions $X(u), u \in M$. $G_{m,t} = (M, F_t)$ and $G_{m,p} = (M, F_p)$ respectively regard inclusion and photometric relationships between expected regions, where M is the set of nodes, F_t and F_p are oriented edges. Note that the underscripted 'm' stands for model image (a priori known image content), 't' stands for topology (limited to inclusion in our case) and 'p' stands for photometry.

Oriented edges of F_t correspond to inclusion relationships between image regions:

$$k \xrightarrow{t} l \Leftrightarrow X(k) \subsetneq X(l) \quad (1)$$

Oriented edges of F_p denote order relations between mean intensities of related region:

$$k \xrightarrow{p} l \Leftrightarrow \bar{X}(k) < \bar{X}(l), \quad (2)$$

means that region $X(k)$ is assumed to be less bright than $X(l)$.

$$k \xleftrightarrow{p} l \Leftrightarrow \bar{X}(k) \simeq \bar{X}(l), \quad (3)$$

means that region $X(k)$ is assumed to be either slightly brighter or darker than $X(l)$ (i.e. numerically similar, compared to other regions). This enables to manage uncertain brightness relationships (detailed in section 2.3).

2.2 Graphs from image

As illustrated by figure 2 (input image and built relationships), we consider that a labelled image I (e.g. resulting from oversegmentation) is composed of a set of regions $X(u), u \in N$, where the set N denotes region labels. $G_{r,t} = (N, E_t)$ and $G_{r,p} = (N, E_p, \mu)$ ('r' stands for real image by opposition to the model image) respectively regard topological and photometric relationships between expected regions, where N are graph nodes, E_t and E_p are oriented edges. $G_{r,p}$ is an attributed graph so that each node i is associated to an attribute μ_i representing the mean intensity of region $X(i)$: $\mu_i = \bar{X}(i)$ (similar to the appearance information considered in [6]).

The graph $G_{t,r}$ is built first, by automatically detecting inclusion relationships from labelled regions, using morphological operations. Note that initial labelled regions may need to be split into several ones. This concerns regions composed of several connected components, some of them being included in different

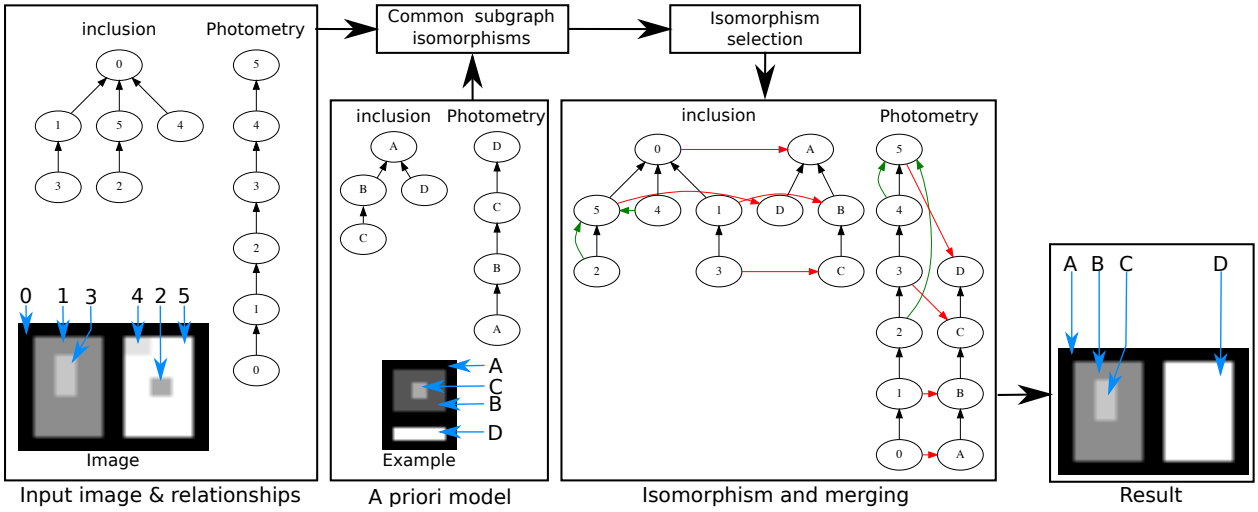

Fig. 2. Principle of the approach applied to an input segmented image (labelled regions 0 to 5), using an a priori model (relationships regarding expected regions A, B, C and D). Labelled regions 2 and 4 represent artefacts that can be encountered in real images. Based on automatically built relationships, one first searches for subgraph isomorphisms (between model and built graphs) that are common to both inclusion and photometric relationships. According to a selection criterion (section 2.4), only one is considered (represented by red arrows). Remaining unmatched nodes are finally matched (green arrows) for recovering and identifying regions (see section 2.5). Both red and green arrows represent the inexact graph matching.

regions. Roughly speaking, region splitting is done recurvely. For any given pair, if one region intersects the other without being included into it, then the region is appropriately splitted, leading to new regions and relationships. This procedure is called itself until no new split is required. The $G_{p,r}$ graph is finally computed from regions related to $G_{t,r}$, by comparing their mean intensities.

2.3 Subgraph isomorphisms

As reported in figure 2, the first step of the proposed method aims at finding common subgraph isomorphisms between $\bar{G}_{r,t}$ and $\bar{G}_{m,t}$, as well as between $\bar{G}_{r,p}$ and $\bar{G}_{m,p}$. $\bar{G}_{.,.}$ denotes the transitive closure of $G_{.,.}$. Transistive closure is required to consider matchings such as the one reported in figure 2 (red arrows) regarding photometry, where nodes 3 and 5 are respectively matched with C and D, although the node 4 stands between 3 and 5. Common (subgraph) isomorphisms will enable to identify candidate region sets (of the image under analysis) that simultaneously verify both inclusion and photometric relationships declared by the a priori model.

Let us consider that $H_.$ is the set of isomorphims between subgraphs of $\bar{G}_{r,.}$ and the graph $\bar{G}_{m,.}$ (. is either t or p). An isomorphism $h_. \in H_.$ is associated to a subgraph $G'_{r,.} \subseteq \bar{G}_{r,.}$ ($G'_{r,.} = (N', E'_.)$, with $N' \subseteq N$ and $E' \subseteq E$), and can be defined by a bijective mapping:

$$h_. = \{(v_i, v_i^h) \in N' \times M\} \tag{4}$$

meaning that any node $v_i \in N'$ is associated to a node $v_i^h \in M$, in a bijective way. In term of edges, $\forall (v_i, v_j) \in N' \times N', (v_i, v_j) \in E'_. \Leftrightarrow (v_i^h, v_j^h) \in F_.$ [2].

The set H of common subgraph isomorphisms is defined by:

$$H = H_t \cap H_p \tag{5}$$

Figure 2 provides an example of common subgraph isomorphism (modeled by red arrows). Note that some cycles may be declared in $\bar{G}_{m,p}$ while not appearing in $\bar{G}_{r,p}$, in case of a priori declared photometric similarities. As a consequence, subgraph isomorphisms H_p must be searched over all *unwrapped* versions of $\bar{G}_{m,p}$. For instance, assuming that $\bar{G}_{r,p}$ has no cycles, let us consider $\bar{G}_{m,p}$ depicts two particular nodes (i, j) such that $i \leftrightarrow j$. In such a case, we consider two graphs: $\bar{G}^1_{m,p}$, with $i \rightarrow j$ and $\bar{G}^2_{m,p}$ with $j \rightarrow i$ (illustrated by figure 3-Model and unwrapping). H_p becomes the union of subgraph isomorphisms between $\bar{G}_{r,p}$ and both $\bar{G}^1_{m,p}$ and $\bar{G}^2_{m,p}$. This leads to following formulation for H:

$$H = H_t \bigcap \Big(\bigcup_{\bar{G}^i_{m,p} \in \bar{G}^*_{m,p}} H^i_p \Big) \tag{6}$$

where $\bar{G}^*_{m,p}$ is the set of all possible unwrapped versions of $\bar{G}_{m,p}$ (without cycles). $|\bar{G}^*_{m,p}| = 2^n$, where n is the number of cycles in $G_{m,p}$. H^i_p denotes all subgraph isomorphisms between $\bar{G}_{r,p}$ and $\bar{G}^i_{m,p} \in \bar{G}^*_{m,p}$.

This formulation is justified by the graph theory related to isomorphisms of subgraphs [2], as hereafter shortly detailed. A graph $G_1 = (V_1, E_1)$ is a subgraph

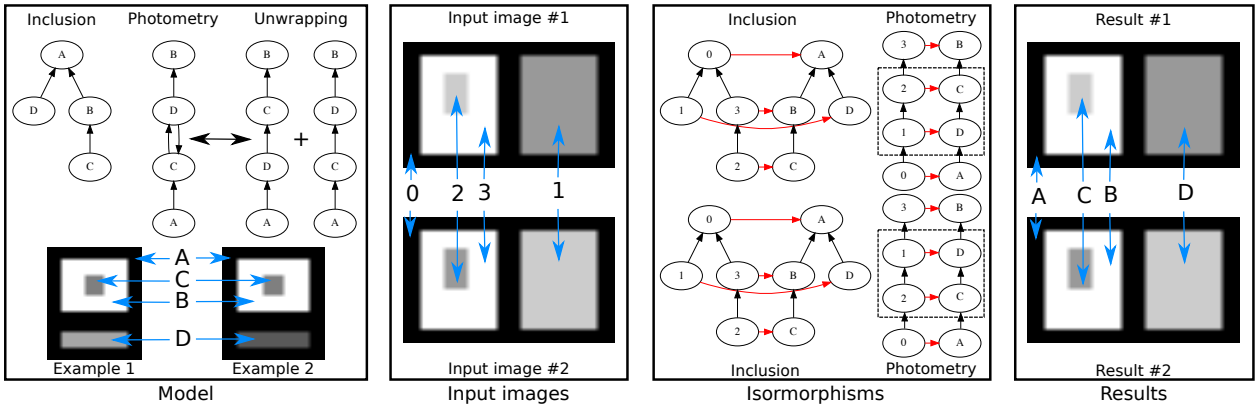

Fig. 3. Management of photometric uncertainty: principle. Region C is declared to be either brighter or darker than D. Based on common subgraph isomorphisms searched on unwrapped photometric graphs, regions 1 and 2 are correctly identified although photometric relations vary from input image 1 to input image 2.

of a graph $G_2 = (V_2, E_2)$ if $V_1 \subset V_2$ and $E_1 = E_2 \cap (V_1 \times V_1)$. An isomorphism h between a subgraph G_1 of G_2 and a graph $G = (V, E)$, involves in particular [2]:

$$\forall e = (v_i^h, v_j^h) \in E, \exists e_1 = (v_i, v_j) \in G_1 \quad (7)$$

This condition is not satisfied when considering a (sub)graph such as $\bar{G}_{r,p}$ (without cycles) with a graph of type $\bar{G}_{m,p}$ (with cycles), justifying equation 6.

The management of photometric similarities is illustrated by figure 3.

In the proposed implementation, in order to obtain H in equation 5, the intersection between H_t and H_p is computed as following. First, one searches for subgraph isomorphisms related to inclusion relationships (H_t), using the VF2 algorithm [13]. Then, we only keep elements of H_t belonging to H_p, that is to say belonging, at least, to one H_p^i, according to equation 6. This consists in filtering H_t with respect to photometric constraints.

2.4 Isomorphism selection

As reported in figure 2, the second step of the proposed method concerns the selection of a subgraph isomorphism among common isomorphisms. Figure 4 provides two examples of situations involving several common isomorphisms (modeled with red arrows), each one leading to a different result.

Before detailing the isomorphism selection criterion, we introduce following entities. For a given isomorphism $h \in H$, we define an equivalence relation $\overset{h}{\sim}$ so that:

$$\forall (k, l) \in N^2, \; k \overset{h}{\sim} l \Leftrightarrow k^h \overset{p}{\leftrightarrow} l^h \quad (8)$$

Based on $\overset{h}{\sim}$, we define a simplified photometric graph $G_{r,p}^h = (N^h, E^h, \mu^h, \sigma^h)$ so that:

$$N^h = N / \overset{h}{\sim} \quad (9)$$

being the quotient set (set of equivalence classes), i.e. each element $i \in N^h$ is a set of equivalent elements of N. For instance, in figure 4-bottom-result 1, we have $N^h = \{\{0\}, \{1, 2\}, \{4\}\}$ (nodes 1 and 2 are equivalent, due to a priori declared similar regions C and D). In figure 4-top-result 1, we have $N^h = \{\{0\}, \{1\}, \{2\}\}$.

For a given subgraph isomorphism h, μ^h represents the mean intensity associated to each equivalence class, while σ^h regards its standard deviation. We consider that $\forall i \in N^h$:

$$\mu_i^h = \frac{1}{|i|} \sum_{k \in i} \mu_k \quad (10)$$

and

$$\sigma_i^h = \sqrt{\frac{1}{|i|} \sum_{k \in i} (\mu_k - \mu_i^h)^2} \quad (11)$$

Note that $\sigma_i^h = 0$ if $|i| = 1$.

Edges of $G_{r,p}^h$ are defined by:

$$E^h = \{(i, j) \in N^h \times N^h | \exists u \in i, v \in j, u \overset{p}{\rightarrow} v \in E\} \quad (12)$$

Among possible subgraph isomorphisms, the selected isomorphism is h^*, maximizing the objective function $E(.)$:

$$h^* = \underset{h \in H}{\operatorname{argmax}}(E(h)) \quad (13)$$

By considering $S^h = \{i \in N^h | |i| > 1\}$, $E(h)$ is defined by:

$$E(h) = \frac{\frac{1}{|E^h|} \sum_{i \overset{p}{\rightarrow} j \in E^h} |\mu_i^h - \mu_j^h|}{\frac{1}{|S|} \sum_{i \in N^h} (\sigma_i^h)} \text{ if } S \neq \emptyset \quad (14)$$

and,

$$E(h) = \frac{\frac{1}{|E^h|} \sum_{i \overset{p}{\rightarrow} j \in E^h} |\mu_i^h - \mu_j^h|}{\sigma} \text{ if } S = \emptyset \quad (15)$$

where σ is the standard deviation of mean intensity differences related to each $i \overset{p}{\rightarrow} j \in E$.

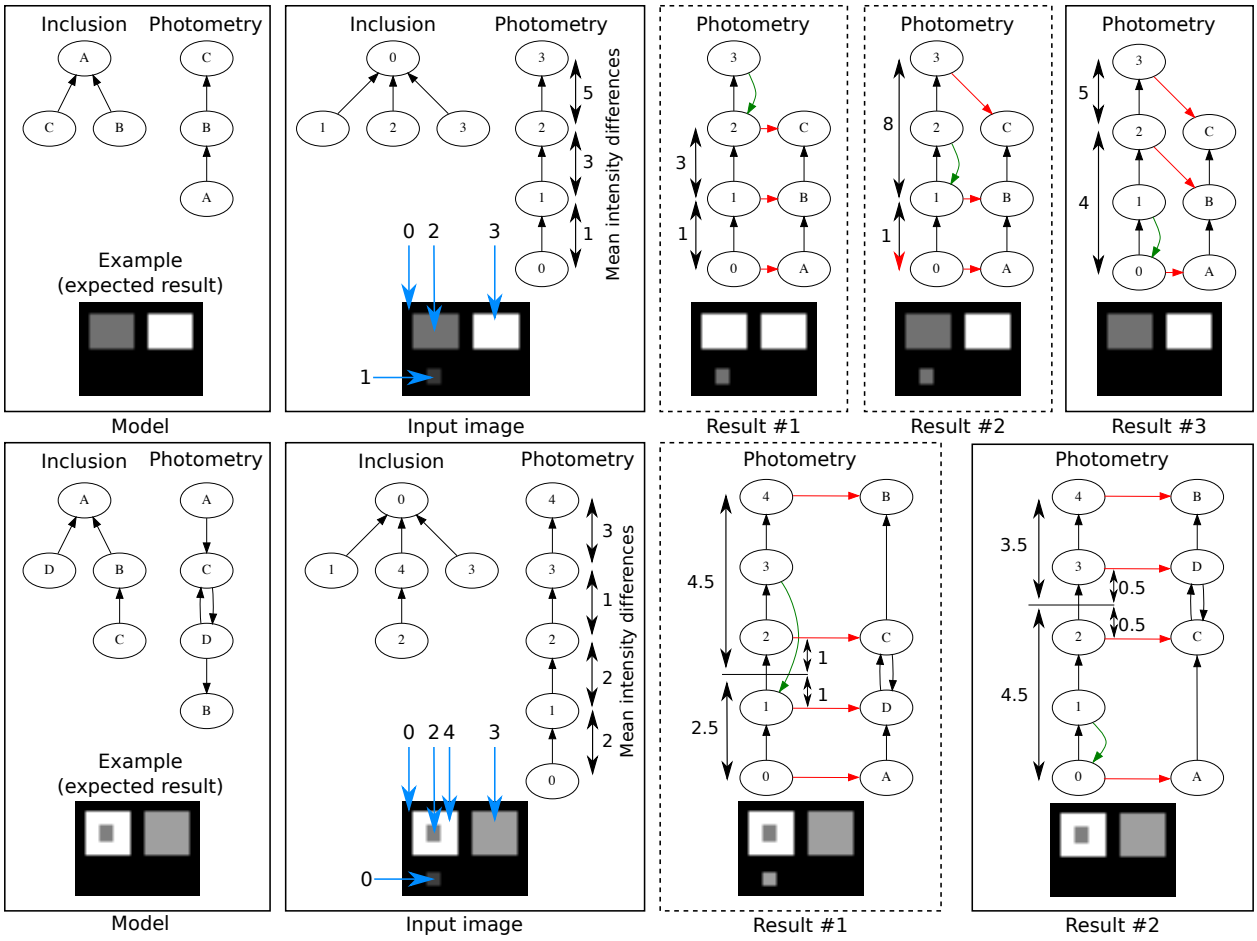

Fig. 4. Isomorphism selection. Top: case without photometric similarities. Bottom: case involving a photometric similarity. From left to right: a priori model, input image and built graphs (intensity differences are indicated along edges of the photometric graphs), results for different isomorphisms. Results surrounded by solid line boxes are obtained using the proposed selection criterion.

In both cases, the numerator aims at maximizing the average distance between neighboring matched nodes, where the distance between i and j is $|\mu_i - \mu_j|$.

For $S = \emptyset$, one simultaneously attempts to minimize variations of the average distance in order to favor regularly spaced matchings. This is illustrated in figure 4-top: the numerator favors the isomorphism of result 3 versus result 1, while the denominator favors result 3 versus result 2. In this example, maximizing E enable to select the isomorphism leading to the best result (i.e. result 3).

For $S \neq \emptyset$, one simultaneously attempts to minimize the mean intensity variation between nodes (regions) matched with a priori declared similar regions. In figure 4-bottom, $\sigma^h_{\{2,3\}}$ (result #2) is smaller than $\sigma^h_{\{1,2\}}$ (result #1), therefore favoring result 2 (corresponding to the best result). This is inspired from criteria typically considered for optimizing clustering by minimizing within-cluster standard deviation. This choice has also a similarity with the notion of signal to noise ratio (the signal being associated with the numerator, and the noise with the denominator).

Note that this step is the only one that does not exploit inclusion relationships: the selection criterion is based on photometry relationships only.

2.5 Merging

The inexact graph matching problem is solved by finally matching remaining nodes (green arrows in figures 2 and 4), similarly to [14]). We consider the A^{h^*} of already matched nodes ('A' standing for 'assigned'): $A^{h^*} = \{i \in N | \exists (i, i^{h^*}) \in h^*\}$. The purpose is to match every node $n \in \bar{A}^{h^*}$ (i.e. $N \setminus A^{h^*}$) with a node $m \in A^{h^*}$ (i.e. that is to say with $m^{h^*} \in M$). This involves the removal of $n \in N$ from $G_{r,.}$ and the updating of the image region $X(m) \leftarrow X(n) \cup X(m)$ (including the mean intensity μ_m).

Regions can be merged according to a specific constraint imposed by the inclusion graph, in particular

Require: h^*, $G_{r,.}$ and $G_{m,.}$
1: **while** $N \setminus A^{h^*} \neq \emptyset$ **do**
2: L update with $G_{r,.}$
3: validity $\leftarrow$ **false**
4: **while** validity is **false do**
5: $(i,j) \leftarrow$ next in L {From first}
6: validity $\leftarrow test(h^*, G_{r,.}, G_{m,.}, (i,j))$
7: **end while**
8: $G_{r,.} \leftarrow$ update using (i,j)
9: $N \setminus A^{h^*} \leftarrow$ update using $G_{r,.}$
10: **end while**
11: **return** $G_{r,.}$

Fig. 5. Pseudo code of the merging procedure

to avoid introducing topological incoherence. For instance, in figure 2, region 2 cannot be merged region 0, 1 or 3. We consider that a node $n \in N$ can be merged either with its direct father ($G^+_{r,t}(n)$, singleton due to the tree structure), or with its direct predecessors ($G^-_{r,t}(n)$), or its brothers ($G^-_{r,t}(G^+_{r,t}(n)) \setminus \{n\}$). This leads to a subset $C(n)$ of target candidates:

$$C(n) = G^+_{r,t}(n) \cup G^-_{r,t}(n) \cup (G^-_{r,t}(G^+_{r,t}(n)) \setminus \{n\}) \quad (16)$$

A merging $n \rightarrow m, m \in C(n)$ involves a modification of region $X(m)$ and therefore an unpredicable modification of relationships in $G_{r,p}$ due to the modification of its mean intensity (μ_m attribute), depending, in particular, on merged region sizes. Care must be taken not to break the overall matching with the model graph regarding photometric relationships $G_{m,p}$. In our proposal, a merging is allowed only if it does not invalidate the initial matching h^*.

Therefore, our proposal involves progressive node merging (local modification) with a global verification of the coherence. This leads to a greedy algorithm where most photometrically similar nodes are merged first, under the condition that the overall matching is not broken.

Figure 5 provides the pseudo-code of the implemented procedure, where the ordered list L embeds merging candidates:

$$L = ordered[\bigcup_{n \in \bar{A}^{h^*}} (\bigcup_{m \in C(n) \cap A^{h^*}} n \rightarrow m)] \quad (17)$$

Elements of L are sorted in increasing order of mean intensity differences (distance $d(i,j) = |\mu_j - \mu_i|, (i,j) \in L$).

The procedure iterates over remaining unmatched nodes (figure 5-line-1). Note that some nodes $n \in \bar{A}^{h^*}$ may not have any valid target (if $C(n) \cap A^{h^*} = \emptyset$) at the beginning of the merging procedure. At the begining of each iteration, the list L is udpated according to the context (figure 5-line-2). A second loop iterates over elements of L, considering the first encountered that do not break the coherence of the global matching (*test* procedure invoked figure 5-line-6). When the first valid merging candidate is found, the entire context is updated before the next iteration. In particular, both input graphs $G_{r,.}$ are updated (figure 5-line-8): node i is removed, region $X(j)$ is updated. $N \setminus A^{h^*}$ updates due to node removal from $G_{r,.}$ (altering the set N).

In this version of the procedure, there is no garantee of convergence. Indeed, if *validity* is never *true* whatever the element of L (figure 5-lines 5-6), then the procedure stops (not specified in figure 5 for clarity) and remaining unmatched nodes (not empty $N \setminus A^{h^*}$) are removed from N. In practice, this can concern background regions (e.g. regions lying outside the modeled road sign reported in figure 1).

2.6 A priori model validity

An ill-posed a priori model can lead to "undecidable" result. For instance, this can concern two regions (without any subregion) included within the same region, and sharing similar mean intensity.

This can be detected by searching for common automorphisms to both $G_{m,t}$ and $\bar{G}_{m,p}$. $\bar{G}_{m,p}$ is considered so that all nodes belonging to a given cycle share the same successors and predecessors. By considering that H_{auto} is defined as H in equation 5, except that it involves model graphs only, we argue that the model is well defined if and only if : $|H_{auto}| = 1$.

3 EXPERIMENTS

In section 3.1, we describe the considered experimental database and related a priori models. Then, section 3.2 concerns the evaluation protocol. Finally, experiments and results are detailed in section 3.3.

3.1 Database

To our knowledge, their is not public evaluation database providing images with a priori knowledge focusing on both considered inclusion and photometric relationships, together with manually identified regions (ground truth). For instance, the well-known Berkeley Segmentation Dataset and Benchmark [15] provides a large set of images and related manually segmented regions, but without identification and classification of regions in terms of inclusion and relative photometric relationships. For this reason, we built our own evaluation database (figure 6), freely available for further evaluation and comparison purposes[1]. Note that image 2 is extracted from [16].

Figure 6 presents the database, including both images and a priori knowledge. Each region has been manually segmented and identified. Considered images are either 2D or 3D and involve from 3 to 13 distinct regions. For color images, a priori knowledge regarding photometric relationships is based on relative mean intensity of grayscale-converted color images (can be seen as the overall brightness of related regions). Note that initial over-segmentations are

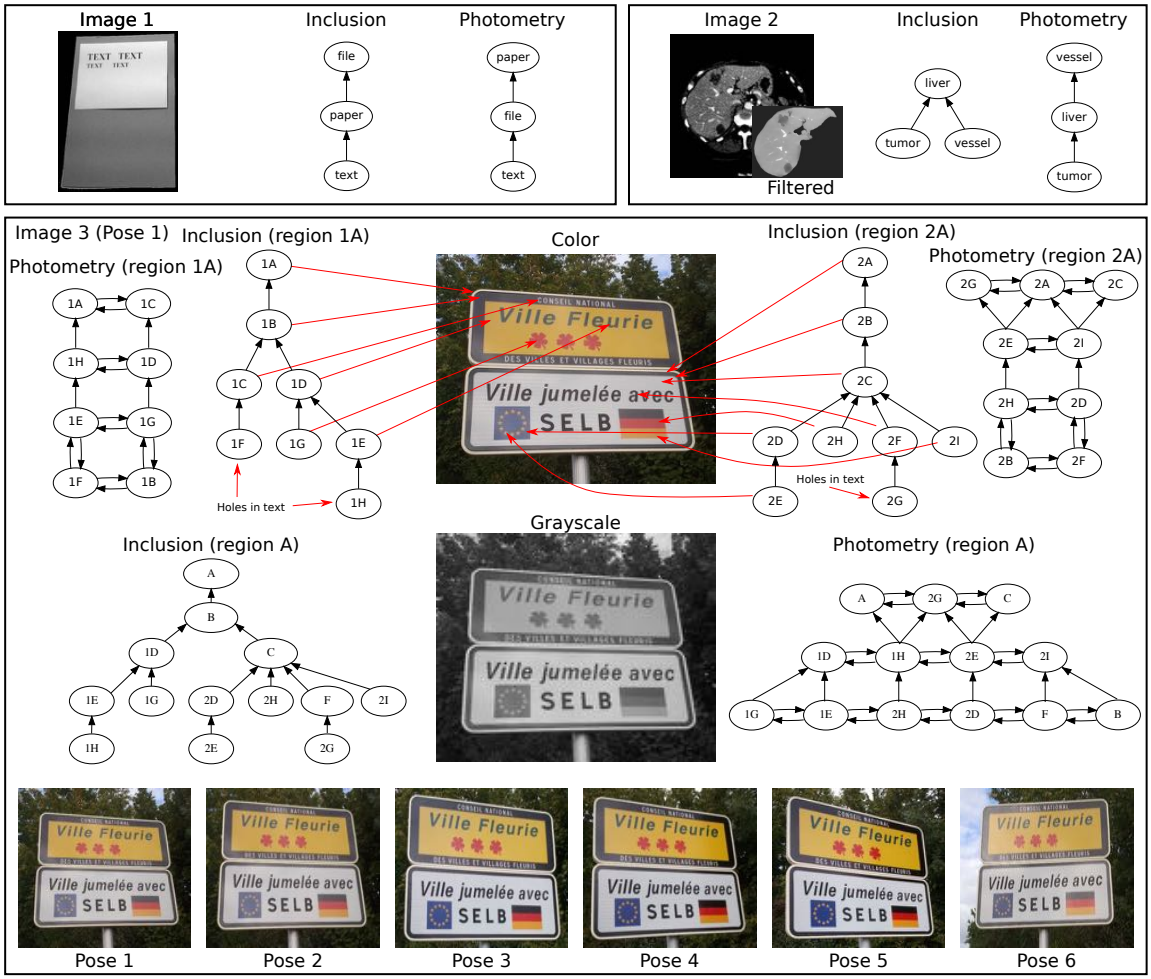

Fig. 6. Experimental database. For image 3, one considers the model regarding each part of the road sign (regions 1A and 2A), as well as the model of the entire sign (region A).

performed on color images, although inexact graph matching considers grayscale-converted images (i.e. for building photometric relationships).

Image 1 is a natural indoor image. A priori relationships are built knowing that the image contains a file of medium tone which, in turns, holds a white paper on which a dark text is written.

Image 2 is a 3D medical image (129 slices, 512×512 pixels per slice) focusing on the abdominal area and acquired with a CT X-Ray Scanner. For experiments, one focuses on liver area only. Topological knowledge is provided by the knowledge of human anatomy (both liver vessels and tumors are included into liver). Photometric relationships (G_P) are established according to the knowledge of the imaging modality and physical properties of tissues (radiodensity). Liver tumors are hypodense structures, therefore darker than liver tissues. Liver vessels are assumed to be brigther than liver tissues, thanks to the use of a radiocontrast agent [17].

Image 3 represents a road sign. One considers three a priori models: the one corresponding to the entire road sign (region A), and two additional ones, respectively regarding regions 1A and 2A (upper and lower parts of the entire road sign). In our sense, inclusion relationships are quite straighforward for each part of the road sign (i.e. 1A and 2A). Note that holes in text letters are associated to dedicated regions. For the model of the entire road sign, to ensure the coherence of inclusions, one assumes that similarly painted regions are merged: for instance, white regions 1A and 2A are seen as white region A. Similarly, B is the union of 1B and 2B, C is the union of 1C and 2C, and F is the union of 1F and 2F. Concerning photometric relationships, we consider only 3 distinct tone levels: dark regions (including black, red and blue ones), medium regions (yellow ones) and white regions. This choice is motivated by mean graylevels reported in figure 7, which furthmore justifies the relevance of the notion of similarities (photometric uncertainty illustrated, for instance, by

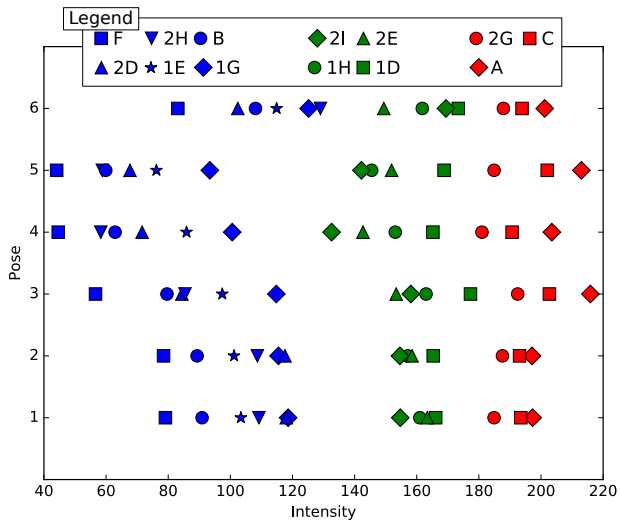

Fig. 7. Mean intensities related to poses of image 3 (region A is considered). Statistics are computed from manually segmented regions. The 3 groups (blue, green, red) correspond to the 3 sets of photometrically similar regions (a priori 3 distincts tones). Groups' ordering is constant, while within group ordering varies.

regions 2I and 2E in poses 3 and 4).

The entire experimental database involves 3 situations and 8 images (6 poses for image 3).

3.2 Evaluation protocol

Comparison with third party similar approaches is tedious because, to our knowledge, there is no similar graph matching based approaches focussing on region recognition using both inclusion and photometric relationships. Therefore, evaluation is limited to measuring the recognition efficiency for different initial oversegmentations, in order to study strengths and weaknesses of our proposal.

Considered segmentation algorithms are k-means, meanshift [18] and quicksift [19]. These popular algorithms are chosen to consider segmentation using photometric information only (k-means and meanshift) and both photometric and spatial information (quickshift). Our approach is then applied to the segmentation result.

For images 1 and 3, computations are performed on the entire image and on regions of interest (ROI) corresponding to the root of inclusion graphs. ROIs are automatically obtained using level-set based active contours, initialized at image boundaries (see tests 1-b and 7-a reported in figure 8).

For image 2, the ROI is the liver, assumed to be interactively segmented using a dedicated software (see e.g. [20], [21]), combining basic algorithms and manual drawing, because the automatic segmentation of abdominal structures remains a challenging issue [20]. Moreover, image 2 is initially filtered (anisotropic filtering) before the segmentation step.

In some cases, after the initial segmentation, additional automated processing steps are considered before applying our method to the result of the segmentation. A first preprocessing step concerns the progressive merging of some segmented regions until at least one common isomorphism is found (see test 2-b in figure 8). In particular, we consider the merging of neighbor regions sharing similar photometric properties using a region adjacency graph [22]. A second preprocessing step concerns the removal of the background when the ROI is the entire image (see test 2-b in figure 8). In such a case, one automatically removes the background by considering connected components of the inclusion graph that maximizes the number of subgraph isomorphisms with the a priori model (limited to a single component in experiments, due to considered a priori models).

For each experiment, the quality of the result is evaluated by comparison with manually segmented images, using the good classification rate (GCR) and the similarity index of each automatically identified region (local measurement). One also reports the number of initial labels (after the initial oversegmentation), the number of nodes in automatically built graphs, the number of subgraph isomorphisms regarding inclusion, and the resulting number of common isomorphims (illustrating the benefits of coupling both topogical and photometric information). The influence of the choice of the common isomorphism is also studied by computing the GCR for each common isomorphism (both best and worst GCR are reported). For each experiment, one indicates whether (Yes/No) the automatically selected isomorphism leads to the best result (in terms of GCR). If relevant, one also reports the proportion (percentage) of common isomorphims leading to the best GCR.

3.3 Results

Table 1 reports experiments and results. Figure 8 details some of them (not all of them, for clarity), and figure 9 focuses on different poses of image 3.

3.3.1 Overview

To understand table 1, let us consider the test #1-b (also detailed in figure 8): image 1 is segmented using Mean-Shift (MS) restricted to the file ROI (the outer boundary of region file being detected using level-sets). The segmentation produces 3 regions (labels). Built graphs ($G_{r,.}$) depict 6 nodes (while the model involves 3 nodes only - see figure 6). Regarding inclusion relationships, 5 subgraph isomorphisms are found ($|H_t|$). By integrating photometric information, one finally reduces the number of possible subgraph isomorphisms to 3 ($|H|$). The good classification rate (GCR) is 0.997 and the three expected entities are

TABLE 1
Results for different ROIs and segmentation algorithms (k-means (KM), meanshift (MS) or quickshift (QS)), where $\#Labels$ is the number of labelled regions. $|N|$ is the number of nodes in built graphs ($G_{r,.}$). $|H_t|$ is the number of subgraph isomorphisms related to inclusion relationships. $|H|$ regards common subgraph isomorphisms. GCR is the good classification rate. For clarity, only smallest and highest similarity indices are reported. 'Best H' indicated if the selected H leads to the best result (Yes/No), as well as the best and worst GCR (among all possible isomorphims). If 'Y', '%' indicates the proportion (in percentage) of isomorphims leading to the best GCR. In some cases, smallest regions have been filtered ('f').

#	Image	ROI	Seg.	#Labels	$\|N\|$	$\|H_t\|$	$\|H\|$	GCR	Similarities (max, min)	Best H ?	%
1-a	1	File	KM	3	5	5	3	0.995	1.0 (file) 0.79 (text)	Y (0.995-0.995)	100
1-b	1	File	MS	3	6	5	3	0.997	1.0 (file) 0.9 (paper-text)	Y (0.997-0.994)	66.7
1-c	1	File	QS	355	31	22	19	0.994	1.0 (file) 0.72 (text)	Y (0.994-0.994)	100
2-a	1	None	KM	4	13	22	8	0.998	1.0 (paper) 0.91 (text)	Y (0.998-0.996)	37.5
2-b	1	None	MS	7	18	28	8	0.997	1.0 (paper) 0.91 (text)	Y (0.997-0.997)	100
2-c	1	None	QS	389	26	29	20	0.99	0.99 (file-paper) 0.72 (text)	Y (0.99-0.99)	100
3-a	2	Liver	MS	7	8	40	6	0.938	0.94 (liver) 0.57 (vessel)	Y (0.938-0.92)	33.3
3-b	2	Liver	MS	9	10	70	14	0.918	0.97 (liver) **0.0** (tumor)	N (0.938-0.918)	-
3-c	2	Liver	MS	9	7 (f)	28	5	0.938	0.94 (liver) 0.53 (vessel)	Y (0.938-0.918)	40.0
4-a	2	Liver	KM	5	9	52	18	0.911	0.93 (liver) **0.0** (tumor)	N (0.938-0.041)	-
4-b	2	Liver	KM (m)	5	9	52	16	0.944	0.94 (liver) 0.5 (vessel)	Y (0944- 0.538)	25
5-a	3-Pose1	1A	MS	4	12	4	4	0.986	0.99 (1A,1B,1D) 0.89 (1H)	Y (0.986-0.945)	50
5-b	3-Pose1	1A	QS	14	22	48	44	0.899	0.99 (1A,1B) **0.0** (1G)	N (0.986-0.899)	-
5-c	3-Pose1	1A	QS	6	11	1	1	0.986	0.99 (1A,1B,1D) 0.92 (1F,1H)	Y (0.986-0.986)	100
6-a	3-Pose1	2A	MS	7	27	1500	63	0.988	1.0 (2A,2B) 0.94 (2D,2E)	Y (0.988-0.824)	38.1
6-b	3-Pose1	2A	QS	9	12	12	3	0.991	1.0 (2C) 0.93 (2E)	Y (0.991-0.946)	66.7
6-c	3-Pose1	2A	QS	6	11	12	3	0.991	1.0 (2C) 0.93 (2E)	Y (0.991-0.946)	66.7
7-a	3-Pose1	A	MS	5	29	3168	71	0.981	0.99 (A,C) 0.89 (1H)	Y (0.981-0.888)	28.2
7-b	3-Pose1	A	QS	7	16	12	3	0.989	0.99 (A,B,C,1D,2D) 0.92 (1H)	Y (0.989-0.888)	28.2
8-a	3-Pose1	None	MS	7	22 (f)	48	8	0.976	0.99 (C) 0.9 (A)	Y (0.976-0.867)	37.5
8-b	3-Pose1	None	QS	10	21	12	1	0.989	0.99 (1D,2D,B,C) 0.93 (1H)	Y (0.989-0.867)	28.1
9	3-Pose2	None	MS	8	22 (f)	48	8	0.979	0.99 (2G,2I,C,F) 0.89 (1H)	Y (0.979-0.869)	37.5
10	3-Pose3	None	MS	7	23 (f)	144	8	0.971	0.99 (1D,C,F) **0.03** (2H)	N (0.982-0.953)	-
11	3-Pose4	None	MS	6	30	23688	292	0.941	1.0 (1D) **0.0** (2I)	N (0.987-0.909)	-
12	3-Pose4	None	MS	6	20 (f)	216	6	0.987	1.0 (1D) 0.89 (1H)	Y (0.987-0.976)	66.7
13	3-Pose5	None	MS	6	21 (f)	216	10	0.982	0.99 (1D,C) 0.79 (1H)	Y (0.982-0.955)	40.0
14	3-Pose6	None	QS	40	19 (f)	24	5	0.929	0.99 (2D,C) **0.0** (1G)	N (0.932-0.905)	-

identified, the less correctly identified ones being 'text' and 'paper' (smallest similarity indices). Note that a (almost) completely unrecognized entity involves a similarity index close to 0. Among the 3 possible common subgraph isomorphisms, the automatically selected one is the best one (i.e. highest GCR). Only 66.7% of the common subgraph isomorphisms (i.e. 2 among the 3 possible ones) lead to the best result, while selecting the less appropriate common isomorphism would lead to a GCR of 0.994.

In most cases, our proposal allows to correctly merge segmented regions and finally identify each one. As illustrated by surrounded area in test1-b of figure 8, apart from recognition, our method often allows to improve the segmentation: regions are merged according to both a priori inclusion and photometric, involving the removal of some artefacts. The worst result is the one related to test #14 (see figure 9-test 14). In our sense, this is mainly due to the initial oversegmentation. Due to the lighting inhomogeneity, the progressive merging of initial labels (obtained by quickshift) involves the merging of labels that correspond to distinct expected regions (surrounded in figure 9-test 14). In our sense, the difficulty should be overcome by trying to improve the initial segmentation first, this being out of the scope of that work.

Concerning image 2, some liver tumors lay at liver boundary, and an enveloppe is considered to ensure that tumors belong to liver in the built inclusion graph. This can be seen as an intrinsic limitation of our proposal that assumes perfect inclusion. For the test #3-b, liver tumors are not recognized, and are uncorrectly merged with liver tissues. It has been visually oberserved that this results from the combination of two aspects. First, the well-known volume partial effect at liver boundaries leads to small regions

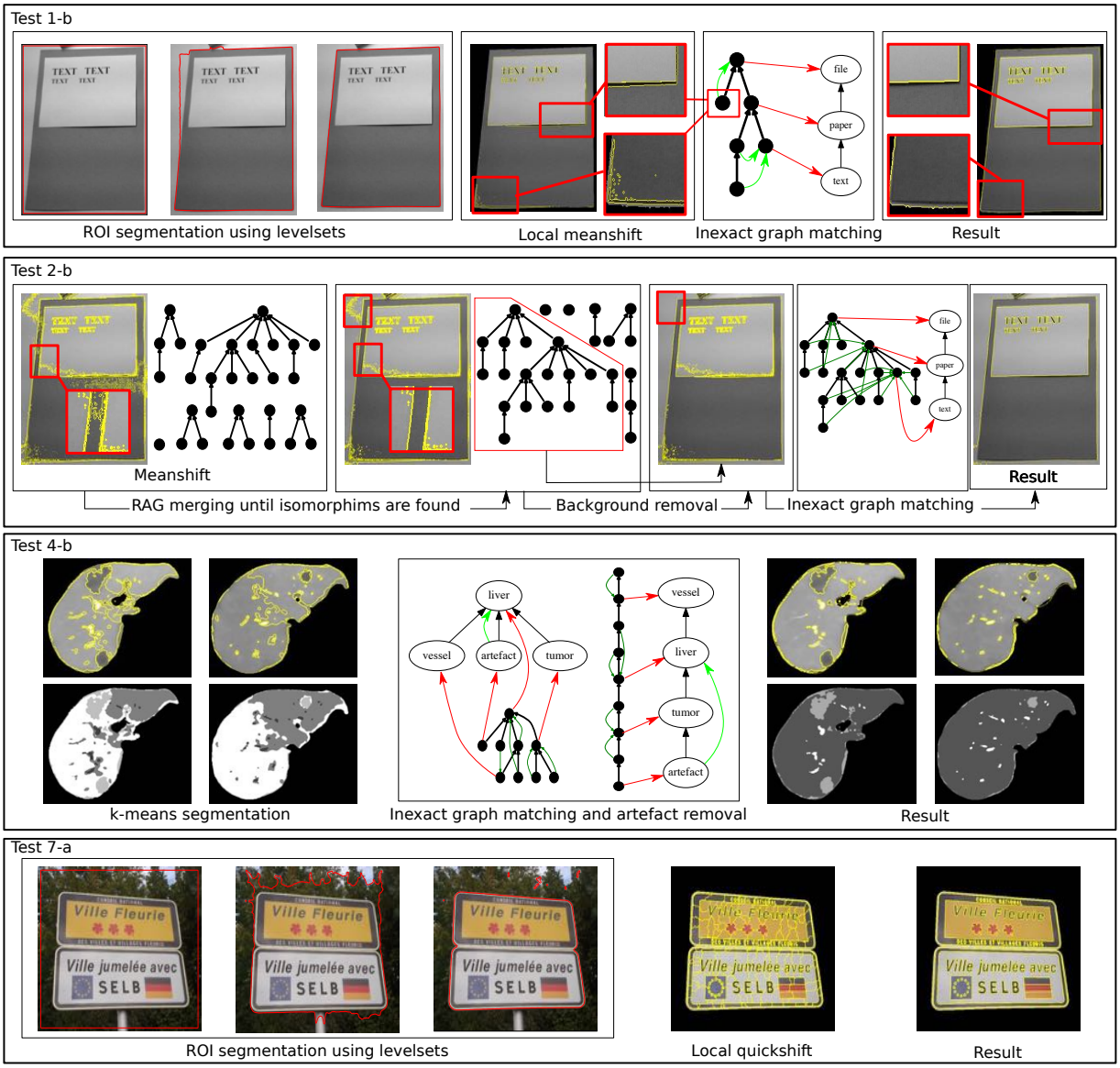

Fig. 8. Details of some experimental results considered in table 1. Concerning inexact graph matching, red arrows represent the initial subgraph isomorphisms and green ones regard nodes (regions) merging. In most cases, only inclusion relationships are reported for clarity.

(artefacts) significantly darker than tumors that are merged with liver tissues (closer mean intensity). Second, meanshift typically produces isolated points that are identified as tumors. As illustrated by test #3-c, this can be overcome in practice by removing small regions before computing ismorphisms. Note that such well known limitation of meanshift is also observed in many tests involving image 3 (overcome using filtering).

Test #4 involves k-means, leading to an initial segmentation that is significantly worse than with meanshift, due, in particular, to the inhomogeneity of the density of liver tissues (see test 4-b figure 8). In such a case, the filtering of small regions related to artefacts is not sufficient. To overcome this difficulty, we evaluate the ability to temporarly declare artefacts in the model (this being not necessary using meanshift), before merging them with liver tissues (see green arrows between 'artefact' and 'liver' in test 4-b of figure 8). In this case, artefacts, resulting from the volume partial effect, are removed by reasoning on the model (and produced identification) rather than by improving the initial over-segmentation algorithm.

A last aspect to be shortly discussed concerns the use of the ROI resulting from levelsets. Considering the entire image ('none' ROI) leads to slightly worse

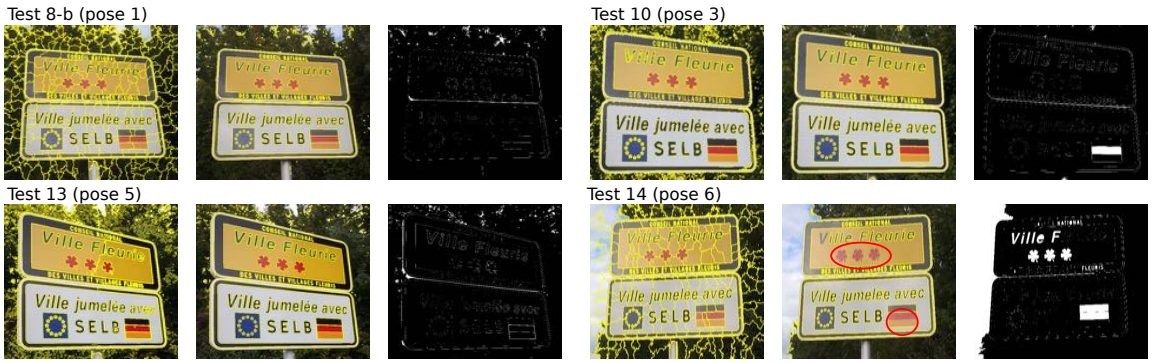

Fig. 9. Some results regarding poses of image 3: initial segmentation, result and difference with truth.

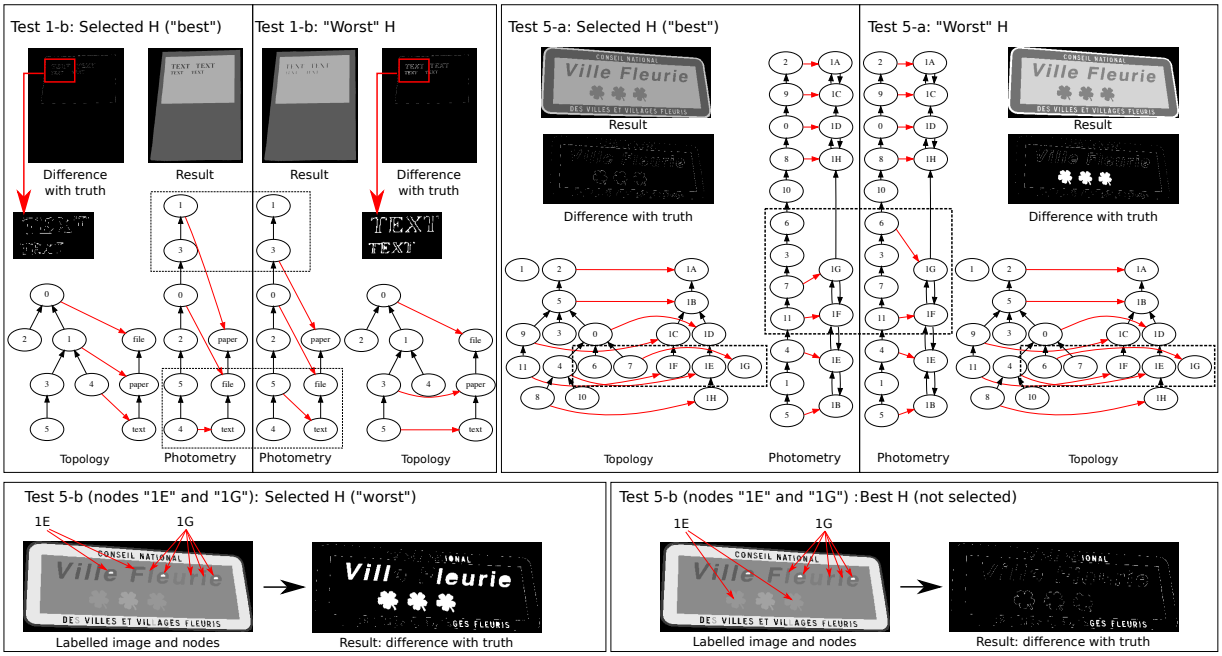

Fig. 10. Influence of the choice of the initial common subgraph isomorphism in some cases reported in table 1.

results, but mainly concerning the head of the inclusion graph. This is illustrated for region 'A' (see tests #7-a, #7-b and #8-a in table 1, test 7-a in figure 8 and test 8-b in figure 9).

3.3.2 Selection criterion

In most cases, the common subgraph isomorphism is appropriately selected among, in some cases, a relatively large number of possibilities (see, in table 1, tests #1-c, #2-c, #6-a and #7-a). In some cases, a minority of possible isomorphisms lead to the best result. Figure 10 provides two examples illustrating the benefits of appropriately selecting the isomorphism (test #1-b and #5-a).

As reported in table 1, the common isomorphism is uncorrectly selected in some situations (tests #3-b, #4-a, #5-b, #10, #11, #14). Note that, in many cases, as shown by GCRs, even if the best H is not selected, the considered criterion does not select the worst one (see tests #4-a, #10, #11 and #14).

For the test #3-b (overcome by filtering - see test #3-c) and the test #4-a, this results from the volume partial effect at liver boundaries, leading to small regions (artefacts) significantly darker than tumors. Because the selection criterion favors large average distances between graph nodes, dark artefacts are considered as tumors, that, in turns, are merged with liver tissues, appearing photometrically closer. Test #11 regards a similar situation (isolated points typically resulting from meanshift [18]), that can be overcome by filtering (see test #12).

For the test #5-b, the problem is due to the fact that an expected entity may be split into several labels or subparts, each one being finally assigned to a priori distinct regions. As illustrated in figure 10 (test 5-b

- left), two subparts of the expected region 1E are respectively assigned to regions 1E and 1G when computing common subgraph isomorphisms. The considered selection criterion favors such an isomorphism because both (a priori similar regions) share close mean intensities. This finally leads to an unperfect recognition. Generally speaking, such a situation is likely if expected regions are composed of several connected components and if spatial information is considered for segmentation (quickshift in our case). Note that, as illustrated by the right part, considering another isomorphism can lead to a better result. To overcome this problem, a solution is to reduce the number of labels (see test #5-c reported in table 1), by preliminary (before computing isomorphisms) assigning the same label to separated connected components than share similar mean intensities.

TABLE 2
Measured runtimes (in seconds) for some cases reported in table 1, regarding the segmentation (Seg), initial graph building (Build), the computation of common subgraph isomorphisms (Iso.) and merging (Merge). Some of them are negligible versus the measurement accuracy.

#	Seg	Proposed method			
		Build	Iso.	Merge	Total
1-a	3.82	0.16	0.0	0.03	0.19
1-b	3.0	0.2	0.0	0.19	0.39
1-c	7.69	2.11	0.0	14.82	16.93
2-a	10.32	0.63	0.0	0.34	0.97
2-b	3.55	1.01	0.0	0.83	1.84
2-c	7.87	1.78	0.0	1.56	3.34
3-a	51.81	17.45	0.0	4.17	21.62
3-c	59.71	23.43	0.01	5.74	29.18
4-b	132.13	26.1	0.0	5.18	31.28
5-a	1.15	0.44	0.0	0.23	0.67
5-c	7.23	0.39	0.0	0.18	0.57
6-a	1.45	1.57	0.21	5.86	7.64
6-b	7.05	0.43	0.0	0.2	0.63
6-c	6.57	0.4	0.0	0.12	0.52
7-a	2.51	1.82	0.9	5.67	8.39
7-b	7.13	0.67	0.01	0.27	0.95
8-a	5.38	3.99	0.31	3.31	7.61
8-b	6.79	1.9	0.04	0.86	2.8

For the test #10, this is due to a segmentation artefact (at the expected 2H region boundaries) that is too large to be remove by a simple filtering (i.e. without altering the overall segmentation). Note that, althought the selection criterion does not select the best isomorphism (GCR of 0.982 instead of 0.971), it avoids selecting the one leading to an even worse result (GCR of 0.953). Indeed, as illustrated by figure 9, only region 2H is not recognized (the other 12 regions being correctly recognized).

For the test #14, even if the selection is not optimal, it excludes isomorphisms leading to worse results (0.905 reported in table 1). Note that this image (image 3-pose 6) would require a more sophisticated initial segmentation algorithm. Indeed, quickshift involves the initial merging of region 1G with 1D, and 2H with 2D (surrounded in figure 9).

3.3.3 Runtime

Table 2 reports some measured runtimes (on a MacBookPro with a Core I7 2.2 GHz Intel processor). In most cases, the proposed method is less time consuming than the preliminary segmentation step. The computation of common subgraph isomorphims appears negligible. Although not significantly time consuming in most cases, it appears that graph building from labelled image is not negligible, in particular for image 2. This is due, in particular, to computations that are required to retrieve inclusion relationships (described in section 2.2), involving large regions. The runtime related to the merging step strongly varies from one situation to another. For image 2, this is due to the image size which increases the runtime when combining large regions. In the other cases, this results from the size of built graphs which increases the number of tests performed when evaluating the validity of merges (see pseudo-code 5).

The influence of the number of nodes N (scalability, limited to 31 nodes in experiments) on the runtime is studied in the particular case considered in figure 11. The synthetic image (3 nested regions) is corrupted by artefacts (similar to noisy pixels), increasing the number of input nodes (see images in figure 11). According to measured runtimes, the observed experimental complexity seems, in this example, to be quadratic with respect to N (runtime is in logarithmic scale), giving an idea about the impact of a strong over-segmentation. In our sense, considering so many nodes (e.g. 1000) is probably not realistic in practice, and can be avoided by controling the initial oversegmentation (e.g. early removal of some irrelevant regions or use of a ROI). Measured runtimes also show that both building and merging steps are more time consuming than the computation of subgraph isomorphisms, due to underlying operations performed on images (e.g. morphological operations) that come in addition to graph-based operations. In our sense, both steps could be optimized by considering local computations rather than global ones. For instance, when building inclusion relationships using the iterative procedure described in section 2.2, adjacency matrices related to inclusion and intersection relationships are entirely recomputed at each iteration. Similarly, when merging regions, one checks that the overall coherence is not broken by recomputing subgraph isomorphisms: this could be done by analysing the local impact of merging two nodes.

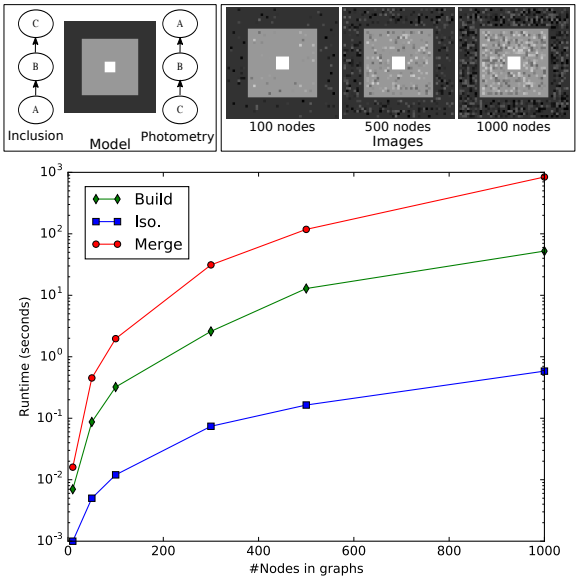

Fig. 11. Influence of the number of nodes on the runtime. Top: *Model* regards a priori relationships and the synthetic image. *Images* are some of considered images, for various number of nodes (corresponding to artificially added artefacts, appearing as noisy pixels). Bottom: runtime (logarithmic scale) for various number of nodes, and for each step (graphs building, subgraph isomorphism computation and final merging).

4 Discussion

Experimental results show the potential of the proposed method to efficiently extract and identify regions, using a priori qualitative description of the image content, and without any input parameters. Even the objective function allowing to select the best initial isomorphism does not require input parameters (e.g. compared to [6], where a parameter governs the relative influence of the appearance information versus the structural one). Our approach assumes that inclusion relationships are preserved between various instances (i.e. images) of a priori model. Althought this can be overcome in some cases (e.g. as illustrated for image 2), this is the main limitation of our proposal.

For a practical use, some post processing could be required to refine the interpretation of the image content (e.g. separating letters from black strip in image 3, for instance for text recognition purposes), depending on the application. Indeed, our proposal only manages the identification of structures that are distinct either in term of photometry or inclusion.

A priori model can be easily built from a qualitative description, as illustrated in the provided Python implementation, where a priori graphs are defined by two strings. For instance, the model of image 1 is defined by *"text<paper<file"* for inclusion (where operator $<$ regards the relation in terms of inclusion), and *"paper<file<text"* for photometry (where operator $<$ concerns the relation between mean intensities). A priori model can also be automatically defined by providing manually segmented example(s), with identified regions. Severals examples are recommended if photometric relationships are not constant from one situation to another (i.e. leading to similarities), as illustrated by image 3 (see figure 7). This is a strength of our approach, compared to related graph based approaches requiring statistical parameter learning (on a large database), as recently considered for (quantitative) relative positions [9].

Concerning photometric uncertainty, the declaration of a priori similarities results, in our sense, from a tradeoff. From one side, enough similarities should be declared to manage photometric uncertainty (to be robust enough versus variability encountered in real images). On this other side, the number of similarities should be minimized to ensure the model validity (see section 2.6), this depending on inclusion relationships.

Experiments show a certain degree of robustness of our approach with respect to the initial segmentation, even in the case of a strong over segmentation producing regions initially depicting inclusion (and photometric) relationships that do not correspond to the a priori model. As illustrated, by guiding a region merging procedure with subgraph isomorphisms, it appears possible to reduce the oversegmentation so that at least a common subgraph isomorphism is found. Nevertheless, the limit of the procedure is that a too strong reduction may involve the merging into a single of several distinct expected regions, leading to recognition errors (as illustrated for test #14). In our sense, this also underlines that the initial segmentation must be appropriately chosen (depending on the application) to avoid such a situation.

Concerning the selection criterion, although experiments show the relevance of our promising proposal, an improvement would concern the establishment of a single criterion instead of the two considered (each being dedicated to a particular situation). More accurately, it would be interesting to find an objective function managing both situations, i.e. maximizing the regularity of the spacing of matched nodes while simultenaneouly mimizing the one related to a priori similar nodes.

This work focuses on the formalism, and on some preliminary experimental results, provided together with the freely available experimental database and software implementation. In our sense, this provides a good understanding of our proposal, as well as, of its strengths and weaknesses. Using the provided implementation, it would be interesting to perform additional studies, dedicated to specific applications, on large databases. Apart from previously mentionned improvements to be investigated, it would also interesting to extend this method (and its software implementation) in order to also manage of unexpected

objects (as considered in [23]) as well as inclusion uncertainty.

5 Conclusion

This paper proposes an inexact graph matching method for image interpretation from a priori qualitative knowledge based on inclusion and photometry relationships. Experiments illustrate the relevance and potential of this method to extract and identify regions from an oversegmented image without any quantitative input parameters (except input information resulting from the visual perception of the image content). An implementation of the method is provided, as freely available Python package, to be used by the community for a deep understanding of the core mechanism and for further evaluations and improvements.

As future step, further studies would be useful to evaluate the potential of this proposal for specific applications. Another future step concerns the extension of our method with additional a priori qualitative information such as other spatial, textural, shape or even size information.

References

[1] B. Peng, L. Zhang, and D. Zhang, "A survey of graph theoretical approaches to image segmentation," *Pattern Recognition*, vol. 46, no. 3, pp. 1020 – 1038, 2013.
[2] O. Lézoray and L. Grady, *Image processing and Analysis with Graphs: theory and practice*. CRC Press, 2012.
[3] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty Years Of Graph Matching In Pattern Recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 3, pp. 265–298, 2004.
[4] S. Iodice and A. Petrosino, "Salient feature based graph matching for person re-identification," *Pattern Recognition*, vol. 48, no. 4, pp. 1074 – 1085, 2015.
[5] K. Madi, H. Seba, H. Kheddouci, and O. Barge, "A graph-based approach for kite recognition," *Pattern Recognition Letters*, pp. –, 2016.
[6] A. Noma, A. Graciano, R. Cesar Jr, L. Consularo, and I. Bloch, "Interactive image segmentation by matching attributed relational graphs," *Pattern Recognition*, vol. 45, no. 3, pp. 1159 – 1179, 2012.
[7] X. Gao, B. Xiao, D. Tao, and X. Li, "A survey of graph edit distance," *Pattern Analysis and Applications*, vol. 13, no. 1, pp. 113–129, 2010.
[8] R. Cesar, E. Bengoetxea, I. Bloch, and P. Larrañaga, "Inexact graph matching for model-based recognition: Evaluation and comparison of optimization algorithms," *Pattern Recognition*, vol. 38, no. 11, pp. 2099 – 2113, 2005.
[9] O. Nempont, J. Atif, and I. Bloch, "A constraint propagation approach to structural model based image segmentation and recognition," *Information Sciences*, vol. 246, no. 0, pp. 1–27, 2013.
[10] A. Deruyver and Y. Hodé, "Qualitative spatial relationships for image interpretation by using a conceptual graph," *Image and Vision Computing*, vol. 27, pp. 876–886, 2009.
[11] M. Mignotte, "Segmentation by fusion of histogram-based k-means clusters in different color spaces," *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 780–787, 2008.
[12] C. Hudelot, J. Atif, and I. Bloch, "Fuzzy spatial relation ontology for image interpretation," *Fuzzy Sets and Systems*, vol. 159, pp. 1929–1951, 2008.
[13] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "A (sub)graph isomorphism algorithm for matching large graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1367–1372, 2004.
[14] F. Serratosa and X. Cortés, "Graph edit distance: Moving from global to local structure to solve the graph-matching problem," *Pattern Recognition Letters*, vol. 65, pp. 204 – 210, 2015.
[15] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.
[16] IRCAD, "3D image reconstruction for comparison of algorithm database," 2017. [Online]. Available: http://www.ircad.fr/research/3d-ircadb-01/
[17] Y. Chen, Z. Wang, J. Hu, and W. Z. Q. Wu, "The domain knowledge based graph-cut model for liver CT segmentation," *Biomedical Signal Processing and Control*, vol. 7, no. 6, pp. 591–598, 2012.
[18] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
[19] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *In European Conference on Computer Vision, volume IV*, 2008, pp. 705–718.
[20] J.-B. Fasquel, V. Agnus, J. Moreau, L. Soler, and J. Marescaux, "An interactive medical image segmentation system based on the optimal management of regions of interest using topological medical knowledge," *Computer Methods and Programs in Biomedicine*, vol. 82, pp. 216–230, 2006.
[21] J.-B. Fasquel, G. Brocker, J. Moreau, N. Papier, V. Agnus, C. Koehl, L. Soler, and J. Marescaux, "A modular and evolutive component oriented software architecture for patient modeling," *Computer Methods and Programs in Biomedicine*, vol. 83, no. 3, pp. 222 – 233, 2006.
[22] A. Trémeau and P. Colantoni, "Regions adjacency graph applied to color image segmentation," *IEEE Transactions on Image Processing*, vol. 9, pp. 735–744, 2000.
[23] A. Deruyver, Y. Hodé, and L. Brun, "Image interpretation with a conceptual graph: Labeling over-segmented images and detection of unexpected objects," *Artificial Intelligence*, vol. 173, pp. 1245–1265, 2009.

Jean Baptiste Fasquel was born in 1975. He received his PhD in computer science from the University of Strasbourg, France, in 2002. From 2002 to 2009, he has been a permanent researcher at the Research Institute Against Digestive Cancer IRCAD, Strasbourg. Since September 2009, he has been an Associate Professor at the University of Angers. His main areas of interest are image processing and software engineering, mainly applied to medical imaging.

Nicolas Delanoue Nicolas Delanoue was born in Angers, France, in 1980. He received the Ph.D. degree in mathematics from the University of Angers, France, in 2006. In 2008, he joined the Department of Electrical Engineering, University of Angers, as an associate Professor. His main areas of research interest are singularities and numerical methods with applications in robotics, computational topology and stable mapping classification.

Research Article Journal of the Optical Society of America A 1

An approach for sequential image interpretation using a priori binary perceptual topological and photometric knowledge and k-means based segmentation.

JEAN-BAPTISTE FASQUEL[1,*] AND NICOLAS DELANOUE[1]

[1] *LARIS laboratory, EA4094, University of Angers, 62 avenue Notre Dame du Lac, 49000 Angers, France*
[*] *Corresponding author: Jean-Baptiste.Fasquel@univ-angers.fr*

Compiled April 16, 2018

The proposed approach exploits a priori known qualitative inclusion and photometric relationships between image regions, represented by oriented graphs. Our work assumes a sequential image segmentation procedure where regions are progressively segmented and recognized by associating them with corresponding nodes in graphs related to the prior knowledge. The main contribution concerns the parameterization of the k-means clustering algorithm, to be used during the segmentation procedure, and the graph-matching-based identification of resulting clusters, corresponding to regions declared in graphs. The parameterization of k-means is based on known relationships as well as on regions which have been segmented and recognized at previous steps. Parameters are the region of interest within which k-means clustering is constrained, the number of clusters and seeding constraints. Photometric relationships built from resulting clusters are matched with a priori known relationships to identify each cluster, this being formulated as an exact graph matching problem. The potential of this approach is studied on four use cases involving real grayscale and color images with dedicated sequential analysis procedures. Processing results are compared with those obtained without the proposed parameterization of k-means, as well as with some other clustering approaches. Results show the relevance of our approach, in particular in terms of segmentation accuracy, computation time and seeding reliability.

OCIS codes: *(100.2960) Image analysis; (100.2000) Digital image processing; (100.5010) Pattern recognition.*

http://dx.doi.org/10.1364/ao.XX.XXXXXX

1. INTRODUCTION

The use of a priori knowledge for image interpretation has been widely studied. Most supervised approaches are based on *quantitative* knowledge at low-level (pixel-level), often related to statistical theory (e.g. statistics related to the photometry of some image regions) [1]. As recently underlined [2], a limitation of such approaches regards the requirement of a representative, and often large, training data set allowing to model a priori knowledge. To overcome this limitation, an alternative is to favor *qualitative* knowledge, resulting from the visual perception of the scene content [1]. As illustrated in Figure 1, considered qualitative knowledge regards *binary* inclusion relationships between regions (e.g. liver tumor is included within the liver) and photometric ones (e.g. liver tumors are darker than liver tissues). Concerning color images, photometric relationships are considered in the HSV space, based on human color perception [3]. As illustrated in Figure 1, the mean hue of the red painting is lower ("Low Hue") than the one of the yellow painting being ("Medium Hue"), in turns lower than the one of the blue painting ("High Hue"). Additionnally, black paintings have a lower value (V component) than white and colored ones, which are, in turns, more saturated (S component) than white paintings. Such an approach is expected to be generic because related models are assumed to be weakly sensitive to variations observed in real images. For instance, one expects that a hypodense liver tumor remains darker than liver tissues whatever the patient, even if its mean intensity may vary from one patient to another. The counterpart of such a genericity regards the difficulty of translating such high-level knowledge into relevant and usable numerical values at low-level (i.e. algorithm parameters) [1, 2].

To our knowledge, most related works are limited to spatial relationships [2, 4, 5], sometimes together with photometric information [4, 6], but in a quantitative manner (e.g. quantified relationships and fuzzy logics [4]). Purely qualitative knowledge have been recently proposed but limited to spatial relationships [1, 7]. In this paper, we consider of a sequential image analysis procedure [2, 7–9], where regions are progressively segmented

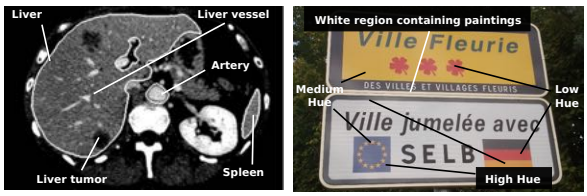

Fig. 1. Image examples. Medical image (left) : CT Scanner image (abdominal area). Color image (right): the sign depicts two white parts containing paintings of a priori known colors.

and recognized. One starts with simplest regions (e.g. spleen, artery and liver in Figure 1) and then exploits them for finding and identifying complex ones (e.g. liver content) [9]. A difficulty is to combine prior knowledge and knowledge resulting from already recognized regions. To address this issue, we consider the parameterization of the *k*-means clustering algorithm, being widely used for image segmentation [3, 10–14]. Additionally, recognition of resulting segmented region is also adressed by reasoning on relationships using a graph matching approach [15–17].

The first part of the contribution regards relationships allowing to translate this high-level knowledge into usable parameters for *k*-means. The second part concerns the recognition of resulting segmented regions, based on graph matching.

An overview is given in Section 2. Qualitative knowledge and sequential analysis are detailed in Section 3. Sections 4 and 5 respectively concerns the parameterization of *k*-means and the recognition of resulting regions. Section 6 focuses on experiments. Before concluding, our work is discussed in Section 7.

2. METHOD OVERVIEW

Figure 2 provides an overview of our proposal, using a synthetic image processed by a particular sequence (Figure 2-A). Prior knowledge (Figure 2-B), detailed in Section 3, regards a priori known inclusion and photometric relationships between regions. The context knowledge (Figure 2-C), also detailed in Section 3, stores progressively recognized regions (region 0, then 2 and finally 1). Our proposal focuses on the two major aspects:

- The parameterization of *k-means* (Figure 2-D, detailed in Section 4), involving the dataset to be considered (ROI in Figure 2-D), the number of clusters (2 in Figure 2-D) and the seeding (initial cluster centroids).
- The recognition of regions resulting from the clustering, based on graph matching (Figure 2-E, detailed in Section 5).

3. KNOWLEDGE AND SEQUENTIAL ANALYSIS

A. Knowledge

We consider that an image I is composed of a set of regions $X(u), u \in S$. *A priori* knowledge is modeled by oriented graphs [1]. As illustrated in Figure 3, $G_T = (S, A_T)$ and $G_P = (S, A_P)$ respectively regard inclusion and photometric relationships between regions (the subscript T stands for "topological"). For color images (HSV space), photometric relationships are modeled using the set of graphs $\{G_{P_H}, G_{P_S}, G_{P_V}\}$.

Edges A_T correspond to inclusion relationships, defined by: $i \xrightarrow{T} j \Leftrightarrow X(i) \subsetneq X(j)$. We define the residue $X_r(u)$, corresponding to region $X(u)$ from which included regions are removed (e.g. liver region minus liver tumors and vessels in Figure 1):

$$X_r(u) = X(u) \setminus \Big(\bigcup_{i \in G_T^-(u)} X(i) \Big), \tag{1}$$

where $G_T^-(u)$ is the set of predecessors of the node u, which can be recursively recovered:

$$G_T^-(u) = \bigcup_{\{i \in S | i \xrightarrow{T} u\}} \left(G_T^-(i) \cup \{i\} \right). \tag{2}$$

Edges A_P denote order relations between mean intensities (or mean value along H, S or V components) of region residues, and are defined by: $i \xrightarrow{P} j \Leftrightarrow \bar{X}_r(i) \leq \bar{X}_r(j)$. The entity $\bar{X}_r(i)$ is the mean value of $X_r(i)$:

$$\bar{X}_r(i) = \sum_{p \in X_r(i)} \frac{I(p)}{|X_r(i)|} \tag{3}$$

where $|X_r(i)|$ is the number of points of $X_r(i)$. Note that due the conic shape of the HSV space with the circularity of Hue value $h \in [0, 2\pi[$ [3], the corresponding graph G_{P_H} depicts a cycle (see Figure 3). Bidirectional edges concern regions sharing similar intensities: $i \overset{P}{\leftrightarrow} j \Leftrightarrow \bar{X}_{r(i)} \simeq \bar{X}_{r(j)}$.

As illustrated in Figure 3, relationships result from a qualitative description of the image content without any quantitative information (description using a "natural" language).

The modeling of the entire image is not absolutely required (illustrated in Section 6). Nevertheless, some modeling constraints have to be verified to ensure the validity of the parameterization (detailed in Section 4):

- the content of each declared region must be specified (e.g. content of region 0 in Figure 3-top),
- total inclusion has to be avoided: this can be overcome by removing nodes corresponding to empty region residues,
- photometric relationships between regions included in the same region must be declared (e.g. 0 and 1 in Figure 3-top).

Note that the situation where a region overlaps another can be managed by splitting the region into several subregions, each one being assigned to a specific graph node, and being included into a single region.

B. Sequential analysis

As illustrated by Figure 2, a sequential interpretation procedure is associated with a *context* knowledge and an index $t \in \mathbb{N}$ (processing step). This knowledge manages already recognized regions using the set S_t (boxed nodes in Figure 2-right): set of nodes corresponding to recognized regions after t processing steps ($S_t \subseteq S$, with $S_0 = \emptyset$ and $S_\infty = S$). We also define $X_t(u)$:

$$X_t(u) = \begin{cases} X(u) & \text{if } u \in S_t \\ \emptyset & \text{otherwise} \end{cases} \tag{4}$$

The similarly indexed residue $X_{r,t}(u)$ can be expressed as:

$$X_{r,t}(u) = X_t(u) \setminus \Big(\bigcup_{i \in G_T^-(u)} X_t(i) \Big) \tag{5}$$

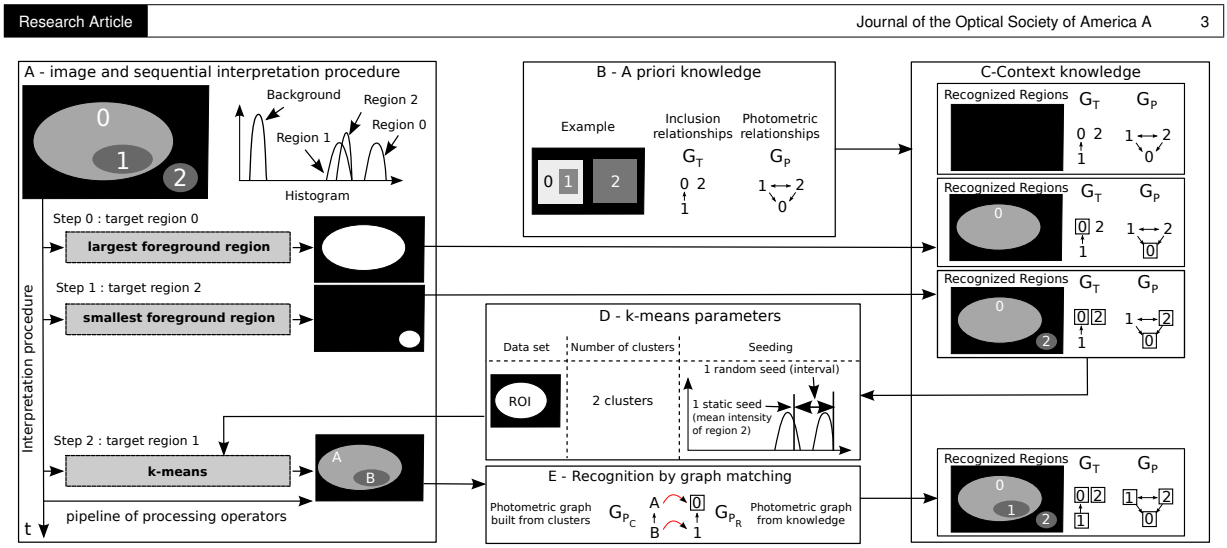

Fig. 2. Method overview. A: image regions are sequentially segmented using a particular sequence, starting with region 0, then 2 and finally 1. B: A priori known relationships. C: *Context* knowledge stores progressively recognized regions (surrounded nodes). D: *k-means* parameters (step 2) are computed from context knowledge: dataset (ROI), number of clusters and seeding constraints. The ROI is the already segmented region 0 because the target region 1, known to be included in region 0 ($1 \rightarrow 0$ in G_T). 2 clusters are considered (regions 0 and 1). The seed associated with region 1 is the mean intensity of the recognized region 2, known to be photometrically similar (i.e. $1 \leftrightarrow 2$ in G_P). The second seed is randomly chosen within an interval because region 0 is declared brighter than region 1 ($1 \rightarrow 0$ in G_P). E: Resulting clusters (regions A and B) are finally identified by exact graph matching, based on photometric relationships: region 1 is then recognized. Context knowledge is finally updated.

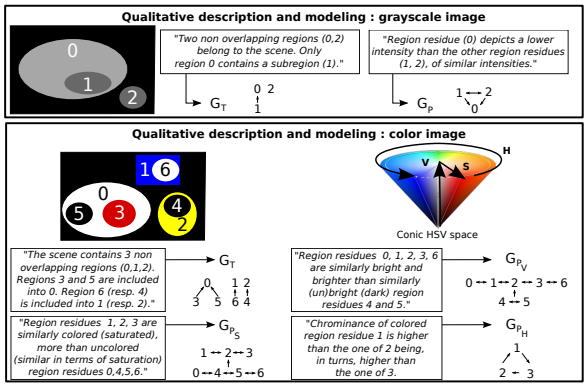

Fig. 3. Qualitative relationships and graphs. Inclusion and photometric relationships correspond to G_T and G_P graphs. A region residue is a region minus included subregions. For the color image, photometric relationships result from the usual representation of colors in the conic HSV space [3].

4. PARAMETERIZATION OF K-MEANS

As recently underlined [18], k-means relies on the minimization the following objective function:

$$\min_{\overline{x}_1,\ldots,\overline{x}_k} E = \sum_{i=1}^{k} \sum_{x_j \in C_i} d(x_j, \overline{x}_i) \qquad (6)$$

where a dataset X (image points) is divided into k disjoint clusters C_i, the center of each one being $\overline{x}_i$. The term $d(x_j, \overline{x}_i)$ regards the dissimilarity between of the image point x_j and the centroid center $\overline{x}_i$. Parameters are the dataset X (region of interest), the number of clusters and initial position of centroids (seeds) [18].

One focuses on the determination of these parameters for segmenting a new target region $X(u)$, such that $u \in S \setminus S_t$, during the $(t+1)^{th}$ analysis step. Sections A, B and C respectively concern the determination of the region of interest, the number of clusters and seeding constraints. For simplification reasons, all relationships involving the graph G_P are applied on the transitive closure of G_P.

Relationships presented hereafter are based on the *root* region within which the target region u is included. The *root* node $r(u)$ is recursively defined by:

$$r(u) = \begin{cases} i & \text{if } \exists i \in S_t \mid u \rightarrow i \\ r(i) & \text{if } \exists i \in S \setminus S_t \mid u \rightarrow i \\ u & \text{otherwise} \end{cases} \qquad (7)$$

Topological successors corresponding to unrecognized regions are recursively visited until the first node corresponding to a recognized region is encountered (root node). The unicity of successors is guaranteed by the tree-like structure of the topological graph.

A. Region of interest

Using inclusion relationships, one can reduce the dataset by ignoring irrelevant image areas (with respect to the target region). This leads to a region of interest (ROI) $R_t(u)$, being the residue of the root region: $R_t(u) = X_{r,t}(r(u))$.

In Figure 2, the ROI at Step 2 is: $R_2(1) = X_{r,2}(0) = X_2(0) \setminus X_2(1) = X(0) \setminus \emptyset = X(0)$, because region 1 is not yet segmented at step 2 (i.e. $X_2(1) = \emptyset$).

In Figure 4, regions of interest are: $R_t(1) = R_t(3) = X_{r,t}(0) = X(0) \setminus (X(2) \cup X(4)$ and $R_t(5) = X_{r,t}(4) = X(4)$. Note that $R_t(3)$ involves the removal of the internal region 4. After recognition, any region would be filled with its already segmented internal regions (region 4 in this case).

In absence of segmented root region ($r(u) \in S \setminus S_t$), a default ROI can be considered:

$$R_t(u) = I \setminus \left(\bigcup_{i \in S} X_t(i) \right) \quad \textbf{(8)}$$

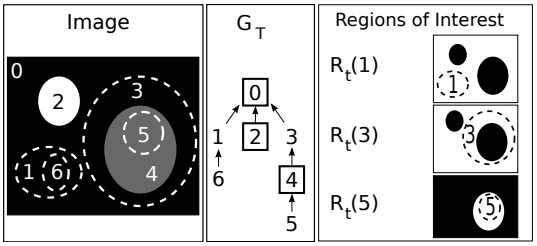

Fig. 4. Image, topological knowledge (G_T) and regions of interest for target regions 1, 3 and 5. Dashed contours surround unidentified regions (left) and target regions (right). According to the context, $S_t = \{0, 2, 4\}$.

B. Number of clusters

We introduce the set $N_{R_t}(r(u))$ of graph nodes corresponding to unrecognized regions appearing within the ROI $R_t(u)$:

$$N_{R_t}(i) = \{i\} \bigcup \left(\bigcup_{\{j \in S \setminus S_t | j \xrightarrow{T} i\}} N_{R_t}(j) \right) \quad \textbf{(9)}$$

N_{R_t} is expressed as a recursive procedure allowing to ignore already recognized regions: predecessors of node i are recursively kept until nodes belonging to S_t are encountered.

In Figure 4, the set related to target regions 1 and 3 is $N_{R_t}(0)$: $N_{R_t}(0) = \{0\} \cup (N_{R_t}(1) \cup N_{R_t}(3)) = \{0\} \cup (\{\{1\} \cup N_{R_t}(6)) \cup \{3\}) = \{0, 1, 3, 6\}$. For target region 5, one has $N_{R_t}(4) = \{4\} \cup \emptyset = \{4\}$.

In Figure 2-step 2, the set is $N_{R_2}(r(1)) = N_{R_2}(0) = \{0\} \cup N_{R_2}(1) = \{0\} \cup \{1\} = \{0, 1\}$

The set of clusters $K_t(i)$ is the subset of $N_{R_t}(i)$ restricted to photometrically distinct regions that would otherwise belong to the same cluster:

$$K_t(u) = N_{R_t}(r(u)) \setminus \{i \in N_{R_t}(r(u)) | \exists j \in N_{R_t}(r(u)), (i < j) \wedge (i \overset{P}{\leftrightarrow} j)\} \quad \textbf{(10)}$$

where the condition $i < j$ avoids counting cluster couples twice. The number of clusters $k_t(u)$ is then: $k_t(u) = |K_t(u)|$.

In Figure 2-step 2, the number of cluster is $k_t(1) = |K_t(1)| = |C_t(0) \setminus \emptyset| = |\{0, 1\}| = 2$.

The determination of $k_t(u)$ requires that the content of any declared region must be entirely modeled, both in terms of topology and photometry (modeling constraint specified at the end of Section A). In absence of identified root region ($r(u) \in S \setminus S_t$), the number of clusters $k_t(u)$ can not be determined.

C. Seeding

Photometric relationships are used to initialize each seed $s(e), e \in K_t(u)$ as close as possible to final cluster centers. These relationships are translated into seeding constraints: each seed is associated with either a fixed value or an interval (see Figure 2-D), as detailed in next Sections C.1 and C.2.

We first propose to introduce the three following sets:

$$\begin{aligned} A^-(e) &= \{i \in S_t \,|\, i \xrightarrow{P} e \wedge (\{j \in S \,|\, j \xrightarrow{T} i\} \subset S_t)\} \\ A^=(e) &= \{i \in S_t \,|\, i \overset{P}{\leftrightarrow} e \wedge (\{j \in S \,|\, j \xrightarrow{T} i\} \subset S_t)\} \\ A^+(e) &= \{i \in S_t \,|\, e \xrightarrow{P} i \wedge (\{j \in S \,|\, j \xrightarrow{T} i\} \subset S_t)\} \end{aligned} \quad \textbf{(11)}$$

The set $A^-(e)$ (resp. $A^=(e)$ and $A^+(e)$) corresponds to recognized regions whose mean intensities that are assumed to be lower (resp. similar and higher) than the one of the region associated with the node e. Regions including unrecognized regions are not considered (incorrect estimates of photometric statistics): all direct topological predecessors of a candidate region i (i.e. $\{j \in S \mid j \xrightarrow{T} i\}$) must correspond to recognized regions (i.e. included in S_t).

C.1. Fixed value based seeding

If $A^=(e) \neq \emptyset$, the seed is set to the fixed value $s(e)$ (mean intensity of the union of residues of already recognized regions depicting similar intensities):

$$s(e) = \overline{\bigcup_{i \in A^=(e)} X_{r,t}(i)}, \text{ if } A^=(e) \neq \emptyset, \quad \textbf{(12)}$$

In Figure 2, $K_2(1) = \{0, 1\}$, $A^=(1) = \{2\}$ and the seed corresponding to region 1 is $s(1) = \overline{\cup_{i \in \{2\}} X_{r,t}(i)} = \bar{X}_{r,2}(2) = \bar{X}(2)$. Because $A^=(0) = \emptyset$, the seed related to the residue of region 0 is defined by an interval.

C.2. Interval based seeding

If $A^=(e) = \emptyset$, $s(e)$ is randomly chosen within an interval $[a_e, b_e]$. To minimize the interval size, we define the lower boundary a_e as the maximum value among following candidate ones:

- Minimal image intensity within the ROI ($\min(R_t(u))$).
- Mean intensity of each darker recognized region residues.

The lower boundary a_e can therefore be expressed as:

$$a_e = \max(\min(R_t(u)), \max_{i \in A^-(e)} (\bar{X}_{r,t}(i))) \quad \textbf{(13)}$$

Symetrically, we define the upper boundary b_e as:

$$b_e = \min(\max(R_t(u)), \min_{i \in A^+(e)} (\bar{X}_{r,t}(i))) \quad \textbf{(14)}$$

In Figure 2, at Step 2, $s(0) = [a_0, b_0]$:

$$\begin{aligned} a_0 &= \max(\min(R_2(1)), \max_{i \in \{2\}} (\bar{X}_{r,2}(i))) \\ &= \max(\min(R_2(1)), \bar{X}_{r,2}(2)) = \bar{X}_{r,2}(2) = \bar{X}_r(2) \\ b_0 &= \min(\max(R_2(1)), \min_{i \in \emptyset} (\bar{X}_{r,2}(i))) = \max(R_2(1)) \end{aligned} \quad \textbf{(15)}$$

The seed $s(0)$ is randomly chosen within $[\bar{X}_r(2), \max(R_2(1))]$. For color images, due the circularity of the Hue component, one may obtain, along this component, an interval $[a, b]$ such as $a \geq b$. In practice, this is managed by considering $[\min(R_t(u), \max(R_t(u)]$ if $a = b$ or $[\min(R_t(u), b] \cup [a, \max(R_t(u)]$ if $a > b$.

In absence of segmented root region ($r(u) \in S \setminus S_t$), seeds are randomly among image points belonging to the ROI.

5. RECOGNITION

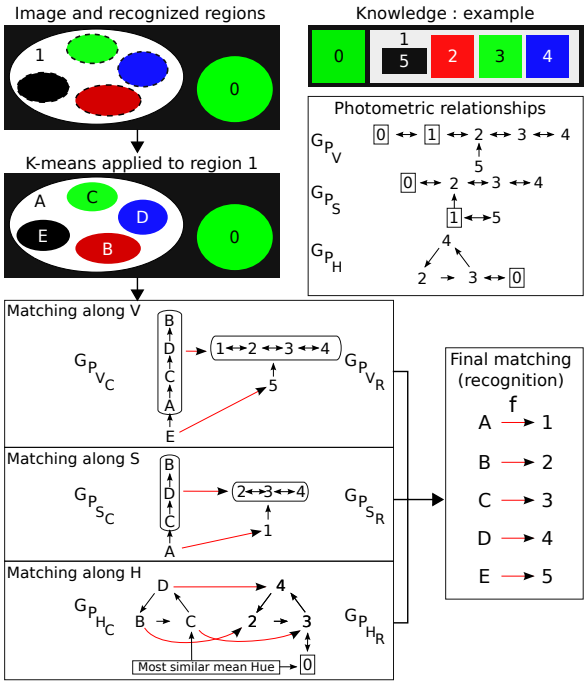

Fig. 5. Recognition and color image (HSV space). Prior knowledge is represented by an image example and photometric relationships. Surrounded nodes correspond to already recognized regions (0 and 1). Applying k-means to the region 1 leads to clusters $\{A, B, C, D, E\}$, identified using $G_{P_{V_R}}$, $G_{P_{S_R}}$ and $G_{P_{H_R}}$ graphs. Dark ($E \rightarrow 5$) and then white ($A \rightarrow 1$) regions are first identified. Note that relationships between color regions along S and V component are an example. Then, exact graph matching identifies remaining colored regions, knowing that region 3 is similar to recognized region 0: The appropriate matching is recovered, among the 3 possible ones (consequence of the circularity along the H component).

As illustrated by Figure 2-E, the recognition of segmented regions (corresponding to resulting clusters) is performed by matching graphs G_{P_c} (the subscript c stands for "clusters") and G_{P_R} (the subscript R stands for "ROI"), related to photometric relationships. The graph $G_{P_c} = (S_c, A_{P_c})$ is built from clusters ($S_c = \{A, B\}$ in Figure 2-D), where edges are obtained by ordering their mean intensities (i.e. cluster centers). The graph $G_{P_R} = (K_t, A_{P_R})$ is the subgraph of G_P related to the ROI, where $K_t \subseteq S$ is defined by equation 10 (photometrically distinct regions belonging to the ROI).

Since $|S_c| = |K_t|$ (consequence of the *k-means* parameterization) and since both graphs have the same structure, one faces an exact graph matching problem [15–17]. Compared to usually encountered inexact graph matching problems, exact graph matching is easier to formulate and faster to solve, even for large graphs (e.g. using VF2 or VF3 algorithms) [16]. As illustrated by experiments in next Section 6, one faces small graphs, therefore facilitating the search of solutions.

Graph matching is based on the search of the isomorphism f between G_{P_c} and G_{P_R}, i.e. $f : S_c \rightarrow K_t$ such that:

$$\forall (i,j) \in S_c \times S_c,\ (i,j) \in A_{P_c} \Leftrightarrow (f(i), f(j)) \in A_{P_R} \qquad \textbf{(16)}$$

In Figure 2-E, the identity of the cluster $i \in S_c$ is $f(i) \in K_t$ ($f(A) = 0$ and $f(B) = 1$). Finding f is facilitated if some unknown regions are photometrically similar to some recognized ones. In Figure 2-E, region 1 is similar to the recognized region 2: the cluster whose mean intensity is the closest to $\bar{X}_r(2)$ (B is the example) is recognized as being 1: the search of f can be guided by this hypothesis (i.e. $f(B) = 1$). If the ROI contains some photometrically similar nodes (i.e. $K_t \neq N_{R_t}$, where N_{R_t} is defined by equation 9), recognition is partial because a cluster can correspond to several identities (i.e. a set of similar nodes). An additional recognition algorithm is then required to identify each entity of this set of similar nodes. This situation is encountered in experiments for color images.

For color images, recognition exploits photometric relationships along H, S and V components to find the matching f (see Figure 5). One searches for dark regions first, if they are present in the ROI (checking the set K_t). This is done by considering, in our case, that $G_{P_{V_R}}$ declares two groups of regions: dark ones (low V value) and bright ones (high V value). The cluster depicting the lowest mean V value is the dark region ($f(E) = 5$ in Figure 5). The same approach is then considered for the S component (the recognized dark region is ignored), assuming white regions (low saturation) and colored ones (high saturation). In Figure 5, this leads to $f(A) = 1$, where the cluster A is the one with the lowest mean S value. An additional recognition algorithm must be considered if more than one dark or white region is present in the ROI (situation encountered in experiments, with the third use case). Note that, if the ROI only contains colored regions, these preliminary steps are useless. Finally, one searches for the isomorphism between $G_{P_{H_c}}$ with $G_{P_{H_R}}$ (ignoring identified black and white clusters). A difficulty concerns the circularity of the Hue component (see Figure 5), leading to several possible isomorphisms (3 possibilities in Figure 5). This limitation can be overcome only if at least one node of the $G_{P_{H_R}}$ is similar to an already recognized region. This is illustrated by Figure 5: the cluster center whose mean H value is the closest to the one of region 0 has the identity of region 3 (because $0 \leftrightarrow 3$). The searched isomorphism f is then the one verifying the constraint $f(C) = 3$.

6. EXPERIMENTS

Experiments aim at illustrating how, and for which benefits, our proposal can be applied to real images. They focus on four use cases, detailed in Sections A to D, and involve 4 real images (Figures 6 and 7). Results are presented in Section E.

Note that, although this paper focuses on *k*-means parameterization and recognition, one also exploits a priori known relationships to guide some third party processing algorithms involved in interpretation procedures (hereafter illustrated).

The dissimilarity between image points i and j is $d(i,j) = ||F(i) - F(j)||^2$ (see equation 6), where $F(i)$ is defined as in [19]. $F(i) = I(i)$ for grayscale images and $F(i) = [S \times V \times cos(H), S \times V \times sin(H), V](i)$ for HSV color images ($I(i) = [H, S, V](i)$), with $H \in [0, 2\pi[$.

A. Use case 1

For this grayscale indoor image, the prior knowledge is established from the observation of the scene content, limited to the

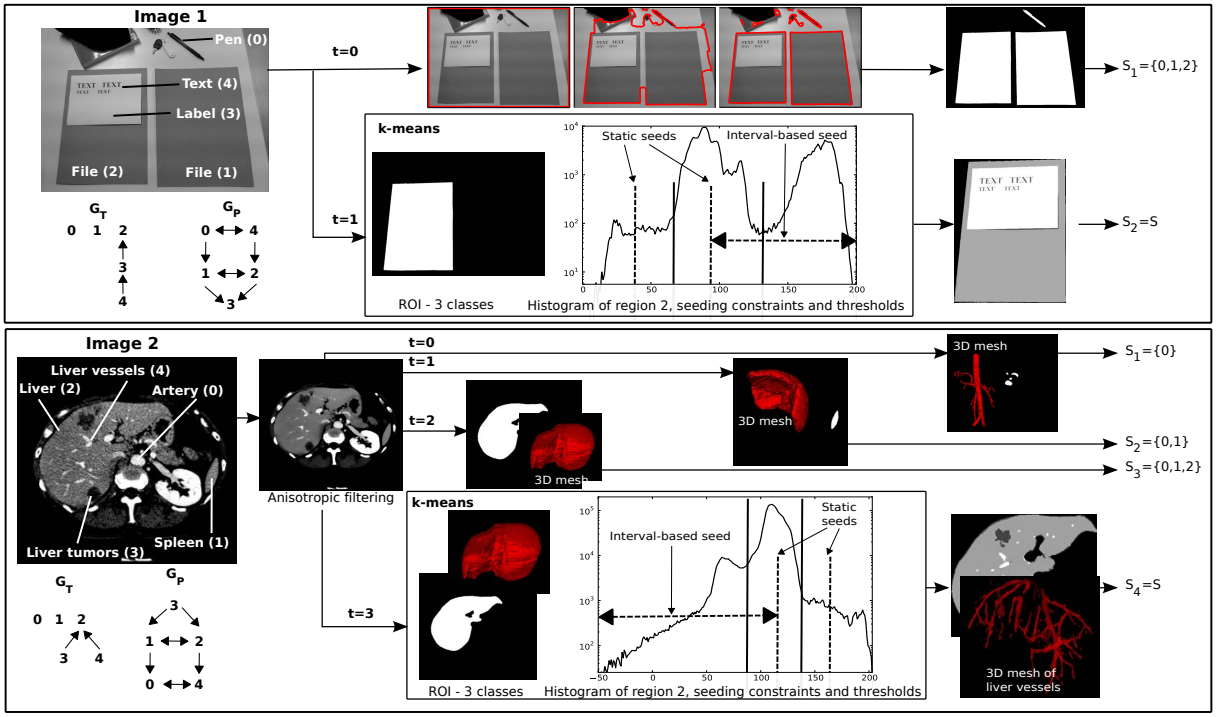

Fig. 6. Use cases 1 and 2: images, a priori knowledge and analysis sequences. For clustering steps, we indicate the automatically determined ROI, number of classes and seeding constraints. Mean values of residues of already segmented regions (seeding constraints) are superimposed on the histogram restricted to the ROI, together with thresholds obtained by *k*-means clustering.

pen and both files (partial qualitative description). Both files are known to be made of the same material depicting a medium tone, compared to both darker text and pen, appearing of similar tone. The label is made of white paper, appearing brighter than any over structure.

The considered processing sequence starts (step $t = 0$) by applying level-set based active contours [20], initialized at image boundaries. The three largest connected components are kept. Region 2 is identified first, assuming that it depicts the largest standard deviation because it is the only one containing subregions of distinct photometry (measured value of 42 against 22 and 8 for the two remaining regions). Remaining regions 0 and 1 are then easily identified (region 0 is the darkest one according to G_P). The second step ($t = 1$) focuses on regions 3 and 4 using *k*-means clustering: the ROI is region 2, containing 3 classes, two of them being associated to a static seeding ($\bar{X}_{r,1}(0)$ and $\bar{X}_{r,1}(1)$), the later one corresponding to an interval-based seeding. Each region is finally identified by exact graph matching.

B. Use case 2

This use case concerns a widely studied application domain, related to computer aided diagnosis which often requires the appropriate segmentation of both anatomical and pathological structures. The image, belonging to an opensource database [1], is a 3*D* medical image (129 slices, 512 × 512 pixels per slice) focusing on the abdominal area and acquired with a CT X-Ray Scanner. Topological knowledge is provided by the knowledge of human anatomy. Photometric relationships (G_P) are established according to the knowledge of the imaging modality and physical properties of tissues (radiodensity). Both liver and spleen parenchyma are assumed to be of similar density [21]. Liver tumors are hypodense structures, therefore darker than liver tissues. The vasculature (i.e. liver vessels and artery in our case, depicting similar densities) is assumed to be brigther than liver and spleen tissues (use of a radiocontrast agent [22]).

After a preliminary noise reduction (anisotropic diffusion), the three first segmentation steps ($t = 0, 1, 2$) aim at interactively segmenting regions 0, 1 and 2 using a dedicated software (see e.g. [7, 23]), because a fully automatic remains a challenging issue [7]. The next segmentation step ($t = 3$) focuses on regions 3 and 4 using *k*-means: the ROI is region 2, containing 3 classes, two of them being associated to a static seeding ($\bar{X}_{r,3}(0)$ and $\bar{X}_{r,3}(1)$), the later one corresponding to an interval-based seeding. As previously, regions are finally identified by exact graph matching.

C. Use case 3

This use case concerns a road sign picture (see Figure 7-top), whose composition is qualitatively described (as in Figure 3-bottom): the sign is made of two parts (top and bottom subsigns), including subparts with a priori known paintings in terms of color, saturation and brightness, leading to G_T, G_{P_H}, G_{P_S} and G_{P_V} graphs. In particular, white and black paintings are "similarly" uncolored (lower saturation than colored paintings, see G_{P_S}). In terms of brightness (G_{P_V}), black paintings are similar, but other

[1] 3D Image Reconstruction for Comparison of Algorithm Database: http://www.ircad.fr/softwares/3Dircadb/3Dircadb.php

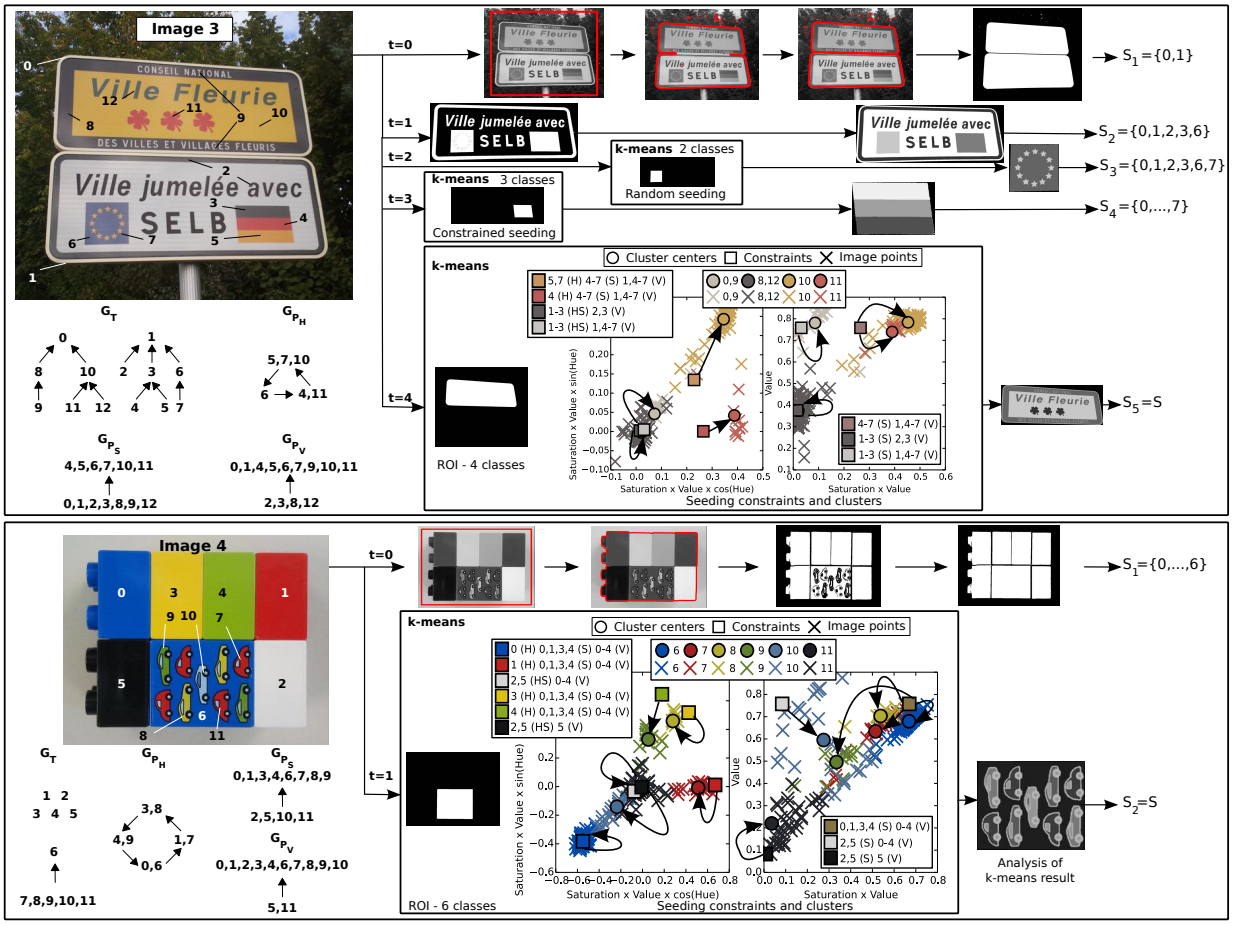

Fig. 7. Use cases 3 and 4: images, a priori knowledge and segmentation sequences (segmentation step t on the left side and evolution of S_t on the right side). For clustering steps, we indicate the automatically determined ROI and number of classes. Automatically computed seeding constraints (mean values of indicated region residues) and resulting clusters (cluster centers and clustered image points are displayed) are represented in the HSV space. For clarity, 2D representations are prefered: H component (left side), and on both S and V components (right side). Arrows indicate links between constraints and resulting cluster centers.

white and colored paintings are similar and brighter. Note that, in terms of topology, region 3 represents the entire flag, being a black region on which two colored strips have been painted.

After a median filtering (noise removal), the segmentation sequence starts ($t = 0$) by applying level-set based active contours to the grayscale-converted image, in order to extract regions 0 and 1 (top and bottom largest connected components).

The second step ($t = 1$) focuses on the content of region 1: foreground objects (compared to the white background) are extracted by a binary thresholding of the grayscale-converted image. Regions 2, 3 and 6 are identified by analyzing foreground objects (connected components): according to G_T and G_{P_V}, regions 3 and 6 depicts highest mean values along V, and region 6 a lower standard deviation than region 3 (numerically verified in this case). Region 2 is to remaining connected components.

The next step ($t = 2$) segments region 7 using k-means: the ROI is region 6, 2 classes are involved, and seeding is randomized. Region 7 is identified by comparison with the mean hue of the inner boundary of region 6.

The next step ($t = 3$) focuses on regions 4 and 5 using k-means: the ROI is region 3, 3 classes are involved, and seeding is constrained by residues of already segmented regions 0, 1, 2, 6 and 7. Regions are identified by graph matching (see Section 5), starting with dark and white regions. Colored regions 4 and 5 are identified by exact graph matching, using G_{P_H} and knowing the mean H value of region 5 is similar to the one of the recognized region 7.

The last step ($t = 4$) concerns the content of the region 0 using k-means: the ROI is region 0 and 4 photometrically distinct classes are involved. Seeding is constrained by the already segmented content of region 1 (fixed value based seeding - see Figure 7-image 3). For instance, yellow region 10 is associated to a static seed (5,7 (H) 4-7 (S) 1,4-7 (V)) in Figure 7-image 3: Along H, this static seed is the mean H value of yellow regions 5 and 7. The dark cluster (lowest mean S value) is first identified as being both regions 8 and 12 (due to photometric similarity),

Table 1. Comparison with other clustering methods (see Figure 8 for segmentations), in terms of good classification rate (GCR), min similarity and runtime. For k-means, the good seeding rate (GSR) and the mean number of iterations are reported. Processing scale (full/low resolution) is indicated.

Image/Step	Measure	Constrained *k*-means	Standard *k*-means	Meanshift	Spectral Clustering	Self-Tuning Spectral Clustering
	Scale	1 (*7*)	1 (*7*)	1 (*7*)	7	7
Image 1	GCR (%)	98.2 (*98.5*)	79.4 (*78.6*)	98.3 (*98.5*)	89.7	98.5
step t=1	Min similarity	0.4 (*0.5*)	0.0 (*0.0*)	0.42 (*0.51*)	0.09	0.5
(3 classes)	Time (s)	0.13 (*0.01*)	0.23 (*0.01*)	6.01 (*0.12*)	34.68	1215.42
	GSR (%) / Iterations	100 (*100*) / 4.3 (*4.1*)	61.0 (*63.0*) / 7.1 (*12.6*)	-	-	-
	Scale	1 (*8*)	1 (*8*)	1 (*8*)	8	8
Image 2	GCR (%)	93.1 (*92.0*)	65.6 (*91.1*)	90.6 (*90.0*)	85.0	4 classes
step t=3	Min similarity	0.6 (*0.51*)	0.09 (*0.0*)	0.0 (*0.0*)	0.43	4 classes
(3 classes)	Time (s)	3.98 (*0.01*)	4.67 (*0.01*)	109.49 (*0.09*)	26.12	916.94
	GSR (%) / Iterations	100 (*100*) / 10.3 (*9.3*)	95.0 (*1.0*) / 11.4 (*11.1*)	-	-	-
	Scale	1 (*3.3*)	1 (*3.3*)	1 (*3.3*)	3.3	3.3
Image 3	GCR (%)	96.1 (*96.7*)	96.1 (*96.7*)	96.1 (*96.7*)	92.5	5 classes
step t=3	Min similarity	0.94 (*0.95*)	0.94 (*0.95*)	0.94 (*0.95*)	0.89	5 classes
(3 classes)	Time (s)	0.06 (*0.01*)	0.13 (*0.01*)	0.96 (*0.10*)	33.46	1031.74
	GSR (%) / Iterations	100 (*100*) / 5 (*4*)	100 (*74*) / 8.9 (*7.1*)	-	-	-
	Scale	1 (*11*)	1 (*11*)	1 (*11*)	11	11
Image 3	GCR (%)	97.9 (*97.7*)	97.9 (*97.7*)	98.3 (*98.2*)	91.7	98.9
step t=4	Min similarity	0.91 (*0.9*)	0.91 (*0.9*)	0.94 (*0.93*)	0.5	0.96
(4 classes)	Time (s)	1.27 (*0.01*)	3.71 (*0.04*)	23.53 (*0.11*)	33.20	913.01
	GSR (%) / Iterations	100 (*100*) / 7.0 (*6.0*)	59 (*62*) / 19.5 (*22.0*)			
	Scale	1 (*3.3*)	1 (*3.3*)	1 (*3.3*)	3.3	3.3
Image 4	GCR (%)	79.8 (*79.8*)	75.3 (*75.7*)	82.8 (*82.8*)	60.5	10 classes
step t=1	Min similarity	0.55 (*0.53*)	0.0 (*0.0*)	0.56 (*0.55*)	0.46	10 classes
(6 classes)	Time (s)	0.15 (*0.02*)	0.47 (*0.05*)	1.20 (*0.17*)	42.81	1301.34
	GSR (%) / Iterations	100 (*100*) / 8.0 (*8.0*)	92 (*89*) / 27.7 (*27.1*)	-	-	-

to be latter distinguished. Similarly, the white cluster is then associated with regions 0 and 9. Remaining colored regions 10 and 12 are identified by exact graph matching, taking into account similarities (along H) with already identified regions 5, 7 and 4. Region 12 is distinguished from region 8 knowing (G_T) that region 12 is included into region 10. Similarly, region 9 is distinguished from region 0, knowing that region 9 is included into the region 8 minus region 10.

D. Use case 4

We consider the segmentation of a toy color picture (see Figure 7-bottom). Topological relationships are built according to the knowledge of the set of cubes (regions 0 to 6), and the nature of the drawings painted on region 6 (leading to subregions 7 to 11). A qualitative description of paintings leads to G_{P_H}, G_{P_S} and G_{P_V}.

The procedure starts ($t = 0$) with level-set based active contours (see Figure 7-image 4). Some basic operators are then applied (not detailed for sake of simplification) to finally obtain the 6 cubes (6 connected components). The largest one is identified as region 6. Then, the 5 remaining regions are recognized using the proposed graph matching technique, starting with dark (region 5) and white (region 2) regions. Remaining colored regions 0, 1, 3 and 4 are identified by exact graph matching, taking into account the similarity of region 0 with the recognized region 6 (mean H value estimated at inner boundaries, because the content of region 6 is not yet segmented).

The second step ($t = 1$) segments the content of region 6 using *k*-means: the ROI is region 6, the number of clusters is 6, each one being associated with a static seeding. Each region is finally recognized by the proposed graph matching approach, taking into account similarities with already recognized regions.

E. Clustering evaluation

This section focuses on the clustering efficiency of some segmentation steps, and on a comparison with results obtained with standard *k*-means, as well as with some other well known approaches, based on mode seeking and graph partitionning [12]: meanshift [24, 25], spectral clustering [19, 26] and self-tuning spectral clustering [27]. Standard *k*-means, meanshift and spectral clustering are manually parameterized to obtain the appropriate number of classes. Self-tuned spectral clustering automatically determines the number of classes, allowing to compare our qualitatively supervised approach with an unsupervised one. Because of the high complexity of spectral clustering (high computation time), related experiments are performed at a low resolution (full resolution for the other clustering algorithms).

Using manually segmentations as ground truth, one measures the good classification rate (GCR), being the percentage of correctly classified pixels. One also computes the smallest similarity index obtained over similarity indices calculated for each segmented region, in order to provide a local measurement (compared to global GCR measurement). Computation time are also measured, in seconds (MacBookPro with a Core I7 2.2 GHz Intel processor).

For *k*-means-based approaches, one considers the Good Seeding Rate (GSR), being the percentage of seedings leading to the best result. This regards the seeding reliability, being the chance of initializing seeds so that the best segmentation is obtained. One also provides the mean number of iterations, related to the number of times the clustering process is run until convergence.

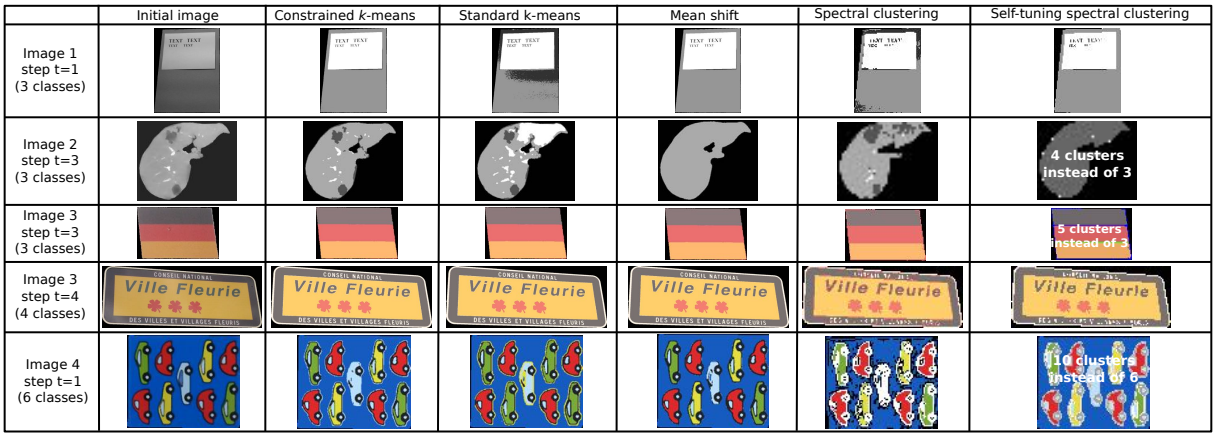

Fig. 8. Segmentations of image regions considered in table 1, for considered clustering algorithms and scales.

Table 1 reports experimental results regarding some clustering steps, and Figure 8 shows corresponding segmentations.

In terms of clustering efficiency (GCR and similarities in table 1), the proposed approach leads either to similar or better results than third party algorithms, for the five considered examples involving grayscale and color images, from 3 to 6 classes. Meanshift is the third party algorithm which appears to be the most efficient except for Image 2: among the 3 obtained clusters, one corresponds to the liver parenchyma, while the two other ones mainly consist in isolated image points (narrow peaks found by meanshift - not appearing in Figure 8 because of their small size). This leads to null min similarities in table 1.

Spectral clustering leads to appropriate results only for images 3 (step 3) and 4. Self-tuning spectral clustering correctly finds the number of classes only for image 1 and image 3 (step 4): for images 2 and 3 (step 3), the number of classes is overestimated (both GCR and similarities are therefore not available).

Standard k-means leads to worst results for images 1, 2 and 4 (yellow and green cars are merged). For the image 3 at step $t = 4$, results are similar to our approach but with a lower reliability (see GSR table 1-image 3-step 4): only 59% of random seedings lead to the appropriate segmentation. Seeding constraints (see e.g. histograms of Figure 6) enable to reduce the search space (i.e. seeding location) to a subspace within which the minimum (local minimum) of E (equation 6) corresponds to the appropriate segmentation.

In terms of computation time (see table 1), the proposed approach appears to be the most efficient one. Compared to standard k-means, runtime reduction is directly related to the number of iterations being, in most cases, smaller with our approach: constrained seeds are closer to final cluster centers than unconstrained seeds. Note that, the runtime is estimated for one seeding: in practice, several random seedings can be required, therefore increasing the runtime. As a consequence, our approach denotes an additional advantage because, due to seeding reliability (related to the GSR), less seedings may be required. For instance, one seeding is sufficient if only static seeds are considered, or in the case of a GSR of 100%.

7. DISCUSSION

Except techniques such as levelsets or anisotropic diffusion, many basic operators are used in experiments (e.g. binary thresholding, connected components analysis), to show the relevance of our proposal. For a specific application, more sophisticated algorithms would probably be required to define an efficient and reproducible interpretation procedure within which our proposal could be incorporated (e.g. clustering requires a preliminary efficient noise reduction, so that clusters can be appropriately separated, as observed for the considered medical image). Apart from the choice of appropriate third party algorithms for a given application, another aspect regards the segmentation strategy (segmentation ordering), to be carefully chosen by starting with "simple" regions providing useful information (e.g. seeding constraints) for recognizing next ones [9].

The initial description of qualitative relationships can be, in practice, captured using user-friendly graphical interface allowing the user to name each region and related relationships without having any technical knowledge about graphs and k-means (i.e. no-expert in computer science). In such a case, the time required for scene modeling depends on the number of regions and on the expert knowledge in the application field. Relationships can also be deduced from properties of image regions in the physical world (e.g. density of tissues), or from a manually segmented representative image example from which relationships can be automatically built. Note that the notion of similarity may require to define a threshold under which regions are declared to be similar (e.g. with respect to the mean intensity difference between regions).

Concerning photometric similarity, it must be underlined that strict numerical equality is unrealistic in practice (e.g. noise in images), as illustrated in experiments where similarity is managed by considering "closest mean values". Similarity focuses on relationships that may vary from one image to another, due to noise for instance, or slight variations of region properties (e.g. density of similar tissues). Such tolerance with respect to the ordering of photometric relationships is therefore useful. Nevertheless, similarity makes recognition, based on edge orientation, more difficult.

Although considered qualitative knowledge is assumed to be

stable (e.g. relative densities of tissues), any mistake in model declaration can lead to a bad estimate of clustering parameters as well as an incorrect recognition of clusters. Nevertheless, as underlined in the literature, this kind of qualitative approach is expected to be more robust than quantitative ones.

8. CONCLUSION

This work provides a generic framework for guiding *k*-means based segmentation and graph-matching-based recognition of resulting regions from a priori known inclusion and photometric relationships expressed in a qualitative manner. Preliminary experiments show how this approach can be applied and illustrate its potential for various application domains involving grayscale or color images.

Perspectives of the work regard the integration of additional qualitative knowledge (e.g. size, location), as well as the parameterization of other segmentation algorithms (not only *k-means*). Next step would concern the evaluation of this approach for specific application domains.

REFERENCES

1. A. Deruyver and Y. Hodé, "Qualitative spatial relationships for image interpretation by using a conceptual graph," Image and Vision Computing **27**, 876–886 (2009).
2. C. Hudelot, J. Atif, and I. Bloch, "Fuzzy spatial relation ontology for image interpretation," Fuzzy Sets and Systems **159**, 1929–1951 (2008).
3. F. Garcia-Lamont, J. Cervantes, A. López, and L. Rodriguez, "Segmentation of images by color features: A survey," Neurocomputing **292**, 1–27 (2018).
4. O. Nempont, J. Atif, and I. Bloch, "A constraint propagation approach to structural model based image segmentation and recognition," Information Sciences **246**, 1 – 27 (2013).
5. O. Colliot, O. Camara, and I. Bloch, "Integration of fuzzy spatial relations in deformable models - application to brain MRI segmentation," Pattern Recognition **39**, 1401–1414 (2006).
6. M. Kobashi and L. Shapiro, "Knowledge-based organ identification from CT images," Pattern Recognition **26**, 475–491 (1995).
7. J.-B. Fasquel, V. Agnus, J. Moreau, L. Soler, and J. Marescaux, "An interactive medical image segmentation system based on the optimal management of regions of interest using topological medical knowledge," Computer Methods and Programs in Biomedicine **82**, 216–230 (2006).
8. A. Moreno, C. Takemura, O. Colliot, O. Camara, and I. Bloch, "Using anatomical knowledge expressed as fuzzy constraints to segment the heart in CT images," Pattern Recognition **41**, 2525 – 2540 (2008).
9. G. Fouquier, J. Atif, and I. Bloch, "Sequential model-based segmentation and recognition of image structures driven by visual features and spatial relations," Computer Vision and Image Understanding **116**, 146–165 (2012).
10. P. Berkhin, *A Survey of Clustering Data Mining Techniques* (Springer Berlin Heidelberg, 2006), pp. 25–71.
11. R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," Pattern Analysis and Machine Intelligence, IEEE Transactions on **34**, 2274–2282 (2012).
12. D. Huang, J.-H. Lai, C.-D. Wang, and P. C. Yuen, "Ensembling over-segmentations: From weak evidence to strong segmentation," Neurocomputing **207**, 416 – 427 (2016).
13. B. Yousefi, J. Fleuret, H. Zhang, X. P. V. Maldague, R. Watt, and M. Klein, "Automated assessment and tracking of human body thermal variations using unsupervised clustering," Applied Optics **55**, 162–172 (2016).
14. L. Zhang, K. Li, Y. Zhang, Y. Qi, and L. Yang, "Adaptive image segmentation based on color clustering for person re-identification," Soft Computing **21**, 5729–5739 (2017).
15. M. Vento, "A long trip in the charming world of graphs for pattern recognition," Pattern Recognition **48**, 291 – 301 (2015).
16. L. Brun, P. Foggia, and M. Vento, "Trends in graph-based representations for pattern recognition," Pattern Recognition Letters pp. – (2018).
17. K. Madi, H. Seba, H. Kheddouci, and O. Barge, "A graph-based approach for kite recognition," Pattern Recognition Letters **87**, 186 – 194 (2017). Advances in Graph-based Pattern Recognition.
18. M. C. Naldi, R. J. G. B. Campello, E. R. Hruschka, and A. C. P. L. F. Carvalho, "Efficiency issues of evolutionnary k-means," Applied Soft Computing **11**, 1938–1952 (2011).
19. J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence **22**, 888–905 (2000).
20. R. Malladi, J. Sethian, and B. Vemuri, "Shape modeling with front propagation: a level set approach," IEEE Transactions on Pattern Analysis and Machine Intelligence **17**, 158–175 (1995).
21. B. A. Birnbaum, N. Hindman, J. Lee, and J. S. Babb, "Multi–detector row CT attenuation measurements: Assessment of intra- and interscanner variability with an anthropomorphic body CT phantom," Radiology **242**, 109–119 (2007).
22. Y. Chen, Z. Wang, J. Hu, and W. Z. Q. Wu, "The domain knowledge based graph-cut model for liver CT segmentation," Biomedical Signal Processing and Control **7**, 591–598 (2012).
23. J.-B. Fasquel, G. Brocker, J. Moreau, N. Papier, V. Agnus, C. Koehl, L. Soler, and J. Marescaux, "A modular and evolutive component oriented software architecture for patient modeling," Computer Methods and Programs in Biomedicine **83**, 222 – 233 (2006).
24. D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence **24**, 603–619 (2002).
25. Y. Guo, A. Şengür, Y. Akbulut, and A. Shipley, "An effective color image segmentation approach using neutrosophic adaptive mean shift clustering," Measurement **119**, 28 – 40 (2018).
26. J. Chen, Z. Li, and B. Huang, "Linear spectral clustering superpixel," IEEE Transactions on Image Processing **26**, 3317–3330 (2017).
27. L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in "Proceedings of the 17th International Conference on Neural Information Processing Systems," (MIT Press, Cambridge, MA, USA, 2004), NIPS'04, pp. 1601–1608.

COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE 82 (2006) 216–230

journal homepage: www.intl.elsevierhealth.com/journals/cmpb

An interactive medical image segmentation system based on the optimal management of regions of interest using topological medical knowledge

Jean-Baptiste Fasquel, Vincent Agnus, Johan Moreau, Luc Soler, Jacques Marescaux*

IRCAD, 1 Place de l'Hôpital, F-67091 Strasbourg, France

ARTICLE INFO

Article history:
Received 19 December 2005
Received in revised form
8 April 2006
Accepted 10 April 2006

Keywords:
Medical image processing
Regions of interest
Hierarchical segmentation
Topological structural model

ABSTRACT

This paper presents an original interactive system for efficient medical image segmentation in computer aided diagnosis. The main originality concerns the method used to manage, according to an a priori topological-based structural model, regions of interest (ROIs) within which computations can be constrained. The goal is then to avoid the processing of irrelevant image points, therefore improving and accelerating segmentations. In the case of a hierarchical modeling procedure, our ROI management method enables, for delineating a given medical structure, to optimally determine image points of interest by taking previously segmented structures into account. We propose a mathematical formulation of the method as well as a possible implementation within an interactive system. We also detail an experience report focussing on the segmentation of several abdominal structures from a CT image. It illustrates the behavior and the potential of our method.

1. Introduction

Modern and future surgery rely on medical images and softwares to assist practitioners. For instance, this concerns computer-aided diagnosis, surgical planning, simulation and robotically assisted surgical interventions [1–3]. In most cases, the modeling of anatomical and pathological structures of patients from their medical images is required, to facilitate visualization for instance. Patient modeling is based on an accurate analysis and segmentation of noisy and distorted medical images to find the real contours of the various structures [1].

Although a large community of image processing researchers tries to develop efficient procedures, fully automatic modeling still remains utopic, particularly due to high biological variations, as underlined in [4,5]. Therefore, for clinical use, it appears more realistic to develop a system enabling to interactively guide segmentation processes [4–8], the constraint being, for an acceptable efficiency, to maximize automation and minimize interactions.

As recently underlined in [5], to guide a segmentation process, a classical alternative consists in using a system allowing to interactively build and parameterize a pipeline of image processing operators, which will finally provide a model of a given object of interest from an initial input image. This means that the system must be able to manage the combination of user selected operators. Data-flow systems such as KHOROS [9], SCIRun [10] or ITK [11] are examples where this strategy is adopted.

Due to the specificity of the different medical structures, it is highly difficult to determine a single pipeline of operators allowing, from the initial image, the simultaneous segmentation of all structures of interest. For this reason, for a

* *Corresponding author*. Tel.: +33 388119079.
E-mail address: Jean-Baptiste.Fasquel@ircad.u-strasbg.fr (J.-B. Fasquel).
URL: http://www.ircad.org.

0169-2607/$ – see front matter © 2006 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.cmpb.2006.04.004

complete modeling, it appears reliable to consider a sequential or hierarchical approach where structures are segmented one after another, each one having its own adapted processing sequence. In such a case, several specific pipelines of operators will be built, leading to a complex dataflow. For instance, the user will start segmenting the lung from the initial image, then the heart (also from the initial image), and so on, until all structures are modeled.

However, to our knowledge, systems managing operator pipelining do not implicitely take already segmented structures into account when extracting a new one. For instance, when extracting liver tumors, it could be highly convenient to automatically constrain computations to the liver area if this one has already been segmented. In our sense, this aspect, related to the use of regions of interest, should be managed to optimize processings. As recently underlined [12], it can reduce computation time, which is crucial in a clinical context. Moreover, processing efficiency can be improved because irrelevant data is not taken into account (e.g. data outside the liver when segmenting liver tumors). For instance, in the case of an histogram-based local thresholding, the use of the overall image can lead to an incorrect determination of the optimal threshold because irrelevant modes can corrupt the histogram. For these reasons, ROIs are often considered in practice to provide a first rough representation of the boundaries of some image regions [13]: for initializing a heart segmentation procedure [14], or to provide an approximative segmentation of some structures to be further refined and regularized using deformable models [15].

In this paper, we propose a method to automatically determine ROIs, during an interactive hierarchical segmentation procedure, using the a priori knowledge of the topological structure of the medical image under analysis. When segmenting a given set of medical structures, this method allows to define the optimal ROI according to contextual information related to the nature of the already segmented objects.

This approach is similar to the one recently considered by Camara et al. [15]. In this case, the a priori information essentially concerns spatial relationships between anatomical structures. They consider a particular strategy (segmentation order) defined by the "segmentation difficulty": the skin is first segmented (the simplest structure to recognize [15]), then the skeleton, lungs, kidneys and finally the liver (the hardest structure to recognize [15]). During the procedure, relative positions are considered to define ROIs within which processing can be constrained. For instance, after having segmented both lungs, they use basic a priori knowledge on human anatomy such as for instance: "liver and kidneys are under lungs". Their approach provides a good approximation of structures, the shape of which are then refined and regularized using deformable models. The method we propose, focussing on approximative segmentation only (regularization being considered as a possible improvement), differs on two main aspects.

First, the proposed method manages any segmentation scenario or strategy. This is required because we consider an interactive segmentation software and the user-defined strategy is a priori unknown. As it will appear, our method is designed to support variations of the contextual information (i.e. nature of previously segmented structures).

The second difference of our approach with [15] concerns the nature of the a priori information and the incorporation of a mathematical formalism to manage it. We focus on topological information, incorporating additional structures such as liver tumors and hepatic vessels. Basic knowledge concerns for instance: "liver tumors belong to the liver", "hepatic vessels enter the liver but not lungs". The proposed formalism is based on an appropriate translation of such information into a graph describing corresponding topological relations. Such a graph-based structural representation has been considered in brain imaging to provide a synthetic description of segmented structures from which analysis is achieved [16–18]. In our case, the graph is used during the segmentation process and automatically adapted to the context and to the strategy, to facilitate the automatic and optimal determination of ROIs, which could be tedious manually. For instance, if the skin (or body), both lungs and a part of hepatic vessels are segmented, the ROI to be considered for extracting the liver will be the region associated with the body excluding areas related to lungs. Because hepatic vessels are known to enter the liver, corresponding image points will belong to this ROI, although they are already identified. After having extracted the liver, if the user decides to segment one or several liver tumors (referred to as "target"), our method will automatically propose to constrain computations within the liver area. Image points related to hepatic vessels will be excluded. Indeed, liver tumor and hepatic vessels are a priori mutually excluded, because a priori information does not mention knowledge such as "liver tumors intersect hepatic vessels". The proposed method manages single user-defined targets (i.e. liver tumor, liver, spleen) but also the preprocessing of an area to further segment a group of targets (referred to as "zone"). The choice of this area to be preprocessed is also optimized by trying to forsee the type of targets to be later considered by the user.

A similar approach has recently been studied by Bloch and Geraud [19] and illustrated for 3D brain imaging. They fusion heterogeneous fuzzy information (object geometry and radiometry, as well as structural information regarding their spatial relations) within a hierarchical procedure. The structural information also integrates topological information on which we focus (inclusion relationships between objects). In this paper, we particularly develop the formalism related to this information as well as to its integration within an interactive hierarchical segmentation procedure, which is not the case in [19]. Moreover, we incorporate the previously enounced notions of "zone" and "targets". Runtime aspects are also considered in our paper. Finally, although our approach could probably be extended to other anatomical regions and imaging modalities, one focuses on abdominal structures and CT images.

In the next section, we propose a mathematical formulation of the method. Then, we describe the software we developed to implement it. Before concluding on this work, we illustrate the potential and the interest of this method by detailing an experience report. This experiment concerns the segmentation of several medical structures of a given patient from a CT image.

218 COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE 82 (2006) 216–230

2. Method

First, this section aims at introducing a mathematical formalism for describing the a priori information, the hierarchical modeling procedure and the contextual information. Then, an explicit definition of optimal regions of interest will be proposed using this formalism. For justification and illustration purposes, we incorporate pratical situations which are expected to be encountered when hierarchically segmenting abdominal medical structures, because it constitutes our illustrative application.

2.1. *A priori information*

The a priori information used by the presented method concerns the topological relations between image regions, according to their type or nature (e.g. liver or lung). Let us consider an image, a priori composed of $N+1$ different kinds of regions corresponding to $N+1$ types associated with the set $S = \{0, \ldots, N\}$. A segmentation procedure applied to an initial image can lead to a segmented image I containing $N+1$ a priori known regions $X(u)$, $u \in S$ ($X(u=0)$ corresponding to the entire image): $I = \{X(u) | u \in S\}$.

The considered a priori information provides topological relations, defined by F, between types, as illustrated in Fig. 1. $\forall i \in S$, $F(i) \subseteq S$ represents direct fathers of i, with $F(0) = \emptyset$. According to Fig. 1, $F(5) = \{2, 4\}$ means that hepatic vessels belong to both the liver and the body, but do not intersect the spleen. The set $F^{\infty}(i)$ represent the ancestors of i (e.g. $F^{\infty}(4) = \{0, 2\}$ in Fig. 1). We propose to note $C = \{S, F, I\}$ the a priori information.

For each $u \in S$, we define $\bar{u}$ (associated with the set $\bar{S} = \{\bar{u} | u \in S\}$) so that $F(\bar{u}) = \{u\}$ and:

$$X(\bar{u}) = X(u) \setminus \left(\bigcup_{i \in S | u \in F(i)} X(i) \right) \quad (1)$$

$X(\bar{u})$ can be seen as the residue of $X(u)$. In the case presented in Fig. 1, $X(\bar{4})$ can correspond to the liver area (`Liver` = 4) from which both tumors (6) and vessels (5) are removed: $X(\bar{4}) = X(4) \setminus (X(6) \cup X(5))$.

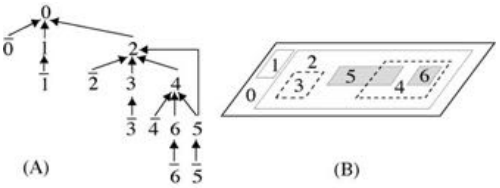

Fig. 1 – Example of a priori information. The set $S = \{0, 1, 2, 3, 4, 5, 6\}$ is associated with regions having specific overlapping properties (A) defined by F: $F(0) = \emptyset$, $F(1) = \{0\}$, $F(2) = \{0\}$, $F(3) = \{2\}$, $F(4) = \{2\}$, $F(5) = \{2, 4\}$, $F(6) = \{4\}$. S could be `Acquisition`, `Table`, `Body`, `Spleen`, `Liver`, `HepaticVessel`, `LiverTumor`, the patient being lying on a `Table` during the image `Acquisition`. B gives an example of topological structure of the image $I = \{X(u)\}$, $u \in S$. $X(\bar{u})$ corresponds to the residue with respect to the node $u \in S$.

2.2. *Hierarchical segmentation procedure*

In this work, we consider a hierarchical approach consisting in a sequence of processings P_t, $t \in \{0, \ldots, M\}$. At the end of the processing P_M, the segmentation is assumed to be terminated. Each P_t is assimilated to a pipeline of image processing operators. We propose the following formulation:

$$I_{out,t} = P_t(I_{in,t}, u, C_t) = \begin{cases} A_t(I_{in,t}, u, C_t) & \text{if } u \in S \\ B_t(I_{in,t}, u, C_t) & \text{if } u \in \bar{S} \end{cases} \quad (2)$$

$I_{in,t}$ is the input image to be processed, leading to an output image $I_{out,t}$. The parameter u defines the type of processing (A_t or B_t), and is interactively selected in the case of the implementation we describe in Section 3. In both cases, processings are restricted to an optimal ROI $R_t(u)$, computed according to the nature of the "already segmented structures" (referred to as the "contextual information", noted C_t and described in Section 2.3).

- If $u \in S$, P_t (of type A_t) focuses on the segmentation of a structure of type u, that we propose to call "target": liver, liver tumor, body, spleen are examples of targets. The segmentation, leading to a new identified region, expands the contextual information for the next step (i.e. C_{t+1}). $I_{out,t}$ is a binary image (or mask) corresponding to the segmented target.
- If $u \in \bar{S}$, P_t (of type B_t) aims at preprocessing a given region. We propose to associate $u \in \bar{S}$ with the notion of "zone". To each target $u \in S$ (e.g. liver, body) corresponds a zone $\bar{u}$ (e.g. liver zone, body zone). Such a processing can concern the optimal denoising of the liver zone so that some targets related to liver tumors can be more easily extracted at a next step. The contextual information is not expanded because such a processing does not identify any region of the image. $I_{out,t}$ is generally a grey-scale image from which targets are expected to be later segmented.

2.3. *Contextual information* $C_t = \{S_t, F_t, I_t\}$

2.3.1. *Definition*

The subset $S_t \subseteq S$ reports the region types which have been segmented during the $t-1$ first steps. At $t = 0$, when starting the first processing sequence, we have $S_0 = \{0\}$ because the entire original image is implicitely identified.

F_t can be associated with the initial graph of dependencies (related to F) within which only few nodes are valid (nodes $S_t \subset S$). Invalid nodes correspond to types of objects which have not been segmented yet. This is illustrated by Fig. 2. $\{S_t, F_t\}$ can be seen as a subgraph of $\{S, F\}$.

$\forall u \in S$, $F_t(u)$ reports valid fathers. For instance, in Fig. 2, valid fathers of 5 are defined by $F_t(5) = \{2\} \cup F(4) = \{2\}$, because $2 \in S_t$ and $4 \notin S_t$. $\forall \bar{u} \in \bar{S}$, $F_t(\bar{u})$, associated with a residue, only exists if the associated object type has been segmented at least once: $\forall \bar{u} \in \bar{S}$, $F_t(\bar{u}) = S_t \cap F(\bar{u})$.

Similarly to I, $I_t = \{X_t(u) | u \in S\}$, where $X_t(u)$ is the union of the already segmented structures of type u ($u \in S_t \iff X_t(u) \neq \emptyset$). According to Eq. (1), $X_t(\bar{u})$, $u \in S$ is naturally defined by:

$$X_t(\bar{u}) = X_t(u) \setminus \left(\bigcup_{i \in S_t | u \in F_t(i)} X_t(i) \right) \quad (3)$$

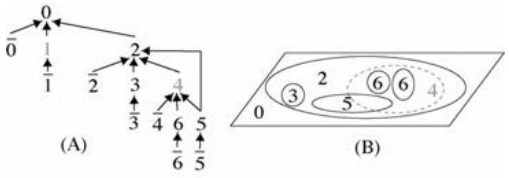

Fig. 2 – Example of contextual information at step t (A and B represent respectively $\{S_t, F_t\}$ and I_t). Nodes in grey are invalid and those in black are valid. In the case of the analogy considered in Fig. 1, the table (type = 1) and the liver (four-dashed frame) are missing, but two liver tumors (6), as well as the spleen (3) and hepatic vessels (5), have been detected (solid frames).

Therefore, if $u \notin S_t$, $X_t(\bar{u}) = \emptyset$ because $X_t(u) = \emptyset$. At $t = 0$, $\forall u \in S \setminus \{0\}$, $X_0(u) = X_0(\bar{u}) = \emptyset$. Besides: $X_0(0) = X_0(\bar{0}) = X(0)$.

2.3.2. *Updating the contextual information*

When a step t terminates, C_t is updated, leading to C_{t+1}, according to the type of processing (i.e. preprocessing or segmentation). If $u \in \bar{S}(P_t = B_t)$, $C_{t+1} = C_t$ as previously said.

If $u \in S(P_t = A_t)$, processings lead to an identified segmented structure of type $u \in S$ which can expand the contextual knowledge. C_{t+1} is defined by:

- $S_{t+1} = S_t \cup \{u\}$.
- The validation of the node u, leading to F_{t+1}.
- $X_{t+1} = X_t$, except $X_{t+1}(u) = X_t(u) \cup I_{\text{out},t}(u)$.

2.4. *Optimal ROI*

2.4.1. *Optimal ROI for a user-defined type $u \in S$: segmentation of a "target"*

We defined the optimal ROI using residues associated with direct valid fathers of u:

$$R_t(u) = \left(\bigcup_{l \in F_t(u)} X_t(\bar{l}) \right) \cup \left(\bigcup_{i \in S_t | u \in F^{\infty}(i)} X_t(i) \right) \quad (4)$$

For the first segmentation step ($t = 0$), because $\forall u \in (S \setminus S_0)$, $F_0(u) = \{0\}$, the optimal ROI is the whole image. If $u = 0$, $R_0(0) = \emptyset$.

The first term of Eq. (4) corresponds to the union of residues related to direct valid fathers of the target u. Therefore, processings will be constrained within the area which, according to the contextual information, is expected to encompass the object of interest. This area is reduced by removing segmented objects which overlap it. Such objects can be of the same type as the requested target. Indeed, according to the second term of Eq. (3), it appears that: if $u \in S_t$ (requested target previously encountered at least once), $R_t(u) \cap X_t(u) = \emptyset$. This is relevant because we assume that two objects of similar type do not intersect (e.g. two liver tumors).

The second term of Eq. (4) ensures that regions overlapping the object of interest belong to the proposed ROI. In our context, this avoids removing structures such as hepatic vessels (assumed to be already detected) when segmenting the liver. If this is not managed, it becomes impossible to find the part of the liver edge overlapped by hepatic vessels. In Fig. 1, this concerns the problem of finding the part of the boundary of 4 which is overlapped by 5 (assuming $X_t(5)$ removed from the ROI). The proposed second term also concerns structures totally included (i.e. not only overlapping) within the target (e.g. liver tumors belonging to the liver).

To illustrate Eq. (4), we propose to detail two situations, associated with two targets, in the context described in Fig. 2.

If the target is the liver (type 4):

$$R_t(4) = \left(\bigcup_{l \in F_t(4)} X_t(\bar{l}) \right) \cup \left(\bigcup_{i \in S_t | 4 \in F^{\infty}(i)} X_t(i) \right)$$

According to the contextual information, we have $\{l \in F_t(4)\} = \{2\}$ and $\{i \in S_t | 4 \in F^{\infty}(i)\} = \{5, 6\}$, leading to:

$$R_t(4) = X_t(\bar{2}) \cup \left(\bigcup_{5,6} X_t(i) \right) = X_t(\bar{2}) \cup (X_t(5) \cup X_t(6)).$$

The contextual information involves that $X_t(\bar{2}) = X_t(2) \setminus (X_t(3) \cup X_t(5) \cup X_t(6))$ (Eq. (3)). We obtain: $R_t(4) = X_t(2) \setminus X_t(3)$.

If the target is a liver tumor (type 6):

$$R_t(6) = \left(\bigcup_{l \in F_t(6)} X_t(\bar{l}) \right) \cup \left(\bigcup_{i \in S_t | 6 \in F^{\infty}(i)} X_t(i) \right)$$

According to the contextual information, we have $\{l \in F_t(6)\} = \{2\}$ and $\{i \in S_t | 6 \in F^{\infty}(i)\} = \emptyset$, leading to: $R_t(6) = X_t(\bar{2}) = X_t(2) \setminus (X_t(3) \cup X_t(5) \cup X_t(6))$.

In our sense, it clearly appears that R_t is the smallest region (therefore optimal) which, according to both a priori and contextual information, contains the object of interest.

2.4.2. *Optimal ROI for a user-defined type $\bar{u}$, $u \in S$: preprocessing of a "zone"*

The purpose of a preprocessing step t is to prepare an input image for further segmenting medical structures (targets). Future target types are a priori unknown when determining the ROI to be used for preprocessing. A reasonable assumption that we propose to consider is that the type $u_{\text{next}} \in S$ of a future target verifies: $F(\bar{u}) \in F(u_{\text{next}})$. In the situation described by Fig. 2, if $\bar{u} = \bar{2}$, the preprocessing concerns the body to further segment the liver, the spleen or hepatic vessels for instance.

According to this assumption, we propose to define a ROI managing all these possible future targets:

$$R_t(\bar{u}) = \bigcup_{i \in S | u \in F(i)} R_t(i) \quad (5)$$

At initialization, $\forall u \in S$, we have: $R_0(\bar{u}) = X_0(0)$.

It appears that, although intuitive, the ROI $R_t(\bar{u})$ does not necessarily correspond to the residue $X_t(\bar{u})$. Hereafter, we propose to comment this point and the proposed formalism for two practical situations.

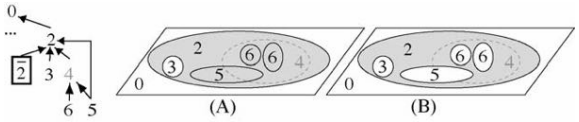

Fig. 3 – Preprocessing step associated with $\bar{2}$ (same context as in Fig. 2). The preprocessed area (in grey) includes hepatic vessels and liver tumors (types 5 and 6) (A) or excludes them (B).

First, we propose to detail $R_t(\bar{2})$ for the situation described in Fig. 2:

$$R_t(\bar{2}) = R_t(3) \cup R_t(4) \cup R_t(5)$$

Types 3 (spleen), 4 (liver) and 5 (hepatic vessels) are involved because, according to the previous assumption, they constitute possible targets after this preprocessing step. Even a priori already segmented structures are considered because nothing ensures that they have been entirely extracted. Indeed, our method supports the "progressive" segmentation of medical structures. In practice, this can be particularly useful for extracting a set of liver tumors in several steps (e.g. starting with the largest ones before extracting the smallest ones using several specific pipelines). For vessels, this can be useful when segments (i.e. parts of a vascular structure) share different properties in terms of intensity for instance (e.g. intensity variations resulting from an inhomogeneous repartition of the contrast agent in X-ray imaging): several segmentation steps can be considered to extract and model the entire hepatic vascular system, each one being optimized for a specific contrast.

According to Eq. (4), $R_t(3) = X_t(\bar{2})$, $R_t(4) = X_t(\bar{2}) \cup X_t(5) \cup X_t(6)$ and $R_t(5) = X_t(\bar{2})$. This leads to $R_t(\bar{2}) = X_t(\bar{2}) \cup X_t(5) \cup X_t(6) \neq X_t(\bar{2})$.

Fig. 3 illustrates the fact that $R_t(\bar{u})$ (Fig. 3A) should not be the residue $X_t(\bar{u})$. Otherwise, structures such as hepatic vessels would not be taken into account during the preprocessing (see Fig. 3B): further liver segmentation would be achieved on a partially preprocessed area excluding liver points overlapped by vessels (type 5 in Fig. 3). This is similar for liver tumors.

The second situation we propose to detail concerns the user requested preprocessing of the liver area (for futher segmenting hepatic vessels and liver tumors), assuming this one has not been segmented yet. Such a strategy could concern a dedicated study of the position of liver tumors with respect to hepatic vessels, the knowledge of liver boundaries being therefore useless. It seems reasonable to say that such a scenario does not seem likely in practice. However, for robustness purposes, the proposed method manages such a situation by proposing the most appropriate ROI according to available information, as illustrated by Fig. 4B: $R_t(\bar{4}) = R_t(5) \cup R_t(6) = X_t(2) \setminus (X_t(3) \cup X_t(5) \cup X_t(6))$. It appears that hepatic vessels and liver tumors are removed from the ROI (see Fig. 4): the ROI is optimized with respect to the user request which implicitely considers (i.e. initial assumption) that future targets will concern liver structures only (e.g. additional liver tumors or hepatic vessels). Therefore, the previously underlined problem (regarding liver boundary for instance) becomes meaningless.

To conclude, $\forall \bar{u} \in \bar{S}$, the proposed formalism leads to ROIs appearing optimal because their size is minimized with respect to possible future user-requested targets.

3. Implementation

The purpose of this section is to describe the software we developed to implement and illustrate the pertinence of our method. We describe the implementation we proposed for managing the information ($C = \{S, F, I\}$ and $C_t = \{S_t, F_t, I_t\}$) as well as of the sequence of processings P_t, $t \in \{0, \ldots, M\}$. Then, we give an overview of the software, from the user point of view. For illustrative reasons, most explanations are illustrated by images resulting from a modeling procedure performed with our software on a CT image. For simplification reasons, this procedure is not described because an experiment is detailed in the next section.

3.1. *Implementation of* $\{S, F\}$ *and* $\{S_t, F_t\}$ *using a medical dictionary*

The part $\{S, F\}$ of the a priori information is provided by a file (the medical dictionary), dynamically incorporated within our software when it starts:

```
...
[Liver]
    Nature  =Target
    Fathers    =Body
    Label  =X011XX
[HepaticVessel]
    Nature  =Target
    Fathers  =Liver,Body
    Label  =XXXX11
...
[LiverZone]
    Nature  =Zone
...
```

Structures `[Type]` explicitely define both S (field `Nature=Target`) and $\bar{S}$ (field `Nature = Zone`). Hereafter, types such as `LiverZone` are elements of $\bar{S}$, while those such as `Liver` belong to S. The list of available types ($S \cup \bar{S}$) is

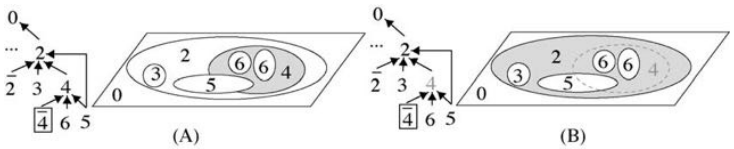

Fig. 4 – Preprocessing step associated with $\bar{4}$ for two different contexts A (liver-4 segmented) and B (liver-4 not segmented). In both cases, the context as well as the preprocessed area (in grey) are reported. It can be noted that the optimal ROI in situation B corresponds to the inappropriate one when the user requests $\bar{2}$ (Fig. 3B).

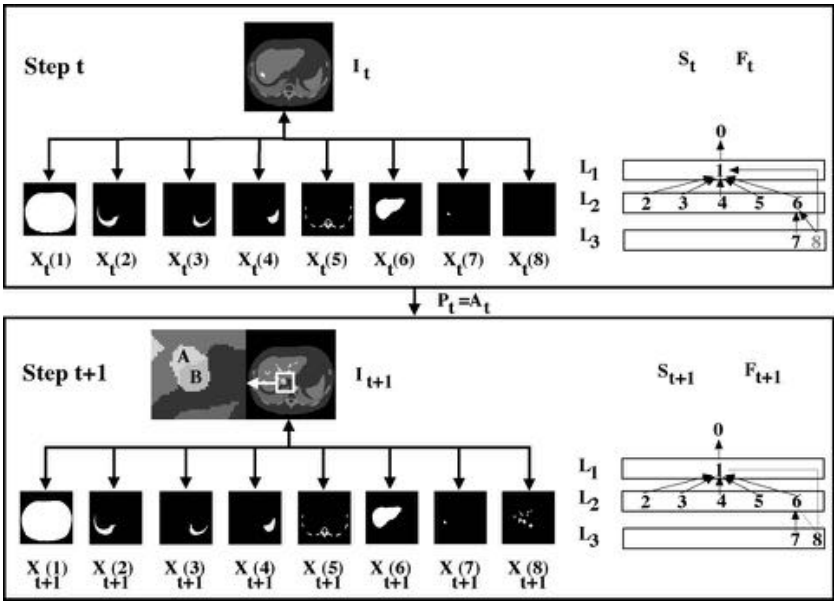

Fig. 5 – Evolution of the contextual information ($C_t = \{S_t, F_t, I_t\}$) resulting from the segmentation (processing step P_t of type A_t) of hepatic vessels (user requested type 8). The node 8 becomes valid (from grey to dark in the graph) at step $t+1$. The a priori information considered is $S = \{0, \ldots, 8\}$ corresponding to `Acquisition`, `Body`, `LeftLung`, `RightLung`, `Spleen`, `Bone`, `Liver`, `LiverTumor`, `HepaticVessel`. $\forall i \in S$, $X_{t+1}(i) = X_t(i)$, except $X_t(8)$ which is updated by the output binary image $I_{out,t}$ providing hepatic vessels ($X_{t+1}(8) = X_t(8) \cup I_{out,t}$). L_1, L_2 and L_3 represent different specific dynamic ranges used to identify $X_t(i)$, $i \in S$. The cropped area focuses on the segmented hepatic vessels at liver boundary: image points lying whithin the liver (A) have not the same intensity as those being outside (B). This results from the considered bit oriented encoding technique.

displayed in the graphical user interface of the software to be user-selected. Note that the information related to S would be sufficient to automatically compute the set $\bar{S}$ (e.g. creating a type `TypeZone` for each `Type`).

This file also provides the topological relationships F thanks to the field `Fathers` (only concerns elements of S): for instance, $F(\texttt{Liver}) = \{\texttt{Body}\}$ and $F(\texttt{HepaticVessel}) = \{\texttt{Liver}, \texttt{Body}\}$. The field `Label` is used to manage I_t, as detailed in the next section.

At each step t, from the knowledge of both $\{S, F\}$ and the processing sequence P_i ($i \in \{0, \ldots, t-1\}$ related to the t first steps), our software computes $\{S_t, F_t\}$ thanks to an appropriate graph structure (managing node validation/invalidation) that we do not detail for simplification purposes.

3.2. *Implementation of I_t*

In the case of the proposed implementation, I_t is a grey-level image (called reference medical image), allowing to store a set of binary images (corresponding to $\{X_t(i), i \in S\}$) in a single structure.

To identify $X_t(i)$ within I_t, the medical dictionary associates, to each type i, a specific value ($V(i)$) within a specific dynamic range (or layer $L(i)$), defined by the field `Label`. Table 1 gives examples of values and dynamic ranges associated with some medical structures. For instance, if the field `Label` equals X011XX (type `Liver`), then $L(\texttt{Liver}) = L_1 = 011100$ and $V(\texttt{Liver}) = 001100$. The `Label` associated with `Acquisition` is 0XXXXX. Fig. 5 illustrates the evolution of the contextual information C_t and superimposes notions of layer and topological graph.

At a given step t, it can be required to extract binary images $X_t(i)$, $i \in S$ from I_t to compute the optimal ROI:

$$\forall p \in D, \quad (X_t(i))(p) = \begin{cases} 1 & \text{if } I_t(p) L(i) = V(i) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where & denotes the bitwise logical AND operator, and D the set of image points. This corresponds to a pattern matching algorithm restricted to a specific dynamic range (set of bits).

At a given step t, when the segmentation of a structure of type $i \in S$ terminates, $X_t(i)$ is modified according to the output image $I_{out,t}$ (i.e. $X_{t+1}(i) = X_t(i) \cup I_{out,t}$). Practically, this is achieved by registering the binary image $I_{out,t}$ within I_t according to the following equation:

$$\forall p \in D, \quad I_{t+1}(p) = \begin{cases} I_t(p) | V(i) & \text{if } I_{out,t}(p) \neq 0 \\ I_t(p) & \text{otherwise} \end{cases} \tag{7}$$

where | denotes the bitwise logical OR operator, and D the set of image points. In Fig. 5, according to values reported in Table 1, image points in of the area A, have the value "`Body|Liver|Hepatic Vessel`" $\Longleftrightarrow$ 100000|001100| 000011 = 101111 = 125, assuming that the lowest significant bit (lsb) is the left one. Those of the cropped area B have the value 100000|000011 = 100011 = 61.

Table 1 – Examples of dynamic ranges (layer) and values (value) for six bits and some of the medical structures reported in Fig. 5

Type	Body	Spleen	Liver	LiverTumor	HepaticVessel
Label	1XXXXX	X111XX	X011XX	XXXX01	XXXX11
Value (V)	100000	011100	001100	000001	000011
Layer (L)	L_1	L_2	L_2	L_3	L_3

$L_1 = 100000$, $L_2 = 011100$ and $L_3 = 000011$.

This encoding technique allows to incorporate more binary images in I_t than a more classical one assigning one bit per type. For instance, two bits can manage three types (excluding the background, related to 00) with our technique, while only two types otherwise. The only constrain is that layers must be defined according to topological relationships: overlapping structures are associated with different layers. Indeed, this avoids merging and therefore losing the information related to $I_{out,t}$ when registering it in I_{t+1} in case of segmentation.

To illustrate this possible undesired effect, we propose to consider that Liver–Tumor belongs to the layer of `Liver`, with V(`Liver Tumor`) = 010000. We assume that the spleen, the liver and some liver tumors are already segmented. Thus, in I_t, liver tumors have the same intensity as the spleen (i.e. 011100 = 010000|001100 according to Eq. (7)). Thus, when extracting the X_t(`Spleen`) from I_t using Eq. (6), it appears that X_t(`LiverTumor`) is also incorporated. If the layer related to `LiverTumor` differs from the one of the `Liver`, this is obviously avoided.

We underline that this technique does not aim at being particularly original. Indeed, it only concerns a possible implementation that we describe for clarity purposes.

3.3. *Implementation of* P_t

Fig. 6 reports the principle of the proposed implementation of a processsing P_t defined by Eq. (2). The required input information is the input image ($I_{in,t}$), the user defined type and the contextual information $C_t = \{S_t, F_t, I_t\}$. The image I_t is also provided as input of the pipeline P_t. The core of the proposed implementation is managed by both implicit operators OPF (for "First") and OPL (for "Last"): they are automatically inserted by the software.

$\{S_t, F_t\}$ is used by the system to determine types of interest to be considered by the specific operator OPF when computing the ROI R_t according to either Eq. (4) (if $P_t = A_t$) or (5) (if $P_t = B_t$). Practically, OPF has two parameters dedicated to ROI computation. The first one (parameter "keep") concerns types $u \in S$ associated with regions belonging to the ROI. The second one (parameter "remove") specifies types related to regions to be removed from the ROI. For instance, let us assume a ROI defined by $R_t(6) = X_t(2) \setminus (X_t(3) \cup X_t(5) \cup X_t(6))$ (situation and notations considered in Section 2.4.1). The type set associated with the parameter "keep" (respectively "remove") is $\{2\}$ (respectively $\{3, 5, 6\}$). In fact, instead of using types for parameters, `Labels` are considered to extract the appropriate X_t(`Type`) from the input image I_t, according to Eq. (6). For instance, if `Type` = `HepaticVessel` is concerned, the field `Label` = XXXX11 is passed as parameter.

The implicit OPL operator is only required for segmentation purposes. When the user considers that the object of interest is correctly segmented, he notifies it to OPL (user action "3" in Fig. 6), thanks to a basic validation mechanism (i.e. dedicated button in the GUI of the software). The output of the last operator OPN is then transferred to the second output of OPL (leading to $I_{out,t}$), and registered in I_t (leading to I_{t+1}). This user action also allows to update $\{S_t, F_t\}$, leading to $\{S_{t+1}, F_{t+1}\}$.

For each user inserted operator (explicit operators, user action "2" in Fig. 6), processings are constrained to the ROI R_t according to the technique described in [12]. This method consists in extending the design pattern "iterator" [20] which is used by algorithms to access image points to be processed.

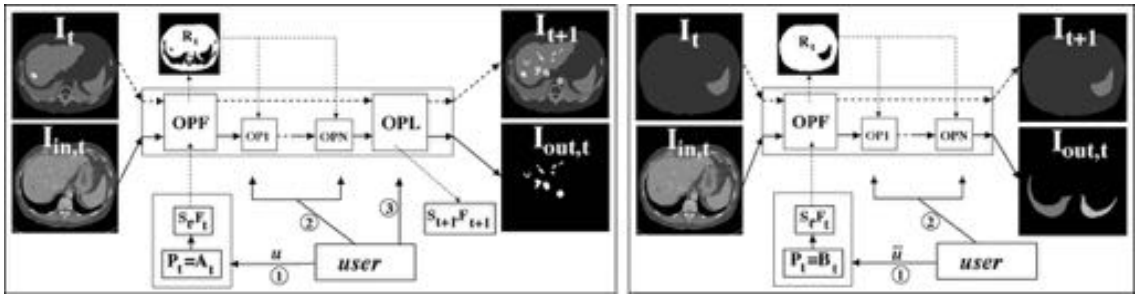

Fig. 6 – Proposed implementation of a pipeline P_t. Left: $P_t = A_t$ (user selects $u \in S$) concerns segmentation (of `HepaticVessel` in this example). Right: $P_t = B_t$ (user selects $\bar{u} \in \bar{S}$) concerns preprocessings (of `BodyZone` in this example). In this example, the preprocessed output image depicts two connected components from which both lungs are expected to be further segmented. 1, 2 and 3 represent the sequence of user actions: selection of the "target"/"zone", insertion of explicit image processing operators (i.e. OP1 to OPN), validation of the result (only if P_t is of type A_t). Operators OPF (for "First") and OPL (for "Last"—only if $P_t = A_t$) respectively manage ROI computation (R_t using $C_t = \{S_t, F_t, I_t\}$) and the updating of the contextual information. The optimal ROI is automatically applied to each downstream operator.

COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE 82 (2006) 216–230 223

The proposed extension concerns the control of image points to be scanned. Technically, iterators are parameterized by a `MaskScanner` in our case, similarly to both `BoxScanner` and `ShapedScanner` considered in [12]. The `MaskScanner` is the binary image (or mask) associated with the ROI and has been preferred because of less memory consuming, as explained in [12]. The ROI is provided to the image processing algorithms (OP1 to OPN), as represented in Fig. 6. The iterator ensures that only image points belonging to the ROI are then processed, what is expected to reduce runtime.

3.4. *Sequence of processings*

In this section, we describe how pipelines are connected together to manage hierarchical modeling procedures. Fig. 7 gives an example of a particular processing sequence (set $P_t, t \in \{0, \ldots, 9\}$). As detailed in the next section, at each step t, the user first explicitely specifies the input image $I_{in,t}$ (solid connections in Fig. 7). Then, he selects the type and manages processings as previously explained.

For $t \notin \{0, 4, 5\}$, $I_{in,t} = I_{out,0}$. $I_{out,0}$ is the grey-scale preprocessed (i.e. denoising) initial radiological image $I_{in,0}$. $I_{in,4} = I_{in,5} = I_{out,3}$ is also a grey-scale image containing two connected components from which both lungs are extracted at $t = 4$ and $t = 5$ (such a situation is further detailed and commented in Section 4).

In Fig. 7, ROIs R_t are also represented:

- At $t = 8$, the type is `LiverTumor` and the ROI is the previously segmented liver (at $t = 7$).
- At $t = 9$, the type is `HepaticVessel`. The ROI is restricted to the union of both body and liver residues which are the two valid direct fathers of this type. Body residue removes all organ areas (i.e. both lungs, the spleen, bones and liver) from the ROI. Liver residue (allowing to recover the liver area previously eliminated) rejects the liver tumor. In both cases, the Eq. (4) is used.
- At $t = 3$, the Eq. (5) is considered because a preprocessing is selected (i.e. the user chose `BodyZone`): the spleen is removed. Nevertheless, if `HepaticVessel` had been considered at $t = 3$, the corresponding mask would have been incorporated in R_3 (as explained in Section 2.4.2). Indeed, such a structure is known to partially belong to the liver which could be segmented after a preprocessing focussing on the body area (even if it is not the case in our example).

3.5. *The software*

The software we developed (see Fig. 8) provides a friendly graphical user interface for managing modeling procedures. User actions are restricted to explicit connections, operator insertion (and removal), type selection and segmentation validation.

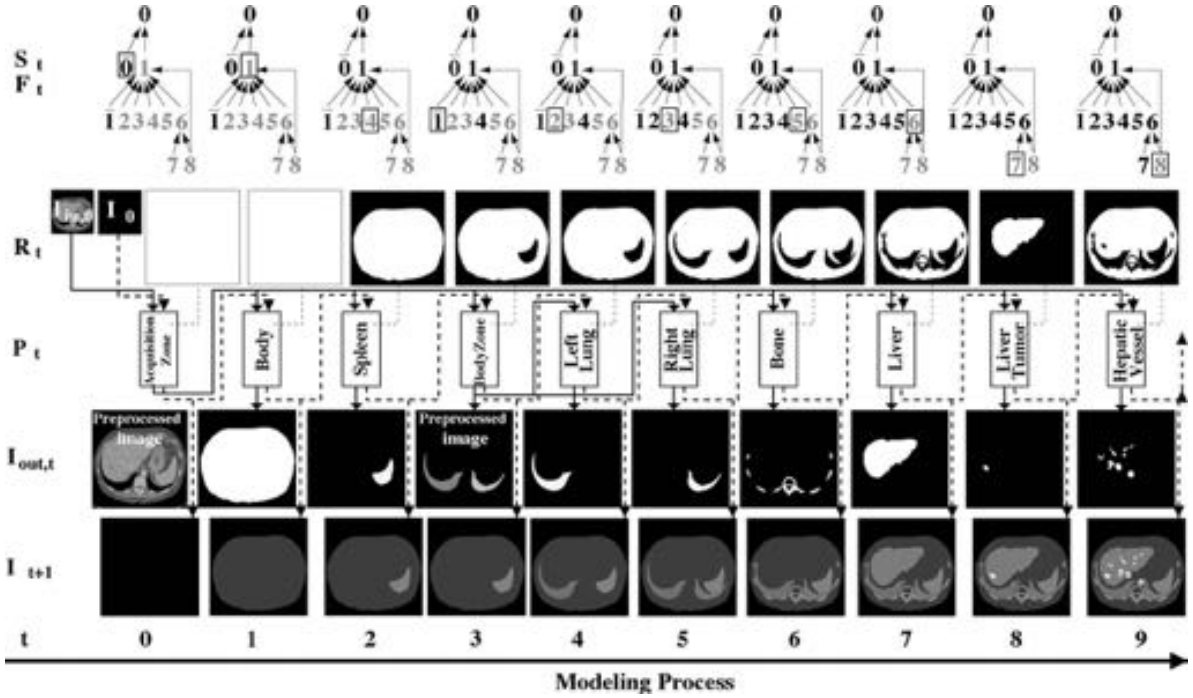

Fig. 7 – Example of processing sequence. Dashed connections are implicitely (the user does not specify them) achieved to transfer the updated contextual information (i.e. $C_t = \{S_t, F_t, I_t\}$) from a processing step to the next one, so that optimal ROIs R_t can be computed. Solid connections are explicitely managed by the user. The processing set P_t associated with steps $\{0, \ldots, 9\}$ is $\{B_0, A_1, A_2, B_3, A_4, \ldots, A_9\}$. The type set is $S = \{0, \ldots, 8\}$ and corresponds to medical structures `Acquisition`, `Body`, `LeftLung`, `RightLung`, `Spleen`, `Bone`, `Liver`, `LiverTumor`, `HepaticVessel`. For each P_t, the user-selected type is surrounded by a frame on the upper graph $\{S_t, F_t\}$ and also reported in the pipelines. $\{P_3, P_9\}$ are pipelines considered in Fig. 6.

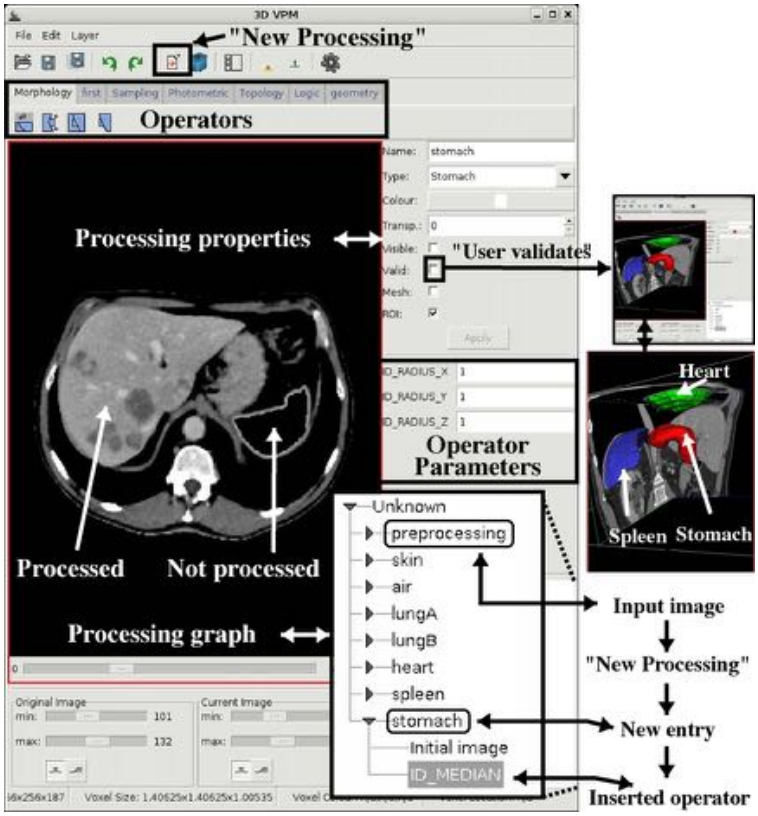

Fig. 8 – Graphical user interface of the developed software. The processing graph is detailed in Fig. 9.

Hereafter, we shortly describe the standard manipulation mode from the user point of view. Typically, the user starts a new segmentation procedure by loading the initial image (CT images are considered in this paper) and preprocessing it (e.g. denoising). For each step t, corresponding to a processing P_t, the user action sequence is:

(1) Selection of the input image $I_{\text{in},t}$. In most cases, the input image is the one issued from the preprocessed step (see Fig. 8: Input image). Then, the user presses the "new processing" button of the software toolbar (Fig. 8). This involves the insertion of a new entry (see Figs. 8 and 9A) and the creation of a new empty pipeline. An implicit connection (corresponding to dashed connections in Fig. 7) regarding the contextual information is automatically performed at this moment. The operator OPF is automatically inserted.

(2) Definition of some processing properties (Fig. 8): type (i.e. "target" or "zone") and name (e.g. "tumor 1" or "air"). Types are provided by the medical dictionary and are shown to the user thanks to a scrollable menu. If $u \in S$, the implicit operator OPL is automatically inserted. Note that the user can also disable the use of ROIs (enabled by default, see checkbox in Fig. 8). If enabled, the ROI R_t is automatically computed, according to the selected type.

(3) Insertion (or removal) and parameterization of image processing operators from those available (see Fig. 8 where the "Inserted operator" is a median filter), as well as effective processings (restricted to R_t). The user can visualize the output of each operator (or pipeline). In Fig. 8, the area related to the spleen (already segmented) has not been processed by the operator. Note that our software allows to superimpose intermediary data (i.e. output of the current operator or pipeline) on the initial image, making the control of the processing evolution easier. Examples of such superimpositions are given in Section 4. This step ends when the user considers that the result is correct.

(4) If $u \in S$ (i.e. processing focussing on segmentation), the user must validate the result (checking the "Valid" box in the "processing properties", Fig. 8), so that contextual information is updated. He can also ask to compute the 3D mesh and define its color and transparency. In Fig. 8(left) ("User validates"), the 3D model of the stomach is displayed with other models corresponding to some of the previously segmented structures. Meshes are superimposed on the 3D representation of the initial image.

Note that the ROI at a given step depends on the accuracy of previously segmented structures. Therefore, if one or several structures are not correctly delineated, the segmentation of future ones can be deteriorated. For instance, the segmentation of the liver can be unaccurate if the previous extraction of a neighbor structure, such as the stomach, has not been correctly achieved (e.g. stomach mask incorporating image points belonging to the liver). In such a case, the user must correct previous processings by inserting additional operators for in-

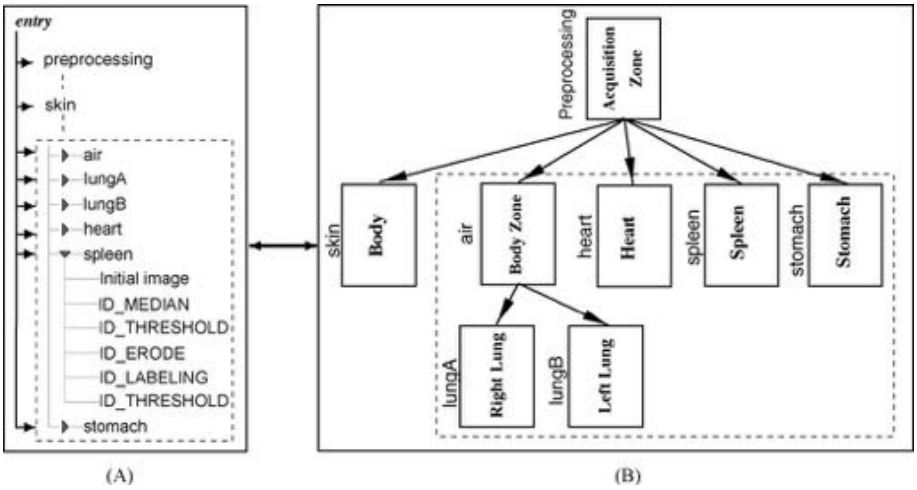

Fig. 9 – Image processing graph representation. A: graphical representation (see Fig. 8). B: associated pipelines (P_t) of operators. Connections correspond to solid arrows in Fig. 7. To each name (e.g. "lung A") is associated a type (e.g. `RightLung`).

stance. Our software provides a mechanism allowing to automatically update downstream data so that this modification is taken into account for subsequent steps: only required ROIs and processings are recomputed. This mechanism is not detailed for simplification reasons.

4. Experiment

The purpose of this paper is essentially to describe a general framework, including both a theoritical description of the method as well as a possible implementation. For this reason, this experiment only focuses on one clinical case, dedicated to the segmentation of abdominal structures from a CT image. The considered example has been chosen to illustrate the relevance of the proposed method for improving and accelerating segmentations. Only trivial image processing algorithms have been considered. Although coarse segmentations have resulted, illustrative images give a good overview of the behavior and the potential of the proposed method.

We first describe the experimental protocol, including image processing operators. Secondly, the considered modeling procedure is accurately detailed. Finally, we comment measured computing times. Note that, both modeling procedure and initial image differ from those considered in the previous section.

Table 2 – Detailed processings corresponding to the considered modeling procedure (Figs. 10 and 12), including runtimes (in s)

Type	OP1(.)	OP2(.)	OP3(.)	OP4(.)	OP5(.)	OP6(.)	OP7(.)
Body, 17.1	T([−260, +∞[), 1.8	S(1), 6	N, 1.8	S(1), 4.8	N, 2.7		
BodyZone, 14.3	T(] − ∞, −560]), 1.2	S(1,2), 1.9	N, 1.1	S(1), 6	N, 1.2	S(1,2), 2.9	
Left Lung, 2.9	S(1), 2.9						
Right Lung, 2.9	S(1), 2.9						
Heart, 25.8	T([71,131]), 1.1	E(3), 1.7	S(3), 2.9	D(3), 1.7	C(13), 18.4		
Spleen, 27.0	M(3), 18.0	T([72,130]), 0.5	E(5), 3.2	S(1), 2.7	D(5), 2.6		
Stomach, 62.1	M(3), 16.7	T([63,82]), 1.1	C(5), 12.2	O(11), 28.9	S(3), 3.2		
Aorta, 24.4	T([111,120]), 1.1	C(3), 9.4	O(5), 9.3	S(4), 2.5	D(7), 2.1		
BodyZone, 23	T([111,168]), 1.1	C(3), 9.4	O(3), 9.6	S(2,3), 4.0			
Left Kidney, 1.9	S(1), 1.9						
Right Kidney, 1.9	S(1), 1.9						
Liver, 155.3	M(5), 67	T([63,111]), 1.1	C(5), 4.6	F, 7.0	E(19), 68	S(1), 1.6	D(19), 6.0
LiverZone, 21.3	M(3), 3.0	T([34,63]), 0.8	C(5), 4.6	F, 8	O(3), 4.9		
Liver Tumor (1), 2.1	S(1), 2.1						
Liver Tumor (2), 2.1	S(1), 2.1						
Hepatic vessel, 34.8	M(3), 16.2	T([120,168]), 1.1	M(3), 14.8	S(1,2), 2.7			
Liver Tumor (3), 19.3	M(3), 2.9	T([73,168]), 0.4	E(3), 3.0	S(1), 1.3	D(3), 1.6	C(3), 4.9	O(3), 5.2

For each `Type`, the operator sequence is given. Operators (detailed in Section 4.2) are: threshold (T), closing (C), dilation (D), erosion (E), fillhole (F), median (M), not (N), opening (O) or extract connected components (S). The overall runtime is 436.3 s (i.e. 7 min and 16 s). Runtimes are reported for each Type and operator (e.g. Body segmentation spent 17.1 s and the first operator 1.8 s).

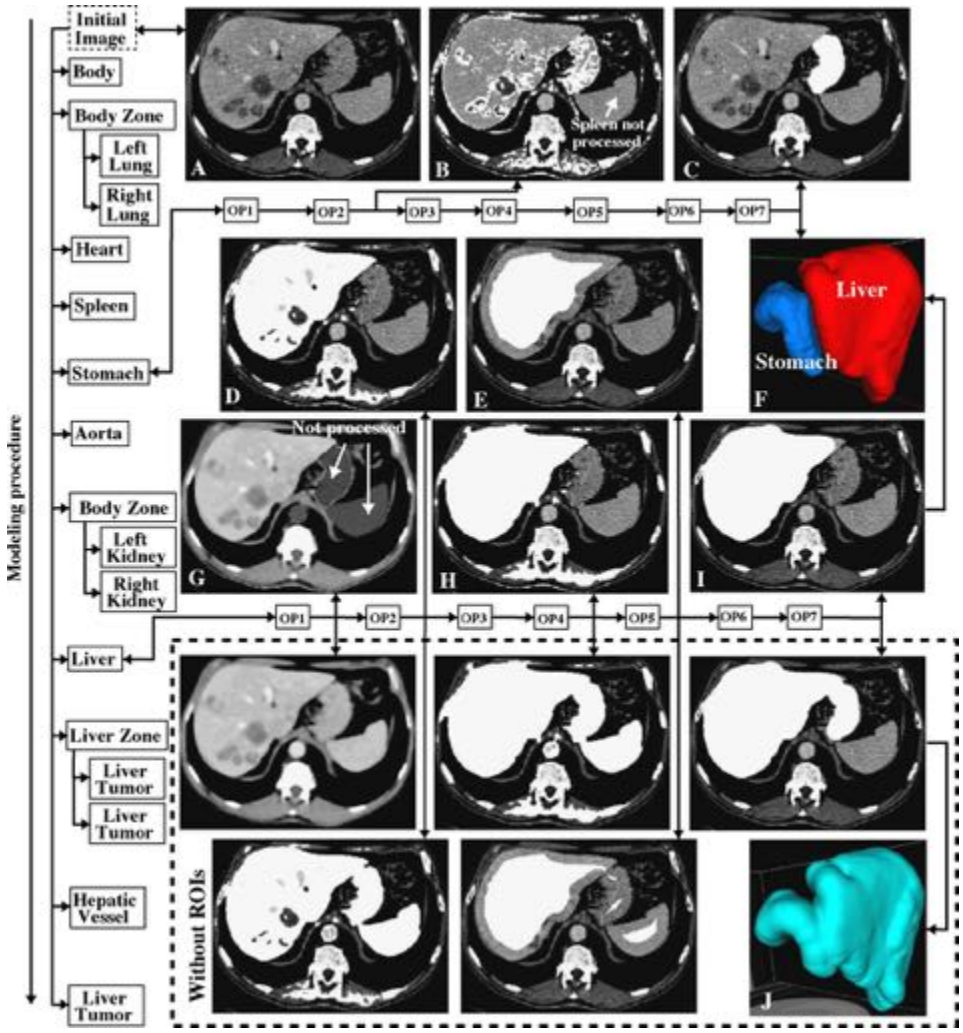

Fig. 10 – Considered modeling procedure (`types` are reported from top to bottom—A is the initial image). Arrows indicate the input image at each processing step. {OP1, . . . , OPN} are the operator sequences detailed in Table 2. Some intermediary images related to `Stomach` and `Liver` types are shown (e.g. B corresponds to the output of OP2). "`Liver`" images obtained without considering ROIs are also reported. Intermediary images are superimposed on the initial one.

4.1. *Experimental protocol*

For this experiment, we considered a 3D CT image of size $256 \times 256 \times 187$ voxels. The voxel size is $1.4 \times 1.4 \times 1$ (mm). The computer used for runtime measurements has an AMD64 3500+ CPU (2.2 GHz), 1 GB of RAM and a linux operating system. ITK version 2.2 has been considered. A limited number of basic operators is available at this time in our software, leading to coarse segmentations. Nevertheless, this appeared sufficient to illustrate the relevance and the potential of this work, this being the purpose of this paper. Hereafter, we shortly present the set of operators involved in our experiment, as well as associated notations (used in Section 4.2).

- Median (noted **M**(p)): the median operator is based on the `Median-Image-Filter` ITK filter [11], restricted to ROIs [12]. It is used for noise removal. The parameter p denotes the size $p \times p \times p$, in voxels, of the kernel.
- Threshold (noted **T**([**a**, **b**])): the image thresholding operator uses the `Binary-Threshold-ImageFilter` ITK filter [11], restricted to ROIs [12]. [a,b] is the interval of intensities of interest (corresponding image points are set to "white", the other ones becoming "black").
- Not (noted **N**): the binary image inversion operator is implemented with the `NotImageFilter` ITK filter [11] and is constrained to ROIs [12].
- Select and extract connected components (noted **S**(**n**)) from a ROI of a binary image. The parameter n is the

user defined list of considered connected components, assuming they are sorted in decreasing order with respect to the size (e.g. 1 is the largest one). It uses several ITK filters, such as `ConnectedComponentImageFilter` and `RelabelComponentImageFilter`.

- Binary morphological operators: dilation, erosion, closing and opening (respectively noted **D**(**p**), **E**(**p**), **C**(**p**) and **O**(**p**)). They are implemented with the `FastIncremental-BinaryDilate-ImageFilter` ITK dilation filter [11], which is based on the work of [21]. Instead of processing all image points, this algorithm directly focuses on object boundaries, therefore reducing computations. Erosion is performed by dilating the inverted binary image, and is combined with dilation for closing and opening. The parameter p denotes the size $p \times p \times p$, in voxels, of the cubic structuring element.
- The "FillHole" operator (noted **F**) allows to fill cavities. In our case, it only concerns binary images and is not yet optimized for ROI. For the quantitative evaluation presented in Section 4.3, this does not dramatically affect the overall runtime which remains representative of the potential of the proposed method.

4.2. *Modeling procedure*

Table 2 describes the different steps of the modeling procedure (runtimes being commented in the next section). Figs. 10 and 12 show some intermediary images resulting from operators detailed in Table 2. No particular overall preprocessing has been performed. All operator parameters have been interactively inserted and tuned to perform processings. Although detailed in Table 2, steps related to the aorta, kidneys, lungs, spleen and body are not described for simplification reasons.

As observed in Fig. 10B, the already detected spleen was not concerned by the thresholding focusing on stomach, although sharing similar intensities. Fig. 10G shows that, for liver segmentation, an important median filtering has been first performed (parameter in Table 2) to smooth internal liver structures (i.e. liver tissues, hepatic vessels and tumors). Then, an image thresholding has been achieved (see Fig. 10D). After a closing, a "fillhole" operator has been applied to the thresholded image in order to incorporate internal liver structures

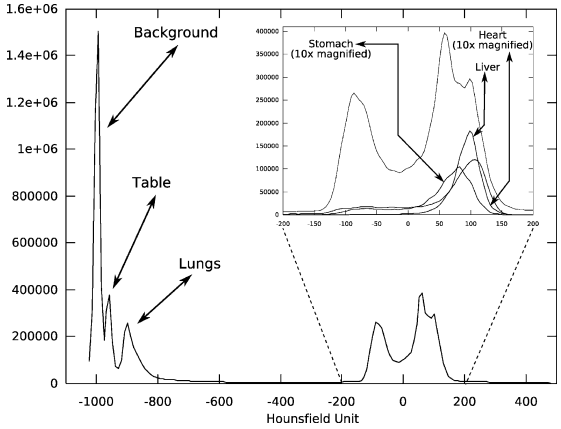

Fig. 11 – Intensity histogram of the image used for experiments. The three left peaks correspond to low intensities and can be respectively associated with the background (i.e. air outside the patient), to the table on which the patient was lying during acquisition and to the lungs (being assimilated to internal air). Partially overlapping peaks corresponding to the liver, stomach and heart have been evaluated after rough segmentation (i.e. histograms restricted to structure areas).

such as tumors and vessels (see Fig. 10H). The image has been then eroded to remove noise and connections (see Fig.'10E). Finally, the liver, corresponding to the largest connected component, has been dilated to approximatively recover its original size.

Intermediary images obtained with this liver operator sequence but without ROIs are also reported in Fig. 10. The observed liver-stomach connection, naturally avoided with our approach, illustrates the potential of our method to improve the robustness of the segmentation (3D meshes of structures are given by Fig. 10F and J). This does not prove that the liver could not have been extracted without ROIs. It only shows that, for the considered operator sequence, segmentation can be facilitated by removing previously extracted structures. Our approach appears particularly interesting for such neighbour objects sharing similar intensities, which is the case for the

Table 3 – Runtime comparison: "ROIs" vs. "No ROIs"

Type	OP1(.)	OP2(.)	OP3(.)	OP4(.)	OP5(.)	OP6(.)	OP7(.)
Liver, 155.3 (ROI), 216.1 (no ROI)	M(5), 67, 101	T([63,111]), 1.1, 0.5	C(5), 4.6, 17.66	F, 7.0, 6.3	E(19), 68, 83	S(1), 1.6, 1.6	D(19), 6.0, 6.0
LiverZone, 21.3 (ROI), 66.3 (no ROI)	M(3), 3.0, 23.7	T([34,63]), 0.8, 0.4	C(5), 4.6, 19.6	F, 8, 6.4	O(3), 4.9, 16.2		
Hepatic Vessel, 34.8 (ROI), 48.8 (no ROI)	M(3), 16.2, 24.5	T([120,168]), 1.1, 0.5	M(3), 14.8, 21.9	S(1,2), 2.7, 1.9			
Liver tumor (3), 19.3 (ROI), 75.9 (no ROI)	M(3), 2.9, 24.5	T([73,168]), 0.4, 0.4	E(3), 3.0, 14.3	S(1), 1.3, 3.1	D(3), 1.6, 2.8	C(3), 4.9, 15.6	O(3), 5.2, 15.2

Reported operator sequences are those of Table 2, and are therefore designed for ROI-based processings. When ROIs are not used, processing results are not necessarily acceptable (e.g. liver–stomach connection described in Fig. 10). Therefore, this comparison is a rough estimation. As in Table 2, runtimes are reported.

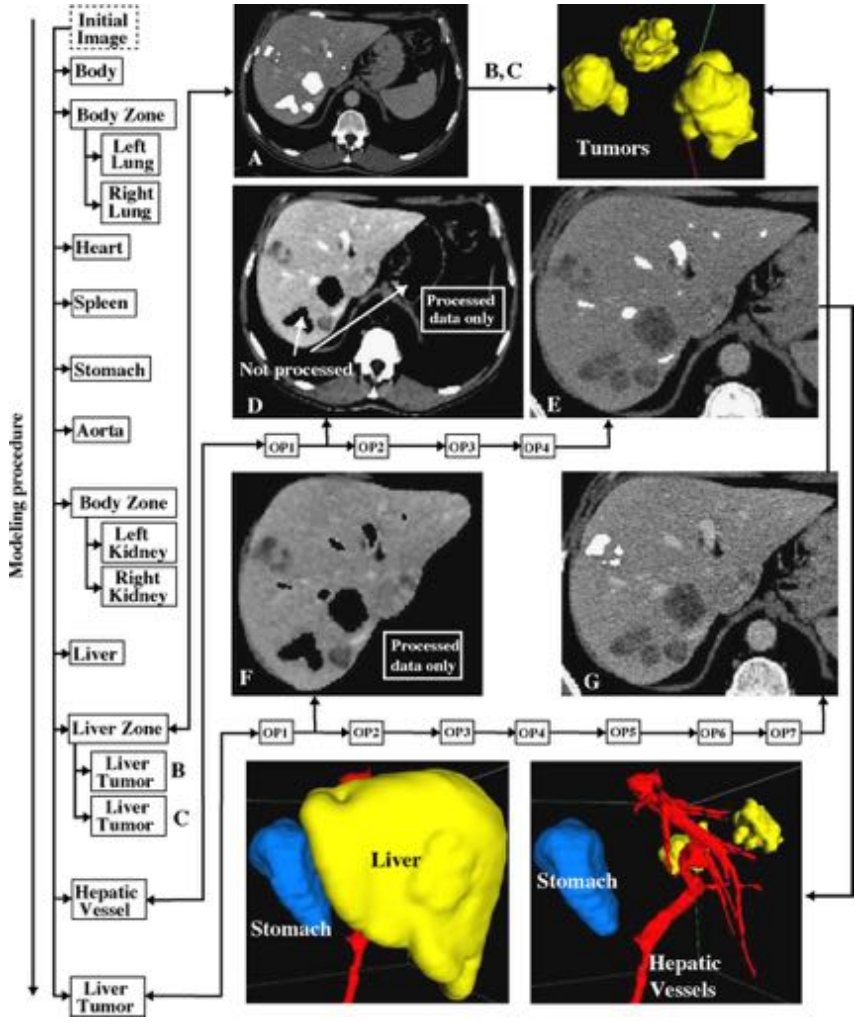

Fig. 12 – Similar to Fig. 10, except that it essentially concerns three `LiverTumors` and `Hepatic Vessel`. Images, chosen for illustrative reasons, show visualization modes provided by our software: A, E and G result from the superimposition of the initial image and the locally processed one, while D and F concern processed data only (data outside the optimal ROI are "black").

liver, the heart and the stomach (see the histogram in Fig. 11). In our opinion, this illustrates how our approach can facilitate segmentation, by avoiding undesired connections for instance.

Fig. 12 illustrates visualization modes provided by our software. For hepatic vessels segmentation, the median filtering (Fig. 12D) did not concern areas related to previously segmented liver tumors and the stomach for instance. 3D meshes confirm (in Fig. 12) that hepatic vessels only partially intersect the liver (this being specified by our a priori topological information). The last liver tumor could have been extracted consecutively to the two first ones (Fig. 12B and C). For illustrative reasons, we preferred to manage it separately to show an example of a locally processed area in such a context (see Fig. 12F): previously segmented hepatic vessels and liver tumors appear in "black". In practice, such a situation can occur to complete the segmentation of liver tumors, assuming that some of them have been previously missed.

This modeling procedure has been interactively tuned for this clinical case, and is not necessarily appropriate for other ones. However, the segmentation of the body (by searching for the "background") and both lungs could probably be automated because, for this imaging modality, associated intensities really differ from those of other structures (see histogram of Fig. 11). Some regularization methods could probably be advantageously used to refine coarse segmentations. This can be essential because the relevance of a ROI at a given step depends on the accuracy of previously extracted and identified structures. Regularization could be achieved using deformable models, as considered in [15] (even interactively as in [6,22]).

COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE 82 (2006) 216–230 229

For instance, the liver mask reported in Fig. 10 could be used to provide a relevant initial shape to such an algorithm. It is likely that such a model would converge to the real contour, because liver mask boundaries obtained with an ROI are not far from real liver edges. Without ROIs, liver mask boundaries are close to those of the stomach (due to the liver-to-stomach connection): the deformable model would probably be trapped.

For the ROI determination, user actions are clearly minimized (only "targets"/"zones" and validations are required), making the software really user-friendly. Operator sequence construction and tuning (i.e. operator parameters) could be clearly facilitated by automating some sequences (e.g. body and lung segmentation), therefore further decreasing interactions. However, for many structures, the choice of an operator sequence is not trivial, as underlined by the important research activity in this field. Therefore, for a clinical use, it appears reasonable that the user interactively adapts sequences to patient particularities.

4.3. *Runtime*

Table 2 depicts runtimes measured for this experiment. The overall runtime is about 7 min and 16 s for the 17 steps, excluding the time spent by the user to choose and insert operators, as well as to tune their parameters. Note that such a runtime evaluation highly depends on the choice of the strategy, operator sequences and algorithm implementation. For this reason, these results must only be considered as an illustration.

The strategy must be chosen so that processing sequences associated with time consuming operators are considered for last steps, where ROI size is minimal. For this reason, `Body`, `Right Lung` and `Left Lung` have been considered first because, due to the little segmentation complexity, only basic and fast processings are required. For instance, time consuming median filterings (such as the $M(3)$ used for `Spleen` for instance) can be avoided because intensitites are distinct enough to manage segmentation using image thresholding, inversion and connected component selection only.

The overall runtime could be further reduced by avoiding some processing redundancies. For instance, the operator $M(3)$ used for `Spleen` also concerns the ROI used for `Stomach`, `LiverZone`, `Hepatic Vessel` and the last `Liver Tumor`. For this reason, this operation could be performed only once, as a preprocessing for all these steps (saved runtime of about 38.8 s). Such redundancies (i.e. repeating median filtering) have been preferred for illustrative reasons, to evaluate the progressive runtime reduction regarding such a time consuming operator, when the procedure evolves (see OP1 for `Stomach`, `LiverZone`, `Hepatic Vessel` and the last `Liver Tumor`). Similarly to the ROI size, the runtime decreases from 18 s (`Stomach`-OP1) down to 2.9 s (`Liver Tumor`-OP1). In our sense, the runtime gain would be clearly more important for more sophisticated and therefore more time consuming operators, such as those based on anisotropic diffusion for instance. Indeed, in [12], runtime decreases from 52 s (anisotropic filtering applied to the bounding box of the liver) down to 16 s (anisotropic filtering constrained to the liver area). Note that the bounding box of the liver is even smaller than the overall image as we consider here. This gives an idea of the practical impact on runtime of the use of ROIs computed from segmented medical structures.

As observed for morphological erosion and dilation, runtime depends on the context. For instance, erosion for `Liver`-OP5 ($E(19) = 68$ s) is more than 10 times slower as the dilation for `Liver`-OP7 ($D(19) = 6$ s). This is because, for the erosion (dilating the inverted image), the algorithm faces many boundaries and connected components (some of them corresponding to false alarms related to bones for instance, see Fig. 10H). This involves more computations than for the dilation focussing on a single component having a shorter frontier (see Fig. 10E).

From Table 3, it appears that the proposed approach highly accelerates computations when ROIs are small (e.g. 3.11 faster for `LiverZone` and 3.93 for `LiverTumor`). The improvement is also non negligible for larger ROIs: about 1.4 faster for `Hepatic Vessel` and `Liver`. Although reported processing sequences without ROIs lead to worse coarse segmentations (e.g. `Liver`–`Stomach` connection), this gives an idea of the possible runtime gain resulting from our approach.

From Table 3, it appears that some processings can be faster without ROIs than with ROIs (e.g. `Liver`-OP2). In our sense, this results from the cost of the image scanning based on the extended iterator. Indeed, in order to determine the next image point to access and process, the iterator evaluates whether it belongs to the ROI. Such a test becomes non negligible if the processing concerns an operation having a similar complexity, as for a thresholding. This also depends on the ROI size: OP4 for `Liver Tumor` becomes faster with ROI. In our sense, this aspect is not really critical in practice because it concerns processings which are not time consuming. The most important is to reduce runtime related to time consuming operators (e.g. median and morphological operators).

5. Conclusion

We proposed a method to automatically determine the optimal ROI according to a priori topological information related to medical structures which are not necessarily mutually excluded (e.g. vessels which enter some organs and exclude other ones). This method automatically takes the segmentation strategy into account by adapting the a priori information according to the nature of already segmented structures.

We proposed a mathematical formalization of the method, as well as a possible implementation. This work provides a framework which seems, according to the experiment, to make segmentation easier and faster, by simply providing a medical dictionary describing overlapping relationships between structures of interest. Although illustrated for abdominal structures only, the method could be probably be advantageously used for any anatomical regions, the only requirement being the definition of an appropriate topological model of medical structures.

An interesting aspect concerns the nature of the considered information. In our sense, such structural information is more stable and reliable than information based on intensity or shape for instance. Indeed, it is reasonable to say that liver shape is less constant from one patient to another, than considered topological information (e.g. lungs and liver a priori do

not intersect, whatever the patient). Similarly, intensity distributions highly depend on imaging conditions, which is also a priori less reliable.

Next steps of this work will concern the integration of a regularization procedure, in order to refine segmentations, this being crucial because ROI depends on modeling accuracy. Then, we will try to improve our model by incorporating spatial information to further reduce ROIs. This information could concern relative positions between structures in order to refine ROI determination. We hope that this will lead to a useful software tool adapted to clinical use, both in terms of runtime and robustness requirements.

Acknowledgements

First of all, we would like to thank our research and development team, and in particular Guillaume Brocker and Nicolas Papier. We are very grateful to the French National League against Cancer, the Alsace Region, the French Research Ministry and the French Agency for Innovation for their financial support. We would also like to thank all our medical partners, who have provided us medical images of patients, and in particular Pr. Afshin Gangi and Pr. Catherine Roy from the radiological department of the Civil Hospital in Strasbourg. We also thank the surgical team of Professor Marescaux for their help in our works, in particular Pr. Mutter, Pr. Leroy, Dr. Vix.

REFERENCES

[1] J. Duncan, N. Ayache, Medical image analysis: progress over two decades and the challenges ahead, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 85–106.
[2] R. Taylor, D. Stoianovici, Medical robotics in computer-integrated surgery, IEEE Trans. Robot. Automat. 19 (5) (2003) 765–781.
[3] H.-P. Meinzer, M. Thorn, C.E. Cardenas, Computerized planning of liver surgery—an overview, Comput. Graphics 26 (2002) 569–576.
[4] Y. Kang, K. Engelke, W.A. Kalender, Interactive 3D editing tools for image segmentation, Med. Image Anal. 8 (2004) 35–46.
[5] S.D. Olabarriaga, A. Smeulders, Interaction in the segmentation of medical images: a survey, Med. Image Anal. 5 (2001) 127–142.
[6] P. Bruin, V. Dercksen, F. Post, A. Vossepoel, G. Streekstra, F. Vos, Interactive 3D segmentation using connected orthogonal contours, Comput. Biol. Med. 35 (2005) 329–346.
[7] I. Wolf, M. Vetter, I. Wegner, T. Bottger, M. Nolden, M. Schobinger, M. Hastenteufel, T. Kunert, H.-P. Meinzer, The medical imaging interaction toolkit, Med. Image Anal. 9 (2005) 594–604.
[8] J. Kaminsky, P. Klinge, T. Rodt, M. Bokemeyer, W. Luedemann, M. Samii, Specially adapted interactive tools for an improved 3D-segmentation of the spine, Comput. Med. Imaging Graphics 28 (2004) 119–127.
[9] K. Kostantinides, J. Rasure, The Khoros software development for image and signal processing, IEEE Trans. Image Process. 3 (3) (1994) 243–252.
[10] Scientific Computing Imaging Institute, SCIRun, 2005, http://software.sci.utah.edu/scirun.html.
[11] National Library of Medicine, Insight Segmentation and Registration Toolkit, 2006, http://www.itk.org.
[12] J.-B. Fasquel, V. Agnus, J. Lamy, An efficient and generic extension to ITK to process arbitary shaped regions of intrest, Comput. Methods Programs Biomed. 81 (1) (2006) 1–7.
[13] S. Hirano, S. Tsumoto, Rough representation of a region of interest in medical images, Int. J. Approximate Reason. 40 (2005) 23–34.
[14] C.A. Cocosco, T. Netsch, J. Sénégas, D. Bystrov, W.J. Niessen, M.A. Viergever, Automatic cardiac region-of-interest computation in cine 3D structural MRI, Int. Cong. Ser. 1268 (2004) 1126–1131.
[15] O. Camara, O. Colliot, I. Bloch, Computational modeling of thoracic and abdominal anatomy using spatial relationships for image segmentation, Real-Time Imaging 10 (2004) 263–273.
[16] J.-F. Mangin, D. Rivière, O. Coulon, C. Poupon, A. Cachia, Y. Cointepas, J.-B. Poline, D. Le Bihan, J. Régis, D. Papadopoulos-Orfanos, Coordinate-based versus structural approaches to brain image analysis, Artif. Intell. Med. 30 (2004) 177–197.
[17] D. Rivière, J.-F. Mangin, D. Papadopoulos-Orfanos, J.-F. Martinez, V. Frouin, J. Régis, Automatic recognition of cortical sulci of the human brain using a congregation of neural networks, Med. Image Anal. 6 (2002) 77–92.
[18] G. Wagenknecht, H.J. Kaiser, U. Buell, O. Sabri, MRI-based individual 3D region-of-interest atlases of the human brain: a new method for analyzing functional data, Methods Informat. Med. 43 (2004) 383–390.
[19] I. Bloch, T. Geraud, H. Maitre, Representation and fusion of heterogeneous fuzzy information in the 3D space for model-based structural recognition—application to 3D brain imaging, Artif. Intell. 148 (2003) 141–175.
[20] E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns, Addison-Wesley, 1995.
[21] L. Vincent, Morphological transformations of binary images with arbitrary structuring elements, Signal Process. 22 (1) (1991) 3–23.
[22] T. Kunert, T. Heimann, A. Schröter, M. Schöbinger, T. Böttger, M. Thorn, I. Wolf, U. Engelmann, H.-P. Meinzer, An interactive system for volume segmentation in computer-assisted surgery, in: R.L. Galloway Jr.(Ed.), Medical Imaging 2004: Visualization, Image-Guided Procedured, and Display, vol. 5367, Proceeding of SPIE, 2004, pp. 799–809.

Characterization of home-acquired blood pressure time series using multiscale entropy for patients treated against kidney cancer

Antoine JAMIN[1], Jean-Baptiste FASQUEL[1], Anne HUMEAU-HEURTIER[1], Pierre ABRAHAM[2], Georges LEFTHERIOTIS[3], and Samir HENNI[1,2]

[1] LARIS, Université d'Angers, 62 Avenue Notre-Dame du Lac, 49000 Angers, France
[2] University Hospital Center of Angers, Vascular Departement and MITOVASC, Angers, France
[3] University Hospital Center of Nice, Vascular Departement, Nice, France

Abstract. This study deals with the telemonitoring, with a connected tensiometer, of 16 patients treated for a kidney cancer. Each one of these patients recorded his/her blood pressure at home during 63 days and the data was sent to his/her medical doctor. At the same time they were treated with antihypertensive medication when necessary. In this work, our goal was to analyze the complexity of the blood pressure time series. For this purpose, we proposed to use the refined composite multiscale entropy (RCMSE) measures. Our results show that the patterns of RCMSE through temporal scales evolve with the antihypertensive medication. The later might therefore have an impact on home-acquired blood pressure complexity. RCMSE could therefore be an interesting information theory-based tool to study home-acquired physiological data.

Key words: telemonitoring, connected tensiometer, blood pressure, time series, multiscale entropy, clustering, irregularity, complexity

1 Introduction

Kidney cancer is the 12th kind of cancer in terms of frequency in the world. It represents 338,000 new cases diagnosed in 2012 [1]. It can be treated by a VEGF[1] chemotherapy that consists in eliminating the capillaries of the tumor. However such a treatment can lead to blood pressure increases. This is why antihypertensive medication are often given to the patients. Our work deals with patients treated against a kidney cancer using a VEGF chemotherapy. During the treatment, patients recorded their blood pressure once a day at home using a connected tensiometer (see Figure 1). Such an IoT-based sensor [2, 3] facilitates the following of blood pressure increases. Our goal herein is to study, with the refined composite multiscale entropy (RCMSE), the complexity of the blood pressure time series.

[1] Vascular Endothelial Growth Factor

RCMSE is an improved version of the multiscale entropy (MSE). MSE relies on the sample entropy algorithm and on a coarse-graining procedure to study irregularity of time series at different time scales [4].

The paper is organized as follows: the MSE and RCMSE algorithms and the measurement procedure are introduced in Section 2. Our results are then detailed and discussed in Section 3. The paper ends with a Conclusion.

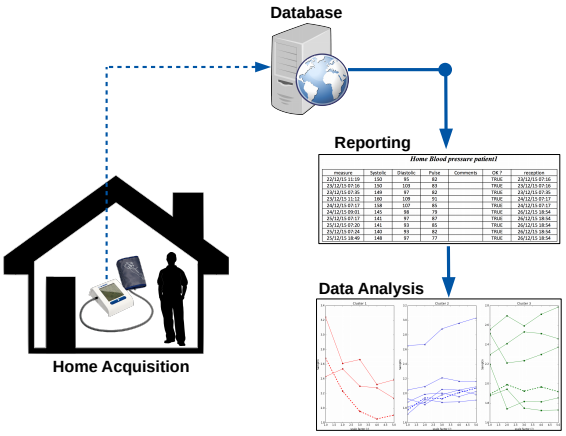

Fig. 1: Project overview

2 Materials and Methods

2.1 Sample Entropy

Pincus proposed to quantify the irregularity of time series with approximate entropy (ApEn) [5]. To overcome the limitation of ApEn, sample entropy has later been introduced [6]. Sample entropy is a conditional probability measure that quantifies the likehood that a sequence of m consecutive data points – that matches another sequence of the same length – will still match the other sequence when their length is increased by one sample (m+1) [4]. Sample entropy is computed as:

$$SampEn(m, r, N) = -\ln \frac{A^m(r)}{B^m(r)}, \tag{1}$$

where r is the tolerance, m is the sample length, $A^m(r)$ and $B^m(r)$ are, respectively, the probability that two sequences will match for $m+1$ and m points.

2.2 Multiscale entropy

MSE allows to quantify the complexity of time series by measuring its irregularity at different time scales [4, 6].

The MSE algorithm is composed of three steps [7, 8]

1. a coarse-graining procedure is used to derive a set of time series representing the system dynamics on different time scales. For a monovariate discrete signal x of length N, the coarse-grained time series y^τ is computed as:

$$y_j^{(\tau)} = \frac{1}{\tau} \sum_{i=(j-1)\tau+1}^{j\tau} x_i, \tag{2}$$

   where τ is the scale factor and $1 \leqslant j \leqslant \frac{N}{\tau}$. The length of the coarse-grained time series is N/τ.

2. computation of the sample entropy for each coarse-grained times series.

3. plot of the sample entropy for each time scale τ.

2.3 Refined composite multiscale entropy

MSE generates some undefined values for short time series [9]. When large scale factors τ are used, the coarse-grained time series may have a small number of samples. This may lead to undefined sample entropy values. This is why RCMSE has been introduced [9].

In RCMSE, the coarse-grained signal $y_k^{(\tau)}$ is computed for different values of a parameter k:

$$y_{k,j}^{(\tau)} = \frac{1}{\tau} \sum_{i=(j-1)\tau+k}^{j\tau+k-1} x_i, \tag{3}$$

where τ is the scale factor, x is the original signal, $1 \leqslant j \leqslant \frac{N}{\tau}$ and $1 \leqslant k \leqslant \tau$.

The RCMSE at scale τ is calculated using the following formulation:

$$RCMSE(x, \tau, m, r) = -\ln \frac{\sum_{k=1}^{\tau} n_{k,\tau}^{m+1}}{\sum_{k=1}^{\tau} n_{k,\tau}^{m}}, \tag{4}$$

where r is the tolerance, m is the sample length, $n_{k,\tau}^{m+1}$ and $n_{k,\tau}^{m}$ are the number of matched vector pairs (computed on $y_k^{(\tau)}$) for $m+1$ and m, respectively.

2.4 Measurement procedure

The study was conducted on 16 patients (5 women, 11 men; 171 cm ± 9 cm; 87 kg ± 29 kg; 62 years ± 8 years). The 16 patients were daily monitored, during 63 days, using a connected tensiometer (Tel-O-Graph®, I.E.M. GmbH). Each recorded time series (Systolic Blood Pressure (SBP) and Diastolic Blood Pressure (DBP) time series) had therefore 63 samples. In our work, $MBP(k)$ processed by RCMSE was defined as $MBP(k) = (SBP(k) + 2 \times DBP(k))/3$, $1 \leqslant k \leqslant 63$, where $SBP(k)$ is the systolic blood pressure at day k, $DBP(k)$ is the diastolic blood pressure at day k and $MBP(k)$ is the mean blood pressure at day k.

2.5 Parameters used

To compute RCMSE, 3 parameters have to be set : m, r, and τ. It has been recommended that, to compute the sample entropy, the time series length has to be between 10^m and 20^m [6]. Our data have 63 samples. We therefore have chosen $m = 1$ and time scale $\tau \leqslant 5$. Moreover, we have chosen $r = 0.15 \times \sigma$ (where σ is the standard deviation of time series at scale factor $\tau = 1$) [8].

3 Results and Discussion

Figure 2 presents three examples of MBP time series. Figure 3 shows the corresponding RCMSE curves. By analyzing all the time series we observe 3 kinds of RCMSE patterns : global decreasing sample entropy with scales (cluster 1), global increasing sample entropy with scales (cluster 2), and non-monotonic sample entropy values with scales (cluster 3); see Figure 4.

If sample entropy increases with scales, this means that the signal contains complex structures across multiple scales. If the sample entropy decreases with scales, this means that the signal has information only on the shortest scale (similarly to white noise [8]). In the undetermined case, no definite conclusion can be drawn.

We can observe that the signals of cluster 1 are similar to RCMSE of a white noise: the signal irregularity decreases with scale. Cluster 2 corresponds to an increase of the signal irregularity with scales. Cluster 3 corresponds to an intermediate situation.

These clusters have been compared with the number of hypertensive medication taken by the patients. As reported in Table 1, for cluster 1, patients received only 1 antihypertensive treatment. For cluster 2, patients received 2 antihypertensive treatments. For cluster 3, patients received 3 antihypertensive treatments.

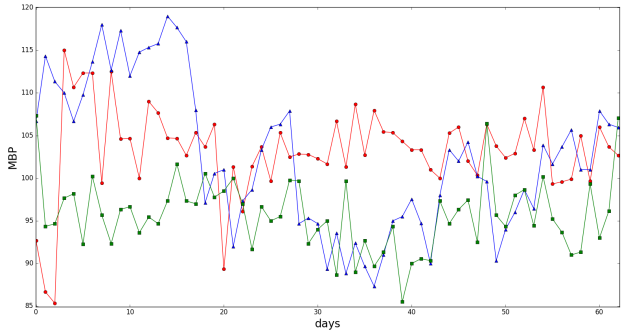

Fig. 2: Representative signals for each cluster
(cluster 1 : red — cluster 2 : blue — cluster 3 : green)

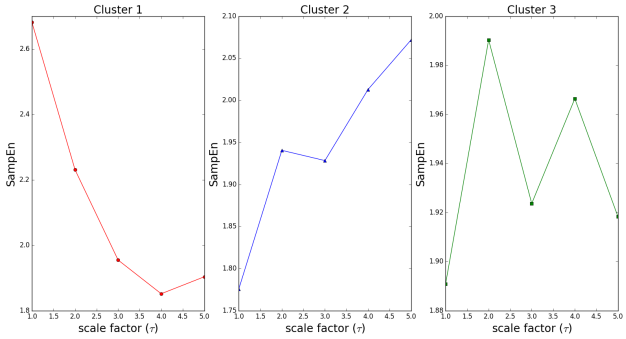

Fig. 3: RCMSE and clusters of representative signals shown in Figure 2.

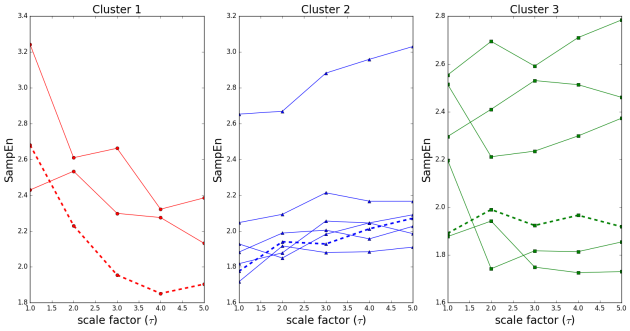

Fig. 4: RCMSE and clusters.
Original signals leading to dotted RCMSE are represented in Figure 2

Cluster	Cluster 1	Cluster 2	Cluster 3
RCMSE shape	decrease	increase	undetermined
#patients	3	7	6
#medication	1	2	3

Table 1: Clusters, RCMSE and #medication

Anti-hypertensive medication may therefore play a role in the complexity of mean blood pressure time series.

4 Conclusion

In this work we studied the complexity of blood pressure time series with RCMSE. Further work is needed to apply the RCMSE algorithm on much more data. However, our work shows that antihypertensive medication might have an influence on mean blood pressure complexity. Our study shows that the complexity of data extracted by connected devices may be interesting for physiological purposes. Now, RCMSE could be applied to other IoT-based physiological time series. The next step of our work will consist in identifying patient profiles in order to personalize the telemonitoring.

References

1. World Cancer Research Fund International, http://www.wcrf.org [may 2017]
2. Y. Yin, Y. Zeng, X. Chen, Y. Fan: The internet of things in healthcare: An overview. Journal of Industrial Information Integration, Volume 1, 3–13 (2016).
3. S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, K. S. Kwak: The Internet of Things for Health Care: A Comprehensive Survey. IEEE Access, Volume 3, 678–708 (2015).
4. A. Humeau-Heurtier: The Multiscale Entropy Algorithm and Its Variants: A Review. Entropy, Volume 17, 3110–3123 (2015)
5. S. M. Pincus: Approximate entropy as a measure of system complexity. Proceedings of the National Academy of Sciences, Volume 88, 2297–2301 (1991).
6. J. S. Richman, J. R. Moorman: Physiological time-series analysis using approximate entropy and sample entropy. American Journal of Physiology - Heart and Circulatory Physiology, Volume 278, H2039–H2049 (2000).
7. M. Costa, A. L. Goldberger, C.-K. Peng: Multiscale Entropy Analysis of Complex Physiologic Time Series. Physical Review Letters, Volume 89, 068102 (2002).
8. M. Costa, A. L. Goldberger, C.-K. Peng: Multiscale entropy analysis of biological signals. Physical Review E, Volume 71, 021906 (2005).
9. S.-D. Wu, C.-W. Wu, S.-G. Lin, K.-Y. Lee, C.-K. Peng: Analysis of complex time series using refined composite multiscale entropy. Physics Letters A, Volume 378, 1369–1374 (2014).

Computers in Biology and Medicine 66 (2015) 269–277

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/cbm

A semi-automated method for measuring the evolution of both lumen area and blood flow in carotid from Phase Contrast MRI

Jean-Baptiste Fasquel [a,*], Aldéric Lécluse [b], Christine Cavaro-Ménard [a], Serge Willoteaux [b]

[a] LARIS Laboratory, EA4094, University of Angers, 62 avenue Notre Dame du Lac, 49000 Angers, France
[b] Radiology Department, University Hospital, 4 rue Larrey, 49933 Angers, France

ARTICLE INFO

Article history:
Received 28 May 2015
Accepted 19 September 2015

Keywords:
Semi-automated analysis
Region-based active contour
Blood flow
Magnetic Resonance Imaging
Phantom

ABSTRACT

Phase-Contrast (PC) velocimetry Magnetic Resonance Imaging (MRI) is a useful modality to explore cardiovascular pathologies, but requires the automatic segmentation of vessels and the measurement of both lumen area and blood flow evolutions. In this paper, we propose a semi-automated method for extracting lumen boundaries of the carotid artery and compute both lumen area and blood flow evolutions over the cardiac cycle. This method uses narrow band region-based active contours in order to correctly capture the lumen boundary without being corrupted by surrounding structures. This approach is compared to traditional edge-based active contours, considered in related works, which significantly underestimate lumen area and blood flow. Experiments are performed using both a sequence of a homemade phantom and sequences of 20 real carotids, including a comparison with manual segmentation performed by a radiologist expert. Results obtained on *the* phantom sequence show that the edge-based approach leads to an underestimate of carotid lumen area and related flows of respectively 18.68% and 4.95%. This appears significantly larger than weak errors obtained using the region-based approach (respectively 2.73% and 1.23%). Benefits appear even better on the real sequences. The edge-based approach leads to underestimates of 40.88% for areas and 13.39% for blood flows, compared to limited errors of 7.41% and 4.6% with our method. Experiments also illustrate the high variability and therefore the lack of reliability of manual segmentation.

1. Introduction

Cardiovascular diseases remain an important cause of death. Several imaging modalities are available to explore cardiovascular pathologies: angiography, Doppler echo, Computed Tomography and Magnetic Resonance Imaging (MRI). Clinical cardiovascular MRI has evolved from a modality that provided anatomic imaging to one that gives access, in a day-to-day practice, to functional information as well [1]. A key component of this enhanced utility has been the capability, with Phase-Contrast velocimetry MRI to analyze and quantify hemodynamic parameters. Phase-Contrast velocimetry MRI provides not only anatomical information on vessels but also functional information related to blood flow over the cardiac cycle. Fig. 1A reports an example of anatomical information (anatomical image), together with functional information (phase image). The evolution of blood flow through carotid is measured over a cardiac cycle using the intensity provided by phase image.

A major strength of MRI, compared to other modalities such as Doppler echo, regards its ability to give access to all anatomical regions in all orientations and its sensitivity to a broad range of flow velocities. This technique has a variety of clinical applications [1], including evaluation of valvular heart diseases (regurgitation and stenosis), assessment of left and right ventricular stroke volume, assessment of vascular stenosis [2], evaluation of arterial wall shear stress [3].

For a clinical use, analysis of the Phase-Contrast velocimetry MRI data requires the delineation of vessel lumen over the sequence, from which measurements, such as flow calculations, can be performed. The purpose of this work is to automate the delineation of carotid over entire PC-MRI sequences, in order to measure the evolution of both the lumen area and the blood flow over a cardiac cycle, and could be extended to additional measurements such distensibility for instance. Our proposal concerns a semi-automated analysis method. Manual interaction only regards the initialization of the automated analysis, compared to a fully manual analysis involving the manual delineation of vessel lumen over each image of an entire sequence, this being a tedious and time consuming task, non-compliant with clinical routine.

* Corresponding author.
E-mail address: Jean-Baptiste.Fasquel@univ-angers.fr (J.-B. Fasquel).

http://dx.doi.org/10.1016/j.compbiomed.2015.09.017
0010-4825/

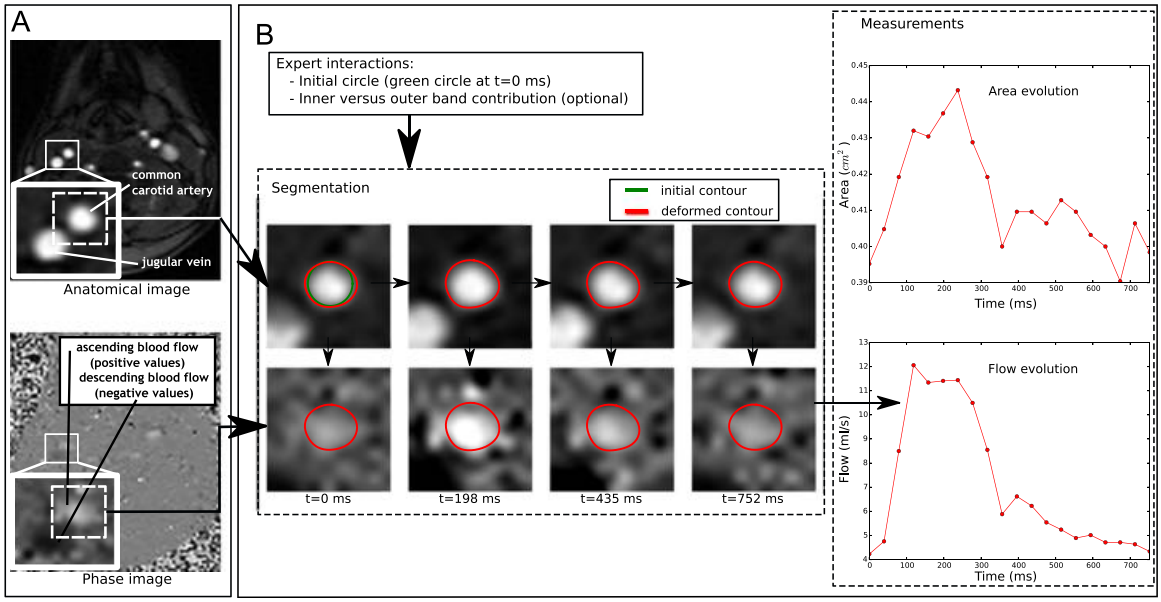

Fig. 1. Images (A) and principle of the proposed semi-automated analysis method (B). (A) Anatomical image (top) and phase image (bottom) from 2D cine PC-MRI for the examination of carotid. (B) Principle of the proposed approach including segmentation and measurements (curves on the right side). Area evolution. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Carotid lumen segmentation has been widely studied but not in the context of PC-MRI. Related studies have concerned 3D CTA (Computer Tomography Angiography) and MRA (Magnetic Resonance Angiography) [4–6]. Some of related methods are based on region-growing, active contours and vessels' centerline prior detection.

Recent studies involving PC-MRI sequences focus on large vessels differing from carotid, being a smaller structure. Considered larger structures belong to the thoracic area (e.g. aorta [7–10] and pulmonary artery [10]). The study related to [9] focuses on the aorta, using manually initialized edge-based deformable models applied to the anatomical image, without any blood flow measurement. 3D PC-MRI sequences are considered in [8], where vessels are semi-automatically segmented using phase images only, by coupling anisotropic filtering (for noise reduction) and edge-based deformable models (level-sets). In [10], vessel lumen is segmented using 3D Angiographic MRI images, and PC-MRI sequences are only used for computing velocity profiles. In [7], segmentation exploits phase information only, using edge-based active surfaces, where edge regards the frontier between the blood flow region and stationary tissues.

These previous works focus on 3D segmentation (except [9]), with underlying difficulty of extracting entire 3D structures depicting bifurcations for instance. Our context differs for these works because one focuses lumen area and blood flow evolution from a single 2D vessel slice from PC-MRI sequences. Besides, compared to previous works dealing with PC-MRI, one focuses on the carotid, being a smaller structure (e.g. diameter of about 25 mm for the aorta, against 7 mm for the carotid [11]). To our knowledge, except preliminary results limited to a single sequence [12], such an application of PC-MRI has not been considered yet.

In this work, we propose to segment the carotid lumen using narrow band region-based active contours: deformation is guided by statistics locally computed within bands surrounding the contour. As studied in this paper, this choice is motivated by the lack of contrast of carotid vessels, making edge-based approach unappropriate (compared to previously mentioned studies on PC-MRI sequences focusing on larger structures [7–10]). Note that such local region-based approaches have been widely studied for general purpose segmentation [13–15] as well as in previously mentioned works on 3D CTA and MRA images [4–6,16].

The main contribution of this paper concerns the proposed semi-automated analysis method for measuring both lumen area and blood flow, using, for the segmentation step, narrow band region-based explicit active contours inspired from [16]. Another part of the contribution regards the comparison performed with traditional edge-based active contours, as considered in the similar work (i.e. PC-MRI sequence analysis) focusing on the aorta [9]. This shows the need of a region-based approach for this anatomical structure. The last part of the contribution regards experiments involving both sequences of healthy volunteers and a sequence acquired using a carotid phantom. The use of a phantom is known to be relevant to evaluate segmentation algorithms for medical applications, compared to validation protocols limited to the use of reference segmentations resulting from manual drawings achieved by radiologist experts, with underlying uncertainty [17].

Section 2 describes the proposed semi-automated method, while Section 3 presents experiments and results. Before concluding, Section 4 discusses this work and related results.

2. Method

2.1. Overview

Fig. 1 A shows an acquisition example at a specific instant of an entire sequence (both anatomical and phase images are reported). Concerning the acquisition protocol, the axial slice is assumed to be manually selected (before acquiring an entire sequence) to be perpendicular to the common carotid artery (flow is maximal for a perpendicular plane): the axial plane is selected by the radiologist on a coronal plane of the carotid.

Fig. 1 B provides an overview of the proposed semi-automated analysis method. The input is the sequence of both anatomical and phase images. Segmentation is performed using anatomical

images. Phase images are only used for measurements occurring after the appropriate segmentation of the lumen of the carotid.

Segmentation is initialized by an expert (i.e. radiologist) who places an initial contour (circle) within the carotid lumen of the first anatomical image of the sequence (green circle in Fig. 1B). As detailed in Section 2.2, active contours are then considered to deform the initial circle towards lumen boundaries, for anatomical image of the sequence (red circles in Fig. 1B). The initialization involves a second (optional) parameter: "inner versus outer band contribution" mentioned in Fig. 1B. This parameter enables us to control the width or extent of the segmented contour (this parameter is further explained in Fig. 2 and Section 2.2, then studied in Section 3, and finally discussed in Section 4).

Measurements, detailed in Section 2.3, are finally automatically performed using obtained carotid contours as well as the velocity information encoded in phase images (see Fig. 1B).

2.2. Narrow band region-based active contours

A contour C is defined by

$$C : [0,1] \rightarrow \mathbb{R}^2$$
$$u \rightarrow c(u) = (x(u), y(u))^t \quad (1)$$

where u represents the curve parameter.

The considered method for contour deformation is based on previous proposals [16,18]: the manually initialized contour (circle) is iteratively deformed (see examples in Fig. 2 – top) in order to minimize the energy E defined by

$$E = \alpha \int_0^1 |C'(u)|^2 \, du + \lambda(\omega \iint_{B_{in}(C)} |I_a(x,y) - \mu_{in}|^2 \, dx \, dy + (1-\omega) \iint_{B_{out}(C)} |I_a(x,y) - \mu_{out}|^2 \, dx \, dy) \quad (2)$$

The first term α regards the internal energy, and aims at ensuring the smoothness of the contour. The last two terms, weighted by λ, concern the so-called external energy, and are attached to the image (anatomical image I_a in our case) so that the contour is attracted by some particular image features. Compared to [16], both α and λ are not linked (i.e. in the form $\alpha = a$ and $\lambda = (1-a)$ in [16]), to have a more general formulation and be able to give significantly more weight to the external energy. This is detailed in Section 3.1 when replacing, for comparison purposes, the considered external energy by a standard edge-based external energy and setting parameters to $\alpha = 0.5$ and $\lambda = 15$. Note that, for the considered external energy, α and $\lambda$are set to 0.5, therefore corresponding to the formulation of [16].

Compared to traditional edge-based external energy attracting the contour towards high image gradients [19], we consider a region-based external energy involving intensity descriptors computed within B_{in} and B_{out} regions respectively lying inside and outside the contour being deformed [18,16]. B_{in} and respectively B_{out} regions are bounded by C as well as by its internal and external parallel curves ($c_{[-B]}$ and $c_{[B]}$) built using the normal vector $n(u)$ [16]:

$$c_{[B]} = c(u) + Bn(u) \quad (3)$$

Fig. 2 A reports deformed contours as well as related inner and outer band boundaries.

The deformation procedure aims at, in particular, minimizing intensity deviations within narrow bands, with respect to mean

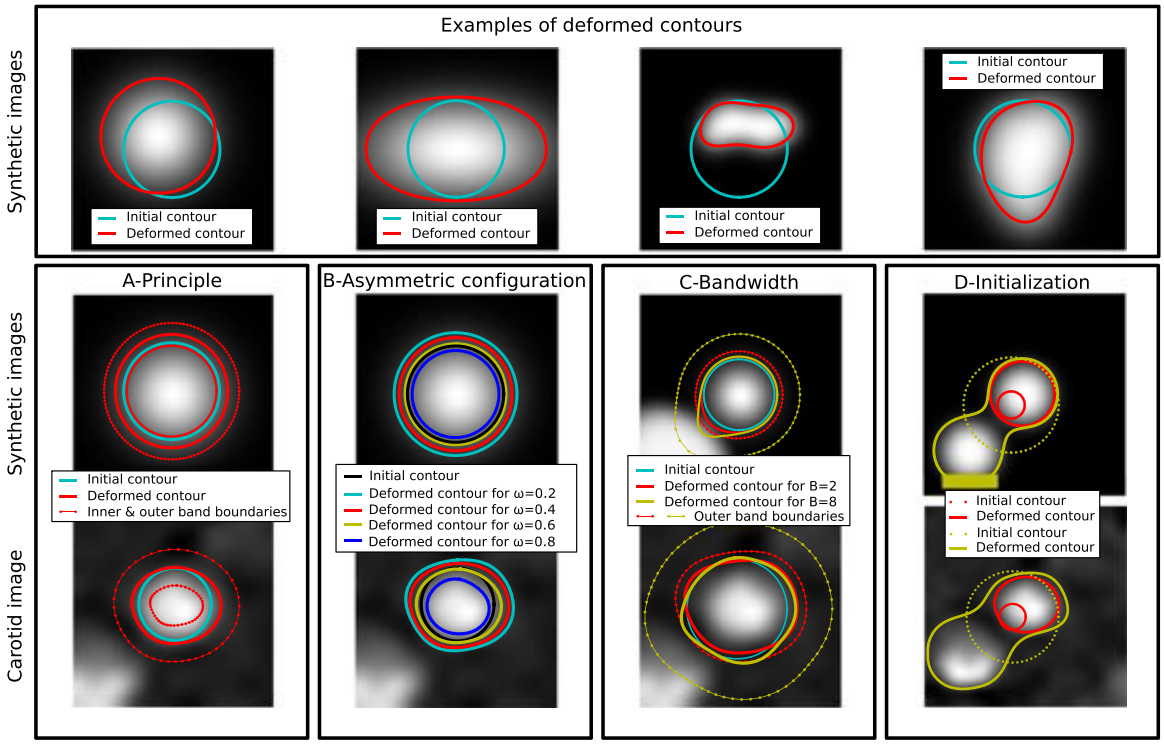

Fig. 2. Narrow band region-based active contours: examples, principle and parameter influence. Both synthetic images (top and middle images) and a PC-MRI image of the carotid are considered (bottom images). Top: examples, on synthetic images, of deformed contours obtained in different situations (various shapes). (A) Principle. (B) Influence of the weight of the inner band with respect to the outer band (e.g. $\omega = 0.2$ means that more weight is given to the outer band). (C) Influence of the bandwidth (deformation is altered by surrounding structures when enlarging the bandwidth). (D) Influence of the initialization.

intensities μ_{in} and μ_{out}:

$$\mu_{in} = \frac{\iint_{B_{in}} I_a(x,y)\,dx\,dy}{\iint_{B_{in}} dx\,dy}, \quad \mu_{out} = \frac{\iint_{B_{out}} I_a(x,y)\,dx\,dy}{\iint_{B_{out}} dx\,dy} \tag{4}$$

Concerning the implementation, our proposal is based on explicit contours for runtime performance reasons (generally considered implicit contours (level-sets) [13–15] being time-consuming). The counterpart of considering an explicit representation is the difficulty to manage topological changes (intrinsically managed by implicit representation). This is not a limitation in our context because PC-MRI sequences are not acquired at carotid bifurcations, therefore excluding possible changes of topology (this being not the case in previously mentioned applications based on CTA or MRA images [4–6,16]).

Contour deformation is performed through the minimization of the energy functional E using a gradient descent. During the deformation, each contour point p_i of the considered explicit representation is iteratively moved (step t to $t+1$):

$$p_i^{t+1} = p_i^t + \Delta t f(p_i^t) \tag{5}$$

where $f(p_i^t)$ is the force vector, combining a regularization force (α term in Eq. (2), see [16] for details) and a region force $f_r(p_i^t)$ being defined, in our case, by

$$f_r(p_i^t) = [\omega \times (I_a(p_i^t) - \mu_{in})^2 - (1-\omega) \times (I_a(p_i^t) - \mu_{out})^2] n_i \tag{6}$$

where n_i is the normal to the curve at the point p_i^t. These relationships show how the deformation is implemented. Both μ_{in} and μ_{out} are recomputed after each iteration (see [16] for details). Note that, to ensure a stable distribution of contour points, the contour is reparameterized after each iteration (see [16] for details).

Compared to [16], we consider an asymmetric configuration (as in Chan-Vese model [18]). This regards the ω parameter in Eq. (2) which determines the relative weight given to the inner band with respect to the outer one. As illustrated in Fig. 2B, a small ω (e.g. 0.2) involves a large carotid lumen because deformation favors a homogeneous outer band intensity (i.e. $1-0.2=0.8$ according to Eq. (2)). By opposition, a high ω (e.g. 0.8 in Fig. 2B) favors a homogeneous inner band intensity, this occurring at lumen center, therefore attracting the contour towards lumen center (underestimate of the lumen area). In our proposal, this parameter can be controled by the expert in order to automatically generate the appropriate segmentation, according to his visual perception of lumen boundaries. This aspect is further studied and discussed in Sections 3.3 and 4.

As underlined in [16], the use of a narrow band enables to be less sensitive to anatomical structures surrounding carotid (e.g. jugular vein indicated in Fig. 1A). Fig. 2C reports deformed contours obtained using bandwidths of 2 and 8 pixels (ω is tuned, in each case, in order to obtain similar deformation result on the upper right part of the carotid). In each case, outer band boundary is also reported: for a bandwidth of 8 pixels, B_{out} covers (in its lower left part) the surrounding structure (jugular vein) which attracts the contour.

As illustrated in Fig. 2D, the initialization must be carefully performed so that the contour is not attracted by surrounding structure. This aspect is further studied in Section 3.3 and discussed in Section 4.

2.3. Measurements

Considered measurements are related to evolution, over the sequence, of the lumen area and blood flow.

The area A_t of carotid lumen at a given instant t is computed according to the following equation:

$$A_t = P_s \times \iint_{\Omega_t} dx\,dy \tag{7}$$

where Ω_t defines the pixel domain bounded by the deformed contour, and P_s is the pixel size (in cm^2).

The blood flow (typically expressed in cm^3 per second or millimeter per second) is computed at a given sequence instant t according to the following expression:

$$F_t = P_s \times \iint_{\Omega_t} \tilde{v}_t(x,y)\,dx\,dy \tag{8}$$

where the velocity element $\tilde{v}_t(x,y)$ (typically in cm per second) is computed for each pixel, from the value of the intensity in the phase image $I_{P,t}(x,y)$, weighted by the encoding velocity *VENC* and the related bit range n (both values being acquisition parameters associated with specific DICOM tags):

$$\tilde{v}_t(x,y) = (I_{p,t}(x,y)+1).\frac{VENC}{2^n-1} \tag{9}$$

3. Results

The proposed approach is experimentally evaluated according to the evaluation protocol described in Section 3.1, using both a homemade flow MRI phantom (Section 3.2) and a set of real sequences from healthy volunteers (Section 3.3). Experiments performed using the phantom allow an objective evaluation in a controled environment: segmentation accuracy is evaluated by comparison to the known geometry and size of the "synthetic" carotid. Complementary experiments performed on the set of real sequences allow us to evaluate our proposal on real carotids.

3.1. Evaluation protocol

Sequences are series of 20 images acquired with a clinical 1.5 T MRI Siemens scanner using retrospective navigator gating with ECG. Acquired images are of size 192×132 pixels.

For comparison purposes, all sequences are manually segmented by one radiologist expert using a homemade software, allowing to simultaneously visualize manual drawings superimposed on both anatomical and phase images.

For experiment, the code provided by the author of [16] is used[1]: the reader can therefore refer to this code for implementation details. To underline the relevance of the narrow band region-based energy for guiding contour deformation, segmentations are compared to those obtained using the traditional edge-based active contours (comparison with the related work on aorta [9]), where external energy is based on the gradient vector flow (GVF) [20]:

$$E = \alpha \int_0^1 |C'(u)|^2\,du - \lambda \int_0^1 V(C(u))\,du \tag{10}$$

where V corresponds to the gradient vector flow computed from the gradient of the anatomical image. This allows us to more easily capture contour even if initialization is far from the target [20].

Segmentations are evaluated by measuring the similarity index [21], the lumen area error and the blood flow error. The similarity index is measured between the obtained contour and a reference contour. The reference contour is either the one obtained by manual drawing (case of experiments on healthy volunteers) or the one corresponding to the known shape and size of carotids (case of experiments on the phantom sequence). The similarity index $S(R,A)$ is a standard evaluation technique for measuring the similarity between two segmented regions [21]:

$$S(R,A) = 2|R \cap A| / (|R| + |A|), \tag{11}$$

[1] http://liris.cnrs.fr/julien.mille/code.html

where R (respectively A) corresponds to the reference region (respectively the segmented region). $S(R,A)$ ranges between 0 (complete dissimilarity) and 1 (perfect similarity) [21].

Both area and blood flow errors are measured for each image of each sequence and are expressed in percentage of measurements obtained using the reference contour (errors averaged over sequences are considered).

For experiments, the initial circle is automatically computed from estimated center and radius of the manual segmentation of the first image of the sequence. In order to avoid being corrupted by structures surrounding the carotid, the radius of the initial circle is set to 80% of radius estimated from the manual segmentation (initialization within carotid lumen). Note that the influence of this initialization is studied in Section 3.3.

The parameter related to internal regularization (see Eqs. (2) and (10)) is set to α=0.5 (as in [16]). Thirty iterations are considered for the deformation procedure (i.e. no particular stopping criterion is considered for assessing its converge): this is experimentally observed to be sufficient for convergence.

For the edge-based approach, the computation of the GVF is performed by considering 5 iterations and 0.2 (μ in [20]) for the regularization parameter (as in [20]). This enables us to obtain a smooth GVF to appropriately capture and attract contours towards highest image gradients.

Other parameters are optimized to maximize the similarity index (averaged over sequences). For the edge-based approach, a high value is considered for the external energy ($\lambda = 15$ in Eq. (10)), in order to significantly favor contour attraction towards high gradients. For the considered region-based approach, the weight of external energy is set $\lambda = 0.5$ (Eq. (2)) in order to obtain the highest similarity.

Bandwidth (B_{in} and B_{out}, being considered identical [16]) and ω (see Eq. (2)) are set to different values for experiments on phantom and healthy volunteers (specifically optimized in both cases). For experiments on the phantom sequence, the bandwidth is set to 18 pixels and $\omega = 0.2$ in order to obtain the highest similarity. For experiments on healthy volunteers, the optimal value for the bandwidth is 2 pixels. The value of ω is optimized for each carotid. Indeed, as detailed in Section 2, this parameter can be interactively tuned in clinical routine, and can therefore be specifically optimized for each carotid in experiments. Results (Section 3.3) obtained with the $\omega = 0.5$ are also reported for comparison purposes: this corresponds to the optimal value maximizing the similarity over all sequences, and corresponds to the symmetric configuration considered in [16].

3.2. Phantom

The phantom (Fig. 3A) consists in a motor generating a pulsative flow at a controled frequency (peristaltic pump DS-M series from Boyser), the value of which being defined according to the desired cardiac frequency. The motor injects the liquid within a tube whose internal diameter is 7 mm (order of size of the carotid diameter [11]), corresponding to a section area of 0.38 cm^2. Fig. 3B shows an example of acquired image (both anatomical and phase images at a given instant).

Both phantom carotids of the sequence are analyzed and results are compared to segmentations, areas and flows obtained using the known tube size and geometry (perfect circle of 0.38 cm^2). Table 1 provides statistics related to these results. It appears that edge-based approach involves an underestimate of areas of 18.68%, compared to an error of only 2.73% for the region-based approach. Concerning flows, the edge-based approach leads to an underestimate of 4.95% (1.23% for region-based active contours). Note that segmentations performed by the expert involve errors of 4.74% for areas and 1.37% for flows (slightly worse than results obtained by the considered region-based approach).

Fig. 3 provides some segmentation results (Fig. 3C) as well area and flow evolutions (Fig. 3D) for one of the two phantom carotids. Similar to manual drawing, the region-based approach encompasses the entire vessel lumen while the edge-based active contour is attracted by high gradients, lying within (although close to boundaries) carotid lumen. This is illustrated by Fig. 3C with contours superimposed on the anatomical image together with the gradient vector flow. This explains the underestimation of measured areas, and therefore on measured flows (see mean areas and flows in Table 1, and their evolutions in Fig. 3D).

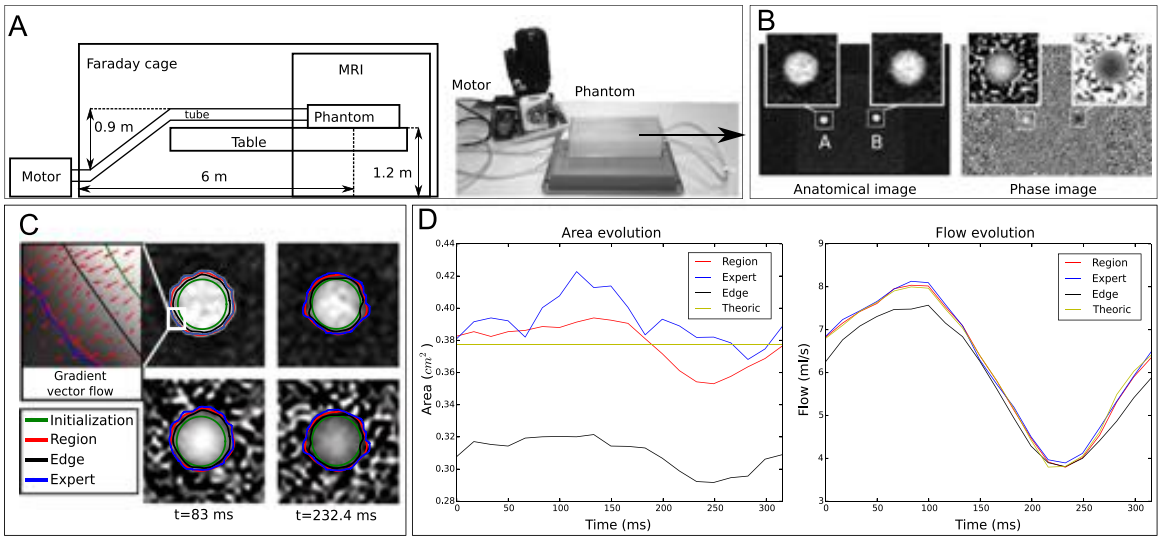

Fig. 3. Experiments on the phantom carotid. (A) Setup, including a schema of the acquisition system and a picture of both motor and phantom. (B) Example of acquired anatomical and phase images. (C) Example of segmentations of region A (contours superimposed on both anatomical and phase images at two instants) using region-based active contours (Region), edge-base ones (Edge) and manual drawing (Expert). (D) Related measured area and flow evolutions for each segmentation, including the one related to the known tube geometry (Theoric).

Table 1
Results on the phantom sequence using the proposed region-based approach (Region), the edge-based one (Edge), the manual drawing (Expert) and the known tube geometry (Theoric). Errors are computed by comparison with measurements obtained using the theoric contour. Values in parenthesis concern the two phantom carotids.

Measurement	Region	Edge	Expert	Theoric
Mean similarity (index)	0.97 (0.97–0.97)	0.90 (0.90–0.89)	0.96 (0.96–0.96)	– –
Mean area (cm^2)	0.38 (0.38–0.37)	0.31 (0.31–0.30)	0.38 (0.39–0.37)	0.38 (0.38–0.38)
Mean area error (%)	2.73 (2.81–2.66)	18.68 (17.73–19.64)	4.74 (4.16–5.31)	– –
Mean flow (ml/s)	6.16 (6.16–6.16)	5.82 (5.85–5.77)	6.16 (6.20–6.11)	6.13 (6.18–6.08)
Mean flow error (%)	1.23 (1.03–1.43)	4.95 (4.88–5.03)	1.37 (1.23–1.51)	– –

Table 2
Results on the healthy volunteers using the proposed region-based approach (Region), the edge-based one (Edge) and the manual drawing (Expert). Results obtained with a constant ω value of 0.5 are also provided (single value optimized over the entire database). Errors are computed by comparison with measurements obtained using the manual drawing. Values in parenthesis are standard deviations.

Measurement	Region	Edge	Region ($\omega=0.5$)	Expert
Mean similarity (index)	0.92 (0.02)	0.73 (0.09)	0.91 (0.02)	– –
Mean area (cm^2)	0.47 (0.1)	0.28 (0.08)	0.49 (0.01)	0.47 (0.09)
Mean area error (%)	7.61 (2.81)	40.88 (12.76)	13.39 (6.97)	– –
Mean flow (ml/s)	7.41 (2.34)	5.47 (1.68)	7.64 (2.33)	7.50 (2.32)
Mean flow error (%)	4.60 (2.26)	26.86 (9.69)	6.70 (3.56)	– –

Another interesting observation regards the variability of expert drawing over sequences. Manual contours are less smooth than those obtained using the proposed method, the shape of which being closer to the theoric circle (see superimposed contours in Fig. 3C). Similarly, area variations measured by the expert also appears less smooth than variations observed by the proposed approach (see area evolution in Fig. 3D). This underlines the difficulty of building appropriate manual segmentations required for evaluation purposes [17].

Note that this evaluation is performed by comparison with a perfect circle of a constant diameter, which is a rough approximation because the area varies (slight variation compared to variations observed on real sequences – see Section 3.3) over the sequence, due to liquid pressure variations (see area and flow evolution observed using manual drawing and active contours). Therefore, errors provided in Table 1 are overestimated because one compares segmentations to a perfect circle of constant diameter.

As a result, this experiment using a carotid phantom shows that the edge-based approach significantly underestimates areas compared to our proposal leading to similar results (and even more regular) than those manually built by an expert.

3.3. Healthy volunteers

These experiments involve the segmentation of 20 carotids from 10 healthy volunteers.

Table 2 provides statistics related to segmentation results, and the two histograms on the top of Fig. 4 indicate the distribution of errors over carotids of healthy volunteers.

As on the phantom sequence, it appears that the edge-based approach involves a significant underestimate of areas of 40.88%, compared to an error of only 7.41% for the region-based approach. Concerning flows, the edge-based approach leads to an underestimate of 26.86% (only 4.60% of error for region-based active contours). This confirms the relevance of using a region-based energy term rather than an edge-based one.

Table 2 also reports results obtained using single common value for ω ($\omega=0.5$, optimized over the entire database). Although significantly better than those obtained using the edge-based approach, these results are worse than those obtained using a specific ω optimized for each carotid (optimal values ranging from $\omega=0.4$ and $\omega=0.6$).

Fig. 4 details results obtained for two particular carotids (carotid 5a and 10b). Carotid 5a is an example of result observed for most carotids of the database (note that the example reported in Fig. 1 corresponds to carotid 3a). In particular, contours obtained by our proposal match the overall shape of expert contours, and the edge-based approach underestimates the area (visually more significantly than in the case of the phantom). Moreover, the area evolution follows the flow one (area increases with blood pressure) and manual drawing involves more fluctuations (as for the phantom).

Carotid 10b (detailed in Fig. 4) corresponds to the highest measured error resulting from our approach. One also indicates results obtained on carotid 10b using $\omega=0.2$ (as for the phantom), leading to a larger vessel lumen. The area evolution becomes more similar to the one of flows, compared to evolutions measured with $\omega=0.6$. Moreover, estimated areas are more coherent with the mean area observed over the entire database of healthy volunteers (0.47 cm^2 reported in Table 2). A post-analysis of this sequence by the expert has visually validated the relevance of the result obtained with $\omega=0.2$, compared to the one obtained with the a priori optimal $\omega=0.6$ (i.e. ω maximizing the similarity with manual delineation). Measured error for carotid 10b with $\omega=0.6$ is therefore likely overestimated. Such a situation illustrates the difficulty of obtaining an objective ground truth for algorithms' validation. It also underlines the difficulty of performing a correct manual segmentation for a clinical use, and therefore justifying the relevance of a (semi-)automated analysis method.

Another aspect of experiments concerns parameters: Fig. 4 – top-right shows the influence, measured in terms of mean similarity over the database, of two important parameters, namely the initialization (in percentage of radius estimated from the manual drawing) and the bandwidth.

Concerning the initialization, it appears that the circle must be large enough to reach lumen boundary, and small enough to avoid being captured by surrounded structures during the deformation process (as illustrated for one particular case in Fig. 1). Fig. 4 shows that a radius ranging between 70% and 90% of the effective carotid radius appears reasonable. Note that experiments are performed with the optimal value of 80% (see Section 3.1).

Concerning the bandwidth parameter, the best mean similarity is obtained for a value of 2 pixels. Note that a bandwidth of 1 pixel must be avoided because it makes the region-based energy similar to an edge-based energy [16]. Such a thin band of 2 pixels is required to avoid being corrupted by surrounding structures (see Section 2.2 for explanations and Fig. 2C for result examples obtained for bandwidths of 2 and 8 pixels). Note that this is not the case for the phantom sequence where parameter optimization leads to a large band of 18 pixels, because phantom carotids do not

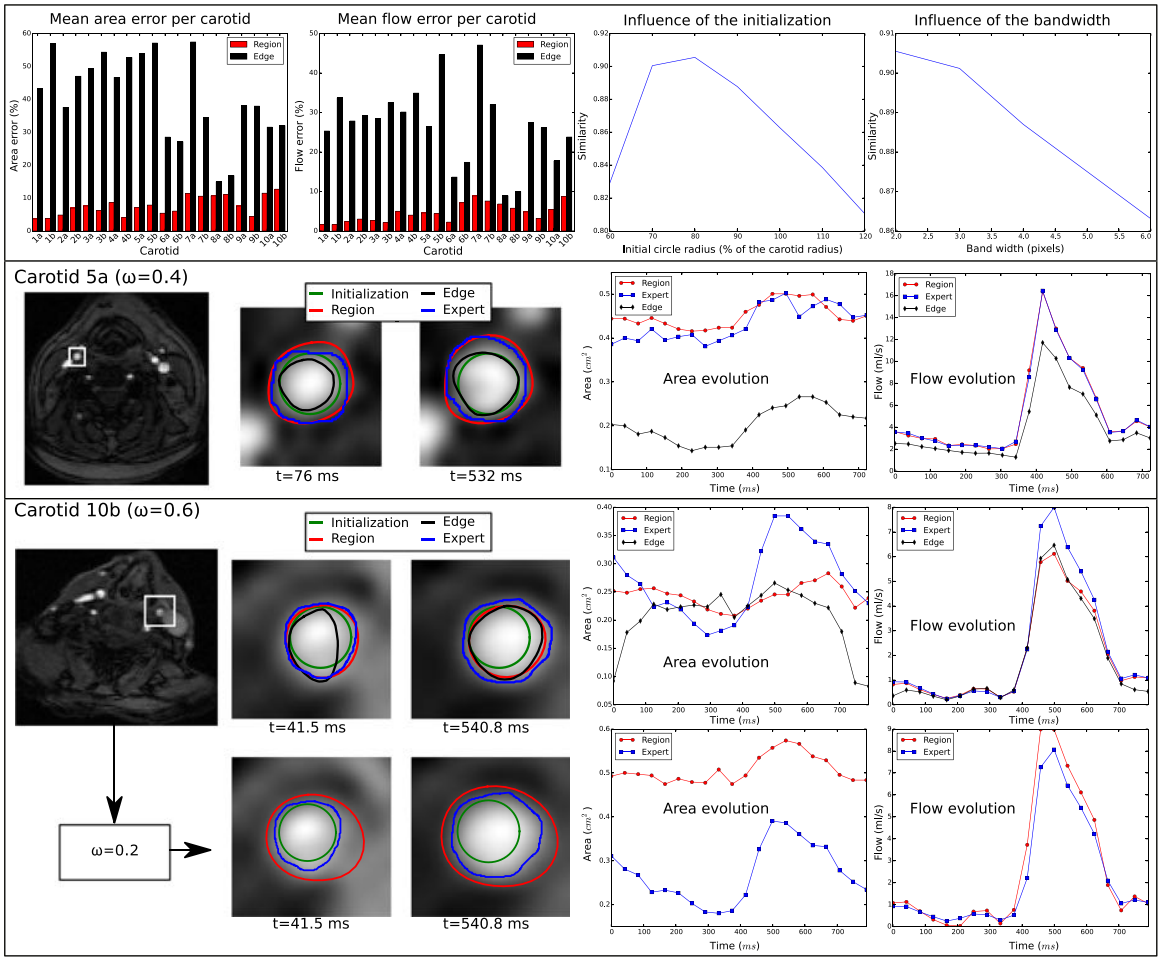

Fig. 4. Experimental results on the healthy volunteers using region-based active contours (Region), edge-base ones (Edge) and manual drawing (Expert): errors per carotid on top-left (carotids 5a and 10b are detailed – corresponding to volunteers number 5 and 10), and influence of both initialization and bandwidth on top-right.

depict any surrounding structures. This justifies the necessity of such narrow band approach, compared to the initially proposed region-based approach where entire inner and outer regions are considered [18].

4. Discussion

The efficiency of the proposed method depends on the image quality, resulting from the acquisition protocol and imaging settings. For instance, the acquisition slice must be perpendicular to the carotid. One also has to avoid motion artefacts (movement minimization), as well as to correctly set imaging parameters, such as, for instance, the velocity encoding one (VENC). As underlined in related works dealing with PC-MRI [10], the VENC must be appropriately chosen (larger that the minimum and maximum value of velocity magnitude in carotid), to avoid aliasing, although this can be corrected (phase unwrapping). It would be interesting to evaluate the influence of these acquisition conditions on the efficiency of the proposal, including their effects on blood flow measurements. This would help in specifying the required and optimal acquisition protocol.

Some preliminary contrast enhancement techniques could be considered, for instance for edge sharpening (e.g. anisotropic filtering for improving the contrast of the carotid). This would probably enable us to use traditional edge-based active contours for instance. This is intrinsically managed, in a unified manner, by the considered region-based method thanks to ω parameter. Note that avoiding such preliminary step also limits the complexity of the entire segmentation procedure (i.e. single segmentation algorithm instead of a combination of several algorithms, each one having its own parameters to estimate).

Concerning the information used for segmentation purposes, our proposal only exploits one information (the anatomical information), the other one (the phase or functional information) being only used for measurements regarding blood flows. As in our case, most recent related works consider only one information for segmentation purposes. For instance, only phase images are used in [7,8], while [9] exploits the anatomical information only. As recently investigated [22,23] (preliminary experiments only), coupling both information may have some potential. The expected benefit would be to overcome the issue related to the lack of contrast appearing in the anatomical image (and the ω parameter to be tuned), by searching, in phase images, for the frontier between flow region and surrounding stationary tissues. The drawback would be the increase of the complexity of the resulting segmentation algorithm, due to related additional parameters to

properly estimate (i.e. parameters related to the phase image and to its combination with the anatomical image).

As mentioned in Section 2, initialization involves two kinds of interaction, one for setting the initial circle and the other one for the ω parameter. From the end-user point of view, setting the initial circle does not require to manually draw the carotid lumen boundary on the first image of the sequence. The circle center can be initialized thanks to a simple interaction, such as a mouse click at the carotid lumen center (as in [9]). The circle radius can be tunable (controled by basic mouse movements for instance), but with an appropriate default fixed radius. According to experiments (Fig. 4, "influence of the initialization"), the fixed default value can be set to 80% of standard known radius of carotids. Considering such a fixed radius enables us to reduce the influence of the variability of end-user interactions, limited to the definition of the circle center (click on the lumen center). The second (optional) initialization parameter (i.e. ω) can be set to a default optimal value of 0.5, according to experiments. It can be modified if needed, using a slider proposing a limited range of values. Theorically, ω is constrained within the $[0-1]$ interval. Practically, this could be reduced to equally spaced values (e.g. 0.1 step), ranging from 0.1 (excluding $\omega=0$) to 0.6 (highest one observed over the database of healthy volunteers). The runtime, excluding the initialization step, is about 0.1 s for processing an entire sequence of 20 images (measured on a MacBookPro with a Core I7 2.2 GHz Intel processor, and using the C++ implementation of Mille [16]). Thanks to this computation time, the user rapidly gets the result (fast visual feedback) and can re-initialize the procedure if needed (e.g. new circle or ω value). This is compliant with the clinical routine because it leads to a really short entire procedure (probably only few seconds in the worst case, if several initializations are required), compared to a tedious manual drawing over the entire sequence. The advantage of controling the initialization, compared to fully automated algorithms with fixed parameters, is the ability to adapt the proposed algorithm to any particular sequence (this is the case of the ω parameter, as experimentally observed for carotid 10b in Section 3.3). Nevertheless, automating the initialization would be interesting to further improve the analysis, keeping the possibility to switch, in case of failure, to the proposed interactive initialization for specific sequences (i.e. patients or pathologies). This automation constitutes one of the next step of this work. It could involve, for instance, the Hough transform, particularly adapted for detecting circular objects such as the carotid (an issue would be to distinguish the carotid from neighboring circular structures such as the jugular vein).

Aside from experiments using the phantom, our proposal is evaluated on healthy volunteers only, as in several recent related works (e.g. 4 and 6 healthy volunteers respectively in [7,8]). Note that only 2 patients are considered in [10], although one of both depicts a pathology. Pathological cases involve less regular lumen boundaries. As recently discussed for 3D PC-MRI focussing on the aorta [8] using active contours, the segmentation should be as feasible and reliable as for healthy volunteers (assuming appropriate acquisition settings, in particular for the VENC parameter). Indeed, active contours should be flexible enough to properly deform towards lumen boundaries, even if they appear irregular, due to a given pathology. This is illustrated on synthetic images in Fig. 2 – top (in particular for the three images on the right side). Nevertheless, this is an important aspect to be confirmed by an additional experimental study. The present study focuses on the proposition of a method for measuring both lumen area and blood flow evolution over the cardiac cycle. The next study will focus on computer aided diagnosis, concerning the application of this method for subjects depicting a pathology. This would enable us to evaluate whether our proposal can help to automatically detect some specific pathologies (by comparison of measurements on healthy volunteers), and to quantify their severity.

A last aspect to be discussed concerns cardiovascular diseases, such as the atheromatous stenosis of the carotid which develops mainly at its bifurcation. In this case, our purpose is not to directly quantify internal carotid stenosis by velocity measurement at the site of the bifurcation. We focus on blood flow quantification, on common carotid artery. The next step, in a clinical study, would be to do the same on internal carotid artery, above the bifurcation, to quantify hemodynamic changes downstream to an internal carotid stenosis (indicating the severity of the stenosis). Except for the requirement of the acquisition and the processing of an additional sequence, this should, a priori, not present any particular difficulty using the proposed algorithm.

5. Conclusion

In this paper, we propose a semi-automated method, using narrow band region-based active contours, for extracting carotid lumen boundary and automatically computing both area and blood flow evolution from PC-MRI sequences. We experimentally show that such region-based active contours are more appropriate than traditional edge-based ones recently considered for the analysis of PC-MRI sequences focusing on the aorta: due to the lower size of carotids and lower contrast at lumen boundaries, such an edge-based approach strongly underestimates vessel area and blood flow. Moreover, experiments, coupling both a phantom sequence and sequences from healthy volunteers, also show the limits, in terms of reliability, of manual drawing.

The work can be used in clinical routine, where the expert only needs to initialize the algorithm by placing a circle within the carotid lumen. If necessary, a single additional parameter enables us to tune, with real-time feedback, the overall extent of segmented contours.

Next step of this work will concern the automation of the initialization using a dedicated algorithm, as well as the evaluation of this work for detecting pathologies and quantifying their severity.

Conflict of interest statement

None declared.

Acknowledgments

This work was supported by a grant from Siemens SAS France.

References

[1] M.B. Srichai, R.P. Lim, S. Wong, V.S. Lee, Cardiovascular applications of phase-contrast MRI, Am. J. Roentgenol. 192 (3) (2009) 662–675.
[2] J.J. Hom, K. Ordovas, G.P. Reddy, Velocity-encoded cine MR imaging in aortic coarctation: functional assessment of hemodynamic events, Radiographics 28 (2) (2008) 407–416.
[3] E.P. Efstathopoulos, G. Patatoukas, I. Pantos, O. Benekos, D. Katritsis, N. L. Kelekis, Measurement of systolic and diastolic arterial wall shear stress in the ascending aorta, Phys. Med. 24 (4) (2008) 196–203.
[4] D. Lesage, E.D. Angelini, I. Bloch, G. Funka-Lea, A review of 3D vessel lumen segmentation techniques: models, features and extraction schemes, Med. Image Anal. 13 (6) (2009) 819–845.
[5] K. Hameeteman, M. Zuluaga, M. Freiman, L. Joskowicz, O. Cuisenaire, L. F. Valencia, M. Gulsun, K. Krissian, J. Mille, W. Wong, M. Orkisz, H. Tek, M. H. Hoyos, F. Benmansour, A. Chung, S. Rozie, M. van Gils, L. van den Borne, J. Sosna, P. Berman, N. Cohen, P. Douek, I. Sanchez, M. Aissat, M. Schaap, C. Metz, G. Krestin, A. van der Lugt, W. Niessen, T. van Walsum, Evaluation

framework for carotid bifurcation lumen segmentation and stenosis grading, Med. Image Anal. 15 (4) (2011) 477–488.
[6] R. Manniesing, M. Schaap, S. Rozie, R. Hameeteman, D. Vukadinovic, A.V. D. Lugt, W. Niessen, Robust CTA lumen segmentation of the atherosclerotic carotid artery bifurcation in a large patient population, Med. Image Anal. 14 (6) (2010) 759–769.
[7] R. van Pelt, H. Nguyen, B. ter Haar Romeny, A. Vilanova, Automated segmentation of blood-flow regions in large thoracic arteries using 3D-cine PC-MRI measurements, Int. J. Comput. Assist. Radiol. Surg. 7 (2) (2012) 217–224.
[8] P. Volonghi, D. Tresoldi, M. Cadioli, A.M. Usuelli, R. Ponzini, U. Morbiducci, A. Esposito, G. Rizzo, Automatic extraction of three-dimensional thoracic aorta geometric model from phase contrast MRI for morphometric and hemodynamic characterization, Magn. Reson. Med. 2015, http://dx.doi.org/10.1002/mrm.25630, in press.
[9] A. Herment, N. Kachenoura, M. Lefort, M. Bensalah, A. Dogui, F. Frouin, E. Mousseaux, A. De Cesare, Automated segmentation of the aorta from phase contrast MR images: validation against expert tracing in healthy volunteers and in patients with a dilated aorta, J. Magn. Reson. Imaging 31 (4) (2010) 881–888.
[10] A. Das, J.P. Wansapura, W.M. Gottliebson, R.K. Banerjee, Methodology for implementing patient-specific spatial boundary condition during a cardiac cycle from phase-contrast MRI for hemodynamic assessment, Med. Image Anal. 19 (1) (2015) 121–136.
[11] P. Reymond, F. Merenda, F. Perren, D. Rüfenacht, N. Stergiopulos, Validation of a one-dimensional model of the systemic arterial tree, Am. J. Physiol.—Heart Circ. Physiol. 297 (1) (2009) 208–222.
[12] G. Trebuchet, J.-B. Fasquel, C. Cavaro-Menard, A. Lecluse, S. Willoteaux, Region-based active contours for computer-aided analysis of carotid phase contrast MRI, in: 2013 IEEE 26th International Symposium on Computer-Based Medical Systems (CBMS), 2013, pp. 47–52.
[13] S. Lankton, A. Tannenbaum, Localizing region-based active contours, IEEE Trans. Image Process. 17 (11) (2008) 2029–2039.
[14] C. Li, C.-Y. Kao, J. Gore, Z. Ding, Minimization of region-scalable fitting energy for image segmentation, IEEE Trans. Image Process. 17 (10) (2008) 1940–1949.
[15] T. Brox, D. Cremers, On local region models and a statistical interpretation of the piecewise smooth Mumford–Shah functional, Int. J. Comput. Vis. 84 (2) (2009) 184–193.
[16] J. Mille, Narrow band region-based active contours and surfaces for 2d and 3d segmentation, Int. J. Comput. Vis. 113 (9) (2009) 946–965.
[17] P. Jannin, E. Krupinski, K. Warfield, Simon, Validation in medical image processing, IEEE Trans. Med. Imaging 25 (11) (2006) 1405–1409.
[18] T. Chan, L. Vese, Active contours without edges, IEEE Trans. Image Process. 10 (2) (2001) 266–277.
[19] M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active contour models, Int. J. Comput. Vis. 1 (1988) 321–331.
[20] C. Xu, J.L. Prince, Snakes, shapes, and gradient vector flow, IEEE Trans. Image Process. 7 (3) (1998) 359–369.
[21] A. Zijdenbos, B. Dawant, R. Margolin, A. Palmer, Morphometric analysis of white matter lesions in MR images: method and validation, IEEE Trans. Med. Imaging 13 (4) (1994) 716–724.
[22] R.L. Janiczek, F. Epstein, S. Acton, Velocity guided segmentation of phase contrast magnetic resonance angiography, in: Proceedings of the 15th IEEE International Conference on Image Processing, 2008, pp. 2264–2267.
[23] G. Trebuchet, J. Fasquel, C. Cavaro-Ménard, S. Willoteaux, Coupling anatomical and functional information for the computer-aided delineation of phase-contrast MRI images using active contours, in: 3rd International Conference on Image Processing Theory Tools and Applications, IPTA 2012, 15–18 October 2012, Istanbul, Turkey, 2012, pp. 172–177.

An aggregation plateform for IoT-based healthcare: illustration for bioimpedancemetry, temperature and fatigue level monitoring.

Antoine Jamin[1], Jean-Baptiste Fasquel[1], Mehdi Lhommeau[1], Eva Cornet[2], Sophie Abadie-Lacourtoisie[4], Samir Henni[1,3], and Georges Leftheriotis[3]

[1] LARIS-ISTIA, Université d'Angers, 62 Avenue notre Dame du Lac, 49000 Angers, France
[2] Bioparhom, 89 rue Pierre et Marie Curie, 73290 La Motte Servolex, France
[3] University Hospital Center of Angers, Vascular Department and MITOVASC, Angers, France
[4] Institute of Cancer Research in Western France, Paul Papin Center, Angers, France

Abstract. In this paper, we detail an in-home aggregation plateform for monitoring physiological parameters, and involving two objective physical sensors (bio-impedanceter and thermometer) and a subjective one (fatigue level perceived by the patient). This plateform uses modern IoT-related technologies such as embedded systems (Raspberry Pi and Arduino) and the MQTT communication protocol. Compared to many related works, monitoring is enterely achieved using a box as a central element, while the mobile device (tablet) is only used for controlling the acquisition procedure using a simple web browser, without any specific application. An example of a time stamped set of acquired data is shown, based on the in-home monitoring of healthy volunteers.

Key words: IoT, healthcare, bio-impedancemetry, MQTT, Raspberry Pi, Arduino

1 Introduction

Internet of Things is a new paradigm offering a large number of possibilities, as underlined in a recent review [1]. In healthcare (see the recent overview [2]), such a paradigm facilitates the interconnection of medical devices and data, with various applications such as home tele-monitoring of patients or elderly people for instance. Many sensors are now available for monitoring many parameters (e.g. heart rate, blood flow, blood pressure, temperature, muscle contraction, weight...) with various technologies and distributed software architectures for communication purposes.

This paper focuses on the conception of an in-home aggregation plateform.

The main contribution regards the detailed description of both hardware and software aspects of the particular plateform that we developed, including various devices such as tablet, two particular sensors and the coupling of both Raspberry Pi and Arduino embedded systems. Our purpose is to show how such a plateform

can be developed for specific applications. A part of this contribution regards the use of the MQTT protocol [3], particularly appropriated for IoT-based applications although rarely considered in healthcare, as recently underlined [4].

Another part of the contribution concerns the heterogenous nature of monitored parameters: we consider both two objective parameters (i.e. measured by sensors) and a subjective one (fatigue level - can be considered as a subjective sensor). In our opinion, most IoT-based healthcare system focus on physical sensors althought, for healthcare, additional subjective parameters such as fatigue level, pain level,... may be meaningful from a clinical point of view. In this paper, note that we also consider a bioimpedancemeter, such a sensor being rarely considered (e.g. compared to previously mentioned sensors).

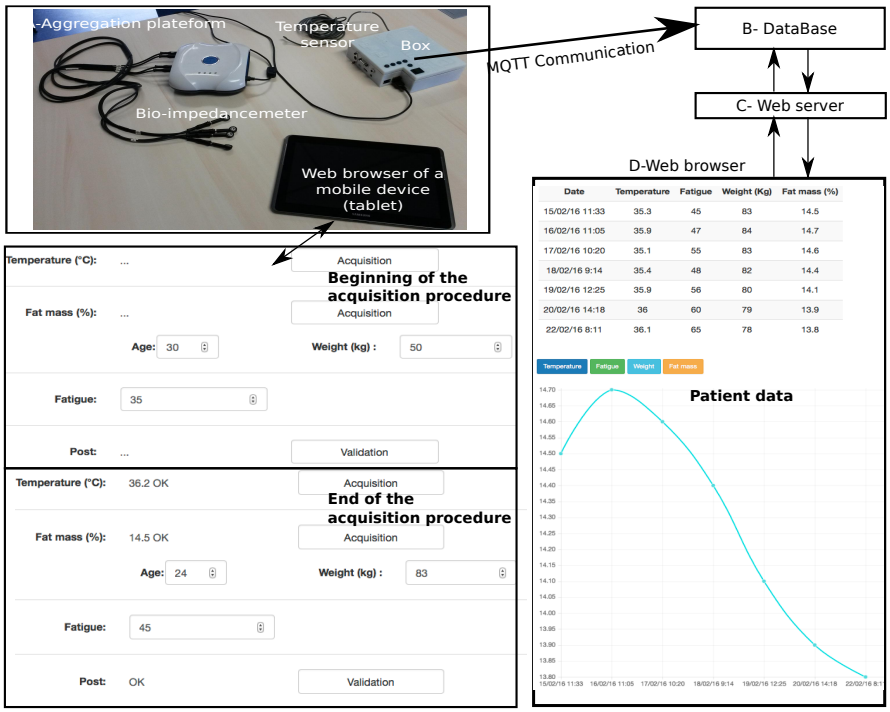

Fig. 1. Overview of the developed aggregation plateform (A) integrated a box (core element) and several external devices. This plateform communicates with a distant database (B) using the MQTT communication protocol, the content of which being rendered to render data within a web browser (D) thanks to web server (C).

Section 2 briefly presents an overview of the developed system, while section 3 focuses on its hardware and software architecture. Section 4 aims at discussing some aspects of this work.

2 System overview

Figure 1 provides an overview of the proposed plateform, including a bioimpedancemeter, a temperature sensor, a mobile device (tablet) and a box for aggregating data and then posting them to the database using the MQTT communication protocol [3].

The bioimpedancemeter (Z-metrix developed by Bioparhom [5]) allows the measurement of various physiological parameters (fat mass, lean mass, total body water, extracellular water, ...).

The temperature sensor is part of the e-health sensor plateform developed by cooking-hacks [6]. Althought the box is conceived to plug 10 sensors (i.e. ECG, SPO2, EMG,...), only the temperature sensor is considered in the paper.

The mobile device is used to interacts with the box using a web browser. This allows to enter parameters that are required to perform measurements (weight and height in our case, being required for fat mass computation using bioimpedancemetry). The mobile device also enables to trigger measurements (i.e. bioimpedancemetry and temperature), acquired values being finally returned and rendered. Other information, useful for health state monitoring, can be entered by the patient, regardless any sensor (subjective sensor mentioned in the introduction). In our case, this concerns the fatigue level (value ranging from 0 to 100). Figure 1-bottom-left provides two snap-shots of the web browser, at the beginning (top) and at the end (bottom) of the acquisition procedure, with acquired values and a transmission acknowledgement.

All these components (temperature, bioimpedancemeter, mobile device) communicate through the central element: the "box". The box aggregates all data (measurements from sensors, information entered by the patient such as the fatigue level) and post them to a distant database (figure 1-B) using the MQTT communication protocol.

A web server (figure 1-C) can finally be used to access to information stored in the database, for rendering purposes. Figure 1-D provides a snapshot of such rendering (time stamped measurements) thanks to the dedicated web server we developed.

3 Software and hardware architecture

Figure 2 provides a synthetic view of the implemented architecture. Section 3.1 details the composition of box, and section 3.2 concerns the MQTT protocol.

3.1 Aggregation plateform

In terms of hardware, the box mainly includes a Raspberry Pi and an Arduino controler together with the Arduino shield developed by Cooking Hacks, for plugging related sensors (in particular the temperature sensor). The bioimpedancemeter is connected through a wired USB connection. Additionnal elements are packaged within the box (not detailed for clarity), such as two wifi dongles for communication (with the mobile device and with the database), a battery

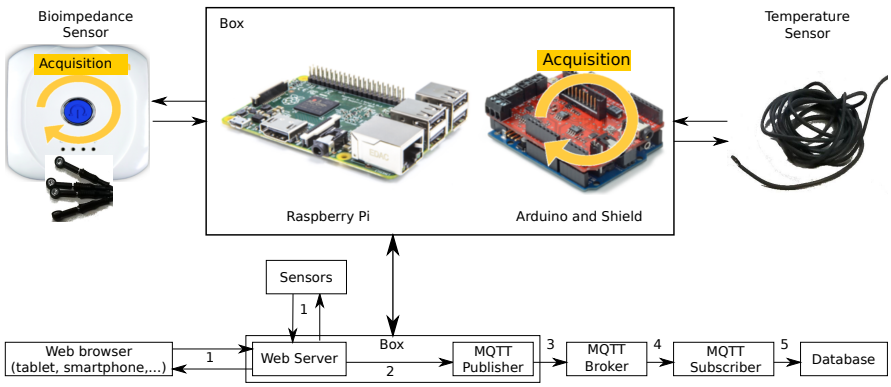

Fig. 2. Architecture overview, including software and hardware components.

and an energy management system, so that the box can work in an autonomous manner. Note that, as underlined in section 4, communication with sensors and mobile device can be modified or extended (e.g. both bio-impedancemeter and mobile device support bluetooth communication).

In terms of software, a web server runs on the Raspberry Pi so that the mobile device can get connected to the box using a web browser. Figure 2-1 models interactions between the mobile device (web browser) and the web server. It also models interactions between the web server and underlying sensors (i.e. acquisition is parameterized and triggered from the web browser as illustrated by figure 1-bottom-left). A REST architecture is considered for web server: each REST resource corresponds to a specific item (i.e. bioimpedancemetry, temperature, fatigue level), with related specific code, ensuring the separation of concerns and the modularity of the application. Note that the specific code related to the bioimpedancemetry embeds fat mass computation from electrical values returned by the sensor. For the temperature sensor, the related REST resource interacts with the Arduino board managing the acquisition (using the library developed by Cooking Hacks [6]).

When the acquisition procedure ends, the user triggers (dedicated REST resource) the post of the data to the distant database using MQTT.

All codes running on the Raspberry Pi are written with the Python language (web server, fat mass computation, communication with the Arduino and database with MQTT), using appropriate libraries.

3.2 MQTT

The principle and features of the MQTT protocol are described in [3] and has been recently considered in the context of healthcare [4]. Such a technology is particularly useful in the field of internet of things. One of the main feature is the related small code footprint and required network bandwidth.

Such a protocol is based on three elements: the publisher, the subscriber and the broker. The publisher publishes information on a certain topic, the subscriber subscribes to (a) topic(s) and receives related published messages. The intermediate entity is the broker, known by both subscribers and publishers. The broker filters all incoming messages and distributes them according to the topic and the subcriptions. Data exchange can be securized thanks to both encryption and authentication mechanisms, this being crucial for healthcare systems.

In our case, a topic corresponds to a patient (specific patient identifier). When the acquisition procedure ends, a REST resource triggers the diffusion (figure 2-2) of the acquired data on the related topic to the broker (figure 2-3). The subscriber receives the message (figure 2-4) and updates the database (figure 2-5).

4 Discussion

This system has been used by healthy volunteers for testing purposes. Acquisitions have been done daily for home during a couple of days. A single distant computer has been considered for running the broker (Mosquitto), the subscriber (Python), the database (MongoDB) and webserver (NodeJS): figure 1-bottom-right provides a snap-shot of the monitored data.

Hereafter, we shortly discuss two aspects: the use of a dedicated aggregator rather than the mobile device, and the ability to integrate additionnal sensors.

Many recent related works consider a mobile device instead of a dedicated aggregator [4, 7] (the mobile device plays the role of aggregator and communicates with the distant broker). For instance, in [4] the mobile device acquires data from ECG and Oxymeter sensors using Bluetooth and a dedicated Android application. The mobile device also embeds the code for publishing data using the MQTT protocol. The advantage of such architecture is that no dedicated box is required, only a dedicated Android application is needed. In our sense, the main drawback of such architecture concerns the communication with sensors: although most mobile devices provide many communication components (e.g. wifi, sim, bluetooth, nfc,...), their main limitation concerns the limited number of wired connections. In our case, as the local server considered in [8], the use of a box offering more connectivity (in particular wired connection) is therefore a relevant alternative. This furthermore allows to decouple the personal mobile device of the patient (e.g. including a GSM connection for general purpose and personal use) from the device used for health monitoring (e.g. which could provide another GSM connection but dedicated to health monitoring). Despite connectivity, such an aggregator involves a dedicated system no only for receiving data but also for performing computations such as fat mass in our case (possible using personal mobile device but at the cost of additional energy consumption). Our proposal has similarities with the local server presented in [8].

In terms of evolutivity, any new sensor can be integrated within the proposed plateform by connected the device to the Raspberry Pi either a wired connection, or a wireless connection (e.g. bluetooth). At software level, this involves

the integration of the corresponding REST resource to web server running on the Raspberry Pi (binding between the tablet and physical sensor). The mobile device basically remains a "touch screen" interacting with the box (through the web browser).

5 Conclusion

This work provides a detailed example of in-home aggregation plateform using standard modern technologies. Next steps will concern the exploitation of this system real healthcare applications.

Acknowledgement

This work is granted by the french league against cancer ("ligue contre le cancer 49"), with clinical trials identifier NCT02161978. Authors thank Franck Mercier, research engineer at LARIS-ISTIA-University of Angers, for the development of some hardware elements of the box. Thanks to the students of the engineering school ISTIA who participated to software developments: Mehdi Bellaj, Pierre Cochard, Mathieu Colas, Antoine Jouet, Audrey Lebret, Julien Monnier, Alexandre Ortiz, Dimitri Robin and Alexis Teixeira.

References

1. Borgia, E.: The Internet of Things vision: Key features, applications and open issues. Computer Communications, 1–31 (2014)
2. Yin, Y., Zeng, Y., Chen, X., Fan, Y.: The internet of things in healthcare: An overview. Journal of Industrial Information Integration, 1, 3–13 (2016)
3. Message Queuing Telemetry Transport (MQTT), http://mqtt.org
4. Barata, D., Louzada, G., Carreiro, A., Damasceno, A.: System of Acquisition, Transmission, Storage and Visualization of Pulse Oximeter and ECG Data Using Android and MQTT. Procedia Technology, 9, 1265–1272 (2013)
5. Bioparhom, http://www.bioparhom.com/en
6. Cooking Hacks, e-health sensor plateform, http://www.cooking-hacks.com
7. Hussaina, A., Wenbia, R., Lopes da Silvab, A., Nadhera, M., Mudhisha, M.: Health and emergency-care platform for the elderly and disabled people in the Smart City. Journal of Systems and Software, 110, 253–263 (2015)
8. Mano, L.Y., Faical, B.S., Nakamura, L.H.V., Gomes, P.H., Libralon, G.L., Meneguete, R.I., Filho, G.P.R., Giancristofaro, G.T., Pessin, G., Krishnamachari, B., Ueyama, J.: Exploiting IoT technologies for enhancing Health Smart Homes through patient identification and emotion recognition. Computer Communications, 1–13 (2016)

COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE 81 (2006) 1–7

journal homepage: www.intl.elsevierhealth.com/journals/cmpb

An efficient and generic extension to ITK to process arbitrary shaped regions of interest

Jean-Baptiste Fasquel, Vincent Agnus, Julien Lamy*

IRCAD, 1 Place de l'Hôpital, F-67091 Strasbourg, France

ARTICLE INFO

Article history:
Received 6 May 2005
Received in revised form 12 July 2005
Accepted 22 September 2005

Keywords:
Medical image processing
Regions of interest
Code reusability
ITK

ABSTRACT

The paper describes a software method to extend ITK (Insight ToolKit, supported by the National Library of Medicine), leading to ITK++. This method, which is based on the extension of the iterator design pattern, allows the processing of regions of interest with arbitrary shapes, without modifying the existing ITK code. We experimentally evaluate this work by considering the practical case of the liver vessel segmentation from CT-scan images, where it is pertinent to constrain processings to the liver area. Experimental results clearly prove the interest of this work: for instance, the anisotropic filtering of this area is performed in only 16 s with our proposed solution, while it takes 52 s using the native ITK framework. A major advantage of this method is that only add-ons are performed: this facilitates the further evaluation of ITK++ while preserving the native ITK framework.

1. Introduction

Nowadays, in modern medicine, there is a growing need for medical image analysis systems and softwares, to assist practitioners: computer-aided diagnosis, surgical planning, simulation, robotically assisted surgical intervention, etc. An important element of such systems and softwares is the image processing library which is expected to provide efficient algorithms allowing to automate and to improve medical data analysis. Therefore, it appears crucial to correctly design such a library to easily integrate new functionalities. This is furthermore required in this research field which is in permanent evolution and which involves a large community of scientists as many new efficient algorithms or pertinent alternatives are regularly proposed and published. To facilitate the integration of these developments, the code of image processing libraries must be generic enough to be easily reused and improved. For example, it appears highly convenient to write the code of an algorithm only once and to be able to use it whatever the spatial dimensionality of images (e.g. mammograms are 2D images, CT-scans are 3D images) or the analyzed image area (i.e. region of interest). Runtime performance is an additional constraint which must also be taken into account so that resulting systems are usable: fast diagnosis and patient modeling (e.g. for surgical planning in emergency), fast medical data analysis and registration for robot guidance during a surgical act (which can also include a system for heart beat or breathing compensation), etc.

The literature dedicated to the methodology for designing an image processing library seems to be particularly sparse. To our knowledge, only Köthe [1]recently published a detailed study dedicated to generic programming for image processing. However, it does not consider the genericity with respect to the region of interest. Among existing image processing libraries, the ITK library (Insight Segmentation and Registration Toolkit [2]), supported by the National Library of Medicine, appears to be a rich and well-designed framework because it provides a large and increasing number of sophisticated algorithms which can be applied, without code rewriting, whatever the spatial dimensionality and intensity (e.g. scalar or multi-

* Corresponding author.
E-mail address: Jean-Baptiste.Fasquel@ircad.u-strasbg.fr (J.-B. Fasquel).
URL: http://www.ircad.org.

0169-2607/$ – see front matter © 2005 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.cmpb.2005.09.006

modal images) of the medical image under analysis. However, ITK is not generic with respect to the region of interest (ROI) because only rectangular regions are managed. For example, it is not allowed to constrain processings to a given irregularly shaped organ (e.g. liver), therefore leading to an important waste of time. Moreover, processing efficiency can be deteriorated because irrelevant data are taken into account (e.g. data outside the liver when segmenting the liver tumors). For instance, in the case of a histogram based local area thresholding, the use of the overall image can lead to an incorrect determination of the optimal threshold.

In this paper, we propose and implement a method to extend, through minimal add-ons, the current ITK framework so that it becomes generic with respect to ROIs. This work has been recently shortly presented [3]and we provide hereafter a more detailed and complete description of this improvement.

Section 2deals with ROIs in image processing, and specifically with medical data. In Section 3, we describe the proposed improvement, leading to ITK++. Then, we present an experimental evaluation of the resulting framework in a practical use case dedicated to liver vessel segmentation. Finally, we discuss and conclude on this work.

2. Regions of interest in medical image processing

A region of interest (ROI) defines the set of image points to be scanned and processed. As previously underlined, ROIs are of great interest in image processing to avoid processing unnecessary regions of an image: this can improve accuracy (results are not degraded by irrelevant information, see [4]for illustration purposes) and speed up processings (less image points are computed).

Rectangular ROIs are the most encountered ones in image processing systems. In our sense, they are preferred because their geometry is relatively easy to manage (only a box is required). ROIs can also be related to non-rectangular geometric shapes (e.g. cylinder, sphere) and irregularly shaped ROIs (e.g. organs). A reference image (e.g. a mask or an equation) is then required. In our opinion, this is almost the most interesting kind of ROI because it is closer to the reality of shapes encountered in computer vision, as objects or structures of interest are rarely rectangular. For medical image analysis, where one deals with anatomical or pathological structures, such arbitrary shaped ROIs appear really appropriate: for instance, it is highly convenient to restrict processings to the liver area for liver vessel segmentation.

The concept of ROI can also be further extended to other situations. For example, when resizing images (using interpolation) or performing multiresolution based analysis [5], it could be interesting to access regularly spaced image points: e.g. $I_k(x, y, z) = [I_{k-1} * g](2x, 2y, 2z)$ (I_k image points are computed by filtering every two points of the I_{k-1} image with the Gaussian g). The ROI is then defined by a translation vector representing the spacing between consecutive image points. Moreover, it can be convenient to randomly scan image points (e.g. for statistical measurements in image energy based algorithms [6]) or to scan points verifying a predefined condition (e.g. region growing based processes [5,7]).

3. Extending ITK

Hereafter, we review fundamentals of ITK and present our contribution. The programming language considered is C++, and it is assumed that concepts such as design patterns [8], generic programming [9]and template mechanism [10]are known.

3.1. *Fundamentals of ITK*

ITK image processing algorithms are implemented in classes which are generally templated (i.e. parameterized) with respect to the input and the output image types. For example, the `RecursiveGaussian ImageFilter<TInputImage, TOutputImage>` smoothes an image of type `TInputImage` and stores the results within an image of type `TOutputImage`. Most ITK images, `Image<PointType,Dim>`, are also defined through two template parameters: the intensity (`PointType`) and the spatial dimensionality (`Dim`). Thus, algorithm code becomes reusable whatever the medical image (2D mammograms, 3D CT scanner, multimodal images, etc.). It must be underlined that ITK also provides a pipeline mechanism allowing to connect algorithms in order to build sophisticated processing flows: it appears highly convenient to preserve this property allowing to easily combine image processing methods.

When applying ITK algorithms, the set of image points to be processed is managed by ITK iterators. An iterator is a special kind of object (a design pattern [8]) widely used in object-oriented programming to scan a dataset and access its elements. In our context, ITK iterators are used to access image points and are templated over the image type: the genericity of algorithms with respect to the image intensity and spatial dimensionality is then guaranteed. Among available iterators, the most widely used in algorithms are both variants of `ImageRegionIterator`and `NeighborhoodIterator`, and are limited to n D rectangular ROIs. Only the `NeighborhoodIterator` provides neighborhood access (i.e. allow accessing neighbours of a given image point, which is required for algorithms such as filtering). The main drawback of ITK, leading to its lack of genericity, is that algorithms are not parameterized with respect to the iterator type: for each algorithm, the iterator type is hard-coded, which is highly critical. Indeed, due to the explicit definition of the iterator type (i.e. at code level) in the core of algorithms, it is a priori impossible to modify the ROI type (rectangular region, arbitrary shaped one, etc.) without rewriting the code of the algorithm, which would be a tedious task due to the huge number of currently implemented algorithms.

To overcome this limitation, we propose to extend and parameterize both iterators with a new pattern. This extension, leading to ITK++, is hereafter described.

3.2. *Proposed improvement: extending iterators*

The design we propose consists of embedding an entity within an ITK-like iterator, this entity being more abstract than a simple rectangular region. This entity, which we propose to name *scanner*, will control the iterator movement over image points.

```
template < typename Image >
class Iterator
{
private :
    Image::ScannerType *scanner;
    Image::PointType *currentPosition;
public :
    void operator ++( )
    {
        int jump = scanner –> getNextJump ( );
        currentPosition += jump;
    }
}
```

Fig. 1 – Illustration of the proposed iterator extension in pseudo-code. The iterator manages image scanning: the private member `currentPosition` indicates the current image point under analysis. The proposed entity (the private member *scanner*) manages the jump to perform to access the next image point from the current one, by invoking the function `operator++()`.

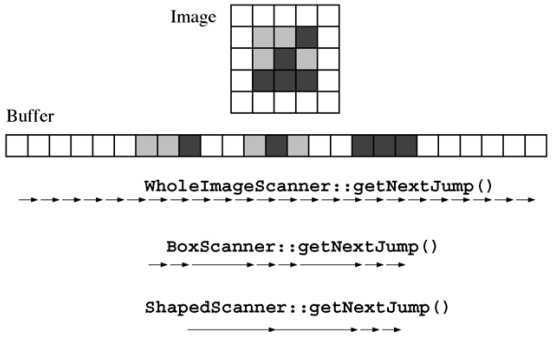

Fig. 2 – Illustration of the principle of the *scanner* entity. *Image*: the grey area is a classical square ROI and the black area represents an arbitrary shaped one. *Buffer*: typical image representation in computer memory. *WholeImageScanner*, *BoxScanner* and *ShapedScanner* represent *scanner* behaviour when respectively processing the whole image, the grey square ROI and the black arbitrary shaped one. Arrows represent memory jumps applied to the iterator for the various *scanners*. Arrow length reflects the number of image points to be skipped when going from one image point to the next one. The number of arrows represents the number of processed image points. This number clearly depends on the considered ROI (i.e. the whole image, the rectangular grey area or the black one).

For rectangular ROIs, this entity will be associated with a rectangular box. For arbitrary shaped ROIs, such as organs, this entity will store the list of image points to scan. Thus, when getting to the next point, the iterator will query the scanner to determine the required memory jump (or image point jump). Pseudo-code reported by Fig. 1 illustrates the use of the *scanner* embedded within an iterator. Fig. 2 graphically illustrates the principle of the proposed extension.

3.3. *Integration within ITK: ITK++*

In order to integrate the *scanner* within ITK, we propose to modify the iterator behaviour according to the image type, becoming itself parameterized by the scanner type. Technically, this involves the creation of a new image type (Section 3.3.1) and the specialization of some ITK iterators (Section 3.3.2).

3.3.1. *A new image type*

The first entity that is created is the class `ImageWithScanner<TImage,Tscanner>` which inherits from the native ITK `Image`, therefore involving minimal code writing. It can be seen as a classical ITK `Image` with an additionnal attribute: a *scanner*, the type of which being defined by the template parameter `TScanner`. Thanks to such inheritance, the `ImageWithScanner<TImage,Tscanner>` is ensured to be compatible with all filters designed for the native ITK `Image`.

3.3.2. *ITK iterator specialization*

The behaviour of the most widely used image iterators must be specialized (i.e. rewritten) for the new image type (i.e. `ImageWithScanner`), in order to let the underlying scanner control the iterator movement. The resulting iterators remain similar to the native ones (to ensure compatibility), except that their movement is then managed by the underlying scanner provided by the new image.

Fig. 3 illustrates how both new image type definition and iterator specialization can make native ITK algorithms generic with respect to the region of interest.

In order to preserve the ITK pipeline functionality, we need to take care of the information propagation when executing connected ITK algorithms (see [2]for details concerning the ITK pipeline). For example, in the case of a downsampling algorithm, the output user-defined requested region of interest must be appropriately upsampled to evaluate the buffer area which must be available at the pipeline input.

4. Evaluation of the proposed design in a practical use case

This section describes the evaluation of the extension of the ITK iterators. We first present the experimental protocol, the implementation of the scanning mode and then the results.

4.1. *Experimental protocol*

In order to evaluate to proposed design, we consider a use case dealing with the segmentation of liver vessels from a

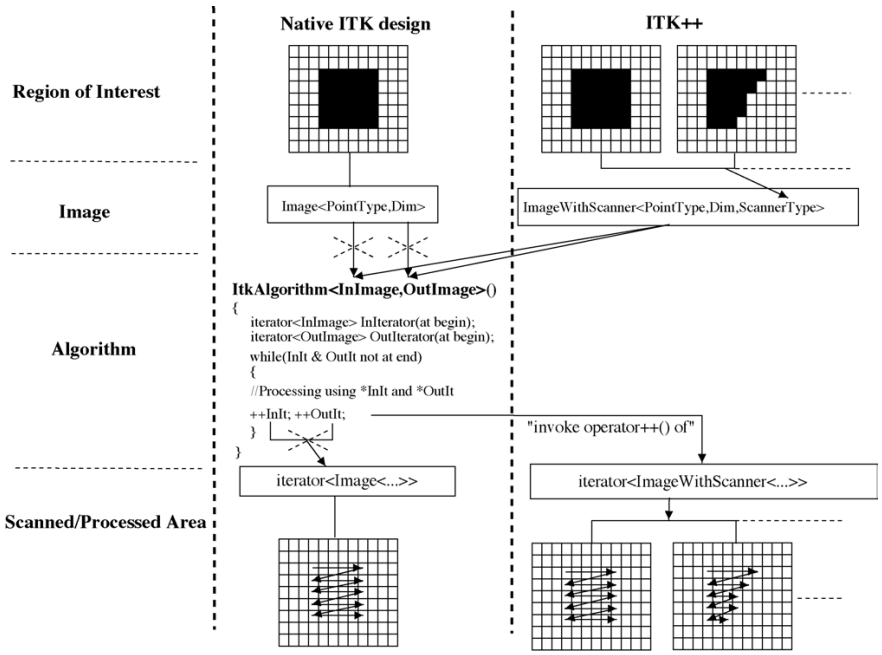

Fig. 3 – Integration of the iterator extension within ITK. ITK algorithms must be used with the new image type (i.e. `ImageWith Scanner< ... >`), which is generic with respect to the type of the integrated *scanner*: by rewriting the behaviour of underlying iterators, the scanner controls the movement. Computations are restricted to a region of interest. Template parameters `PointType` and `Dim` are respectively related to the image point type (e.g. scalar or multimodal images) and its spatial dimension (2D, 3D, 4D, etc.).

CT-scan image, which is an important application in medical image analysis [7]. This is required for surgical planning, where it appears convenient to localize possible liver tumors with respect to the vessel network. We assume that the liver has been previously segmented.

We consider a segmentation procedure based on a simple image smoothing followed by an intensity thresholding. This processing flow has been implemented using the pipeline mechanism provided by the native ITK framework. Three kinds of classical image smoothing algorithms have been considered: anisotropic diffusion, mean filtering and morphological dilation. This procedure, which is clearly too trivial to lead to an efficient and robust segmentation, is used for illustration purposes.

As the analysis is focussed on internal liver structures, the procedure should be applied only to voxels belonging to the arbitrary shaped area related to the liver, thus avoiding the unnecessary processing of external voxels. Using the proposed design, this can easily be performed. Fig. 4 gives a graphical representation of the code which has to be written using ITK++: a classical ITK pipeline is built and the region of interest is specified by a mask.

Using the native ITK design, the best way is to process all voxels belonging to the bounding box of the liver and then to intersect the result with the liver mask, as shown in Fig. 5. The native pipeline involves more operations than with the new design (although it is, of course, not so critical in this really trivial example): the bounding box must be computed and the result must be intersected with the initial mask. This proves that, from the computer scientist point of view, the proposed contribution can make procedure implementation easier.

In order to achieve a pertinent comparison, we propose to evaluate, in this particular use case, the relevance of our contribution by comparing the previously described procedure for two scanning modes. The first scanning mode concerns rectangular regions. This mode is native in ITK and has been implemented with our design for comparison purposes. The second mode we consider is related to the scanning of arbitrary shaped regions of interest. The implementation of these two modes in the ITK++ framework is shortly presented in the next section.

The size of the considered image is $512 \times 512 \times 80$ voxels. The bounding box related to the liver was a rectangular area of size $200 \times 197 \times 77 = 3\,003\,800$ voxels. It has been measured that the liver was defined by only 887 134 voxels, which is much less than the voxels belonging to its bounding box (about 3.4 times less).

The 1.8.0 release of ITK has been considered. A Linux based operating system with the gcc compiler version 3.3.4 has been used. Code source compilations have been done in release mode, with `-O3` option (highest optimization level). The computer used for experiments has an Intel Pentium 4 CPU (2.4 GHz) and 1 GB of RAM.

4.2. *Implementation of two scanning modes*

The entity which has been created for rectangular regions of interest is the class `BoxScanner`, and is generic with respect

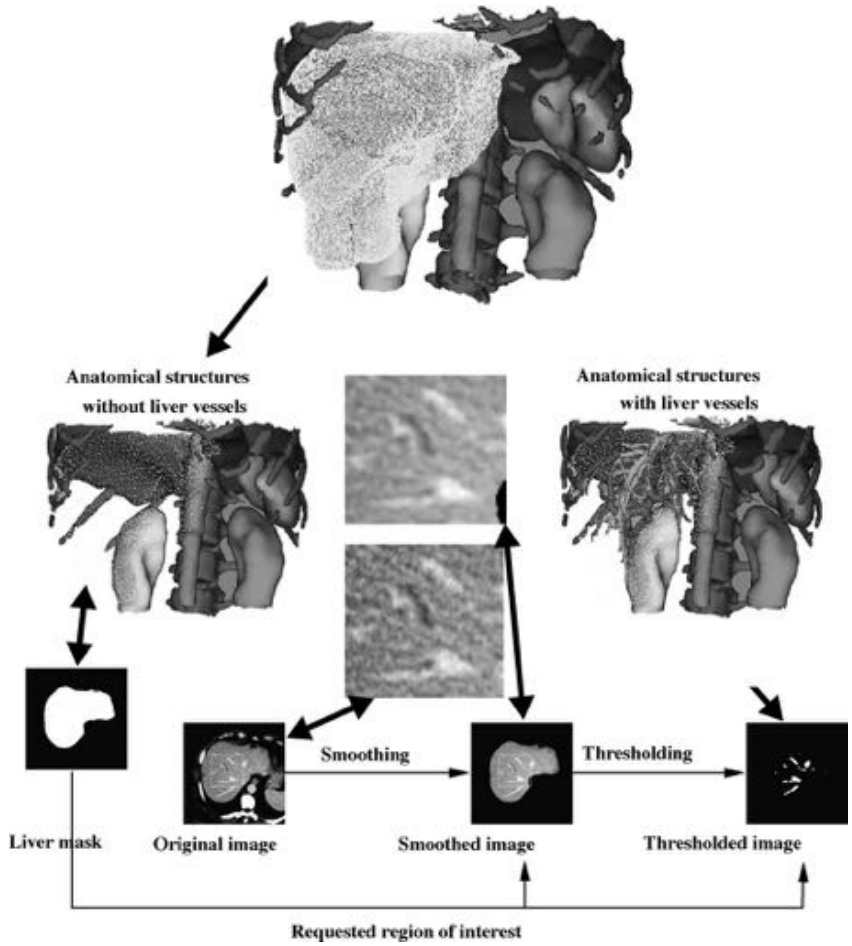

Fig. 4 – Example of processing flow with the proposed contribution (ITK++): detection of liver vessels assuming the liver is already segmented. The original image is smoothed and thresholded. Processings are constrained to the arbitrary shaped region of interest defined by the liver mask in order to optimize runtime and internal liver vessel detection.

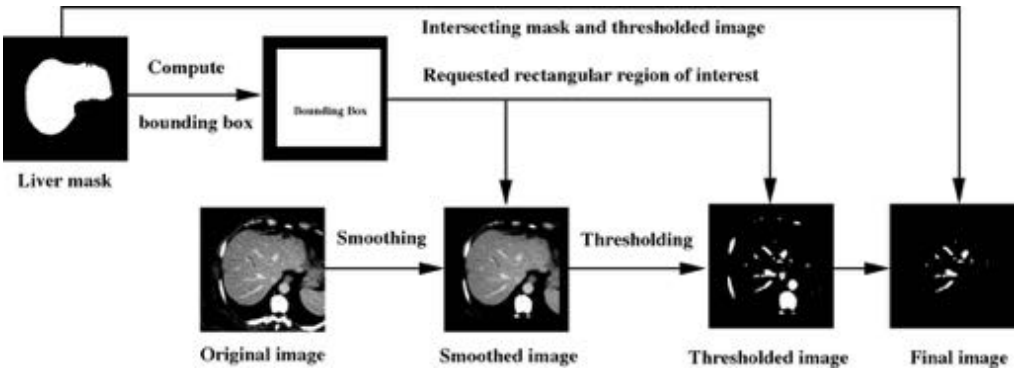

Fig. 5 – Example of processing flow with the native ITK: detection of liver vessels assuming the liver is already segmented. The original image is smoothed and thresholded. Processings can only be constrained to the bounding box of the liver, involving useless computations. By finally intersecting the result with the liver mask, internal liver vessels can be recovered.

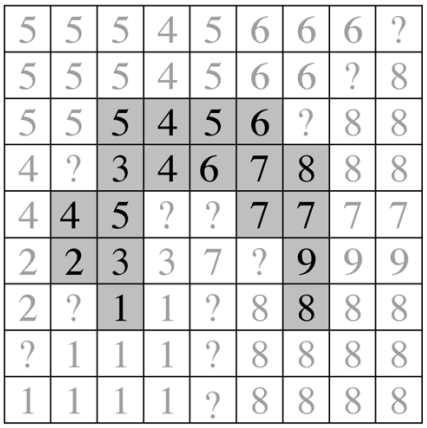

Fig. 6 – Illustration of the ambiguity of data extrapolation at boundaries of an arbitrary shaped region of interest (grey area). Digital values represent intensities and "?" ambiguous values. A classical extrapolation has been considered in this example: the intensity of a given outside point (white area) is the one of the closest image point belonging to the ROI. Extrapolation can become ambiguous for outside points having several closest image points belonging to the ROI (i.e. concavities).

to the spatial image dimension. Its core is an n-dimensional box (object `ImageRegion` of ITK). The core of the code comes from `Image(Const)RegionIterator(WithIndex)` (thus, the reader can refer to these native ITK classes for technical details).

The entity which has been created for arbitrary shaped regions is the class `ShapedScanner`, and is also generic with respect to the spatial image dimension. The core of this scanner lies in a vector containing the index of each point related to the associated shape (i.e. the liver in our case).

Extrapolation is required for processing image points at ROI boundaries with neighborhood based algorithms (i.e. the processing of an image point is based on its neighbours, some of them being outside the ROI). In the latter case of the `ShapedScanner`, extrapolation can be more complex to manage than in case of rectangular ROIs due to concavities. This is illustrated in Fig. 6. This aspect, which must be pointed out, is not the topic of that paper which focuses on the overall design for integrating the genericity within ITK, with respect to arbitrary shaped ROI. For this experiment, no specific extrapolation has been managed: the value of outside points is the one of the initial image. The drawback is that, after several processings, image points which are outside the scanned region are not valid anymore because they have not been transformed as image points belonging to the ROI. This can involve undesirable error propagation at ROI boundaries. Nevertheless, from a practical point of view, this can be overcome by considering a safety margin large enough with respect to the kernel size of the neighborhood operations to be applied.

Table 1 – Runtimes measured in seconds for the three different ITK based smoothing algorithms and for both the proposed design and the native ITK design

Smoothing algorithm	Anisotropic (s)	Dilation (s)	Mean (s)
Native ITK	65	7.23	7.88
ITK++ with `BoxScanner`	52	5.65	6.66
ITK++ with `ShapedScanner`	16	1.62	1.80

For comparison purposes, with our design, the processing of this rectangular area is based on the use of a `BoxScanner`. The `ShapedScanner` corresponds to the processing constrained to the liver area. Morphological dilation and mean filtering are performed using a neighborhood of size $5 \times 5 \times 5$.

4.3. *Experimental results*

Table 1 reports experimental runtimes we measured for the three different image smoothing algorithms previously enounced, within the previously defined processing flow. For comparison purposes, we first reported runtimes obtained when processing the bounding box of the liver area using both the native ITK design and our design (with the `BoxScanner`). We observe that our implementation leads to faster processings: this probably results from slight differences in code writing and compilation optimization. For this reason, it appears more relevant to illustrate the interest of our approach by comparing processings resulting from the same design (i.e. comparing runtimes obtained with both the `ShapedScanner` and the `BoxScanner`). It appears that computations can be clearly faster with arbitrary shaped ROIs. In this use case, a processing constrained to the liver area is about 3.5 faster (on average) than the processing of its bounding box. We point out that runtimes related to overall processing flow are not given because additional computations (i.e. image thresholding, offset computations from the liver mask, intersecting mask and thresholded image) have been evaluated to be negligible (less than 1 s) with respect to the slowest step (i.e. the image smoothing). Experimental results clearly prove the interest of our contribution.

5. Discussion

As illustrated by the experimental results, the use of arbitrary shaped ROIs can allow to spare non-negligible time (which increases with algorithm complexity and data volume), which can be helpful for fast patient modeling. Similarly, in robotically assisted surgery, it can be convenient to directly focus on data of interest (e.g. detection of specific markers) to register robot and real patient using medical data: computation time can be dramatically reduced, which is of great interest for heart beating and breathing compensation for instance.

It must be underlined that the technical implementation of a s canner is not unique. For example, in the case of arbitrary shaped regions of interest, another alternative could be to consider the shape to be scanned as an image mask: the iterator scans the image by skipping points which are outside the mask. The most efficient implementation depends on the shape and on the size of the area to scan: as the first method is memory consuming (one point can be associated to several

coordinates plus a memory jump), it should be preferred for small but widespread shapes, such as areas associated to the liver vascular system in medical image analysis. In such a case, the number of image points is relatively small with respect to the spatial extense of the anatomical structure. Moreover, the scan of irrelevant image points is avoided. Thanks to genericity, the proposed design can easily manage both implementations by just writing both behaviours and invoking the most relevant one. Thanks to the proposed framework, it is possible to design a specific scanner which will be automatically applicable to ITK algorithms.

Another point to discuss concerns the problem of data extrapolation, which has been shortly introduced previously although it is not the topic of this paper. In our sense, it would be interesting to achieve a specific study of possible extrapolation techniques and their impact on runtime and processing efficiency. A relevant solution could be to consider a distance transform based extrapolation: the value of ambiguous points is computed according to the distance with respect to their closest neighbours belonging to the ROI. Technically, this could be implemented at the iterator/scanner level, as it is already done in the native ITK for rectangular ROIs. In our sense, this additional work could be easily integrated within the proposed contribution.

The last point we propose to discuss concerns the limit of validity of our contribution: the proposed solution works only with ITK filters which do not hard-code neither the image type (differing from `ImageWithScanner`) nor an iterator type differing from those which have been specialized. When the first situation occurs, as it is for example the case with the ITK anisotropic diffusion we used, the filter must be adapted by specializing it for our proposed new image type (however, this is not a tedious task). When the second situation occurs (we observed that it is rarely the case), the iterator must be specialized to support the new image type.

In our opinion, the proposed solution has the advantage of allowing to further evaluate the practical interest of such genericity while preserving the native behaviour of ITK (i.e. both versions are available). Furthermore, from a practical point of view, it minimizes add-ons and does not involve any modification of the native library, thus ensuring limited bugs and a stable behaviour of the native version.

6. Conclusion

In this paper, we proposed a solution to improve the genericity of the powerful tool ITK in order to manage regions of interest. Therefore, we defined a new entity, that we named *scanner*, dedicated to the parameterization of the set of image points to be processed when applying an algorithm to a medical image. We implemented this design within ITK and proved the interest and potential of our approach: processings can be restricted to any user-defined ROI without wasting time by considering unwanted parts of images. As underlined, this makes processings faster and more efficient, which is becoming nowadays crucial for many medical systems and especially those dealing with real-time processing (e.g. robot guidance, heart beat compensation, etc.). This work would be useful for medical image processing software/system users and developers in terms of runtime performances and segmentation accuracy. It must be pointed out that there are still some filters which must be adapted to support our add-ons. The next step is to validate our approach on more algorithms and complex processing flows, and to evaluate the impact (in terms of runtime) of extrapolators in the case of arbitrary shaped regions of interest.

Acknowledgements

First of all, we are very grateful to IRCADs computer science and robotics research and development team, and in particular G. Brocker, J. Moreau, N. Papier and S. Thery for their pertinent remarks concerning programming techniques and methodology. We would also like to thank the French National League against Cancer, the Alsace Region, the French Research Ministry and the French Agency for Innovation for their financial support. Finally, we would like to thank all our medical partners, who have provided us medical images of patients, and in particular Pr. Afshin Gangi and Pr. Catherine Roy from the radiological department of the Civil Hospital in Strasbourg. We also thank the surgical team of Professor Marescaux for their help in our works, in particular Pr. Mutter, Pr. Leroy, Dr. Vix.

REFERENCES

[1] U. Köthe, in: B. Jähne, H. Haussecker, P. Geissler (Eds.), Handbook of Computer Vision and Applications, vol. 3, Academic Press, 1999, pp. 103–132 (Ch. Reusable Software in Computer Vision).
[2] NLM, National Library of Medicine: insight segmentation and registration toolkit, http://www.itk.org.
[3] J.-B. Fasquel, V. Agnus, Improving genericity and performances of medical systems and softwares based on image analysis, in: Proceedings of the 18th IEEE International Symposium on Computer-based Medical Systems, 2005, pp. 247–252.
[4] J.-B. Fasquel, M. Bruynooghe, A new hybrid opto-electronic method for fast and unsupervised object detection, Opt. Eng. 42 (11) (2003) 3352–3364.
[5] K. Castleman, Digital Image Processing, Prentice-Hall, 1996.
[6] J. Weickert, Efficient image segmentation using partial differential equations and morphology, Pattern Recogn. 34 (2001) 1813–1824.
[7] D. Selle, B. Preim, A. Schenk, H.-O. Peitgen, Analysis of vasculature for liver surgical planning, IEEE Trans. Med. Imaging 21 (11) (2002) 1344–1357.
[8] E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns, 1st ed., Addison-Wesley, 1995.
[9] K. Czarnecki, U. Eisenecker, Generative Programming: Methods, Tools and Applications, Addison-Wesley, 2000.
[10] D. Vandevoorde, N. Josuttis, C++ Templates: The Complete Guide, Addison-Wesley, 2003.

COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE 83 (2006) 222–233

journal homepage: www.intl.elsevierhealth.com/journals/cmpb

A modular and evolutive component oriented software architecture for patient modeling

Jean-Baptiste Fasquel, Guillaume Brocker, Johan Moreau, Nicolas Papier, Vincent Agnus, Christophe Koehl, Luc Soler, Jacques Marescaux*

IRCAD, 1 place de l'Hôpital, F-67091 Strasbourg, France

ARTICLE INFO

Article history:
Received 10 January 2006
Received in revised form
29 June 2006
Accepted 6 July 2006

Keywords:
Software architecture
Reusable components
XML
Medical image processing

ABSTRACT

This paper deals with the design aspect of a software aiming at modeling the anatomical and pathological structures of patients from medical images, for diagnosis purposes. In terms of functionalities, it allows to combine image processing algorithms, and to visualize and manipulate 3D models and images. The proposed software uses specific extensible and reusable components and a system managing their combination, thanks to a formal XML-based description of their interfaces. This architecture facilitates the dynamic integration of new functionalities, in particular in terms of image processing algorithms. We describe the structural and behavioral aspects of the proposed reusable component-based architecture. We also discuss the potential of this work for developing other softwares in the field of computer aided surgery.

1. Introduction

In the field of computer aided surgery, current research activity involves the development of an increasing number of softwares for different specific applications: computer aided diagnosis [1,2], surgery planning [3], simulation [4], image guided surgical intervention [5]. Most of these applications share several common functionalities such as image analysis and interactive 3D visualization for instance. Indeed, image analysis is required to model medical structures for diagnosis purposes. It can also be needed for modern and future systems dedicated to image guided intervention [5–7], where it is necessary to detect and analyze, during the intervention, medical information to control the behavior of the system. Three-dimensional visualization is required for patient modeling, to control segmentation process for instance, and for both surgery planning and simulation, where it appears necessary to interact with models using virtual tools [3], sometimes also incorporating force feedback [4]. Moreover, due to the important research activity related to this emerging field, softwares must be modular and evolutive enough to easily support the integration of new functionalities. For instance, in the case of patient modeling on which we focus in this paper, it is crucial to easily incorporate or modify image processing algorithms.

In this context, when developing softwares for such applications, it appears relevant to consider a design methodology facilitating the reusability of the code and the extension of functionalities. Development effort, including refactoring [8] and code maintaining, can be therefore minimized and facilitated.

Component oriented programming [9] is an interesting and useful approach for managing this aspect [10–12]. According to Spzyperski [9], a software component is "*a unit of composition*

* *Corresponding author*. Tel.: +33 3 88 11 90 79; fax: +33 3 88 11 90 99.
E-mail address: Jean-Baptiste.Fasquel@ircad.u-strasbg.fr (J.-B. Fasquel).
URL: http://www.ircad.org.

0169-2607/$ – see front matter © 2006 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.cmpb.2006.07.002

with specified interfaces and describe context dependencies only. A software component can be deployed independantly and is subject to composition by third parties." By combining components, one extends their functionalities, therefore leading to the high-level software ones. By standardizing and limiting dependencies between components (restricted to few explicit interfaces), one intends to facilitate refactoring, correction, maintaining and evolution because modifications are localized (e.g. to the code of a given component) and their impact on the rest of the system can be better foreseen and controled. This programming approach is well known by the community dedicated to software engineering and can be, in our sense, advantageously applied to image processing context.

We developed a cross-platform (Windows, Linux and Mac Os X operating systems) component oriented software, based on the C++ language, for medical image segmentation. In our case, the advantage of this approach essentially concerns the extension of software functionalities in terms of image segmentation algorithms: they are embedded within operator components (similar to "*plugins*") which can be easily and dynamically integrated in the software without modifying its code. As previously said, this is highly convenient in this highly evolutive field to facilitate the improvement of existing algorithms and the rapid integration and evaluation of new ones, with minimal programming, as recently underlined by Papajorgji [13] for another application field. Moreover, the use of additional components (dedicated to the graphical user interface, to the rendering of 3D data and to the management of operators for instance) makes our software system highly modular. This is expected to strongly facilitate the future development of additional softwares (associated to some of the above cited surgical applications), by reusing the methodology and most components, with minimal refactoring.

A component-based software involves the definition of suitable and reusable components, but also the management of their combination and configuration. As recently underlined in [14], many software systems suffer from missing support for this aspect. This is often hard-coded, leading to tangled composition and configuration code that is hard to understand and maintain. As in [14], we propose to separate this aspect from the rest of the software. We developed a component management system, based on a XML-based formal description of component interfaces (including XML Schema specifications [15]), the role of which being to define how software components are plugged together (or contribute to each other) and to start effective combinations.

The description of the software architecture we give in this paper, involving the above listed modern aspects related to software engineering, is expected to facilitate the work of researchers and computer scientists developing new systems in this emerging field dealing with computer aided surgery, so that they can more easily integrate their work. In our sense, the originality of this paper essentially relies on the use, in our medical context, of the software component approach as well as of the formal XML-based component management system.

The next section is dedicated to the functional description of the software, to facilitate the understanding of the role of the different underlying components. Then, we shortly describe the different components we developed to build the final application. In Section 4, we describe their structure, and how both interfaces and contributions are specified and effectively implemented. In Section 5, a behavioral description of our software is provided. In particular, we detail how components are plugged together when the application starts. We also shortly describe the communication system. Most explanations will focus on image processing which is the central aspect of the considered software dedicated to patient modeling. Before concluding, we discuss about this work and its perspectives in terms of application (extension to additional softwares for computer aided surgery) and software engineering.

2. Functional description of the proposed software

In this section, we shortly describe the standard manipulation mode of our software, and the major functionalities related to image processing.

2.1. *The standard manipulation mode*

The standard manipulation is described in a recently published work focusing on a specific functionality of this software [16]. Therefore, only main guidelines are reported hereafter. Fig. 1 reports a snapshot of the developed software. Fig. 2-right represents the effective image processing graph (or processing line) corresponding to the simplified representation provided to the user (Fig. 2-left). Such a two-level hierarchical representation has been preferred to a traditional one implicitely displaying operator connections (such as in MevisLab [17] or Amira [18] for instance) because, in our opinion, it makes the visualization of the modeling process clearer. The representation is similar to the one of a file system where the first level, associated to processing sequences (e.g. segmenting organs, tumors), is related to pipelines of operators which are listed at the second level.

Typically, the user starts a new modeling procedure by loading and preprocessing (e.g. noise removal, rotation) the 3D medical image representing the anatomical and pathological structures of a given patient. For each new sequence, the user first selects the reference sequence (in frame F, Fig. 1), being usually the first one (dedicated to noise removal for instance, and corresponding to "preprocessing" in Fig. 2). This could be another sequence such as "air" (Fig. 2), from which both lungs could be further segmented [16]. Then, he presses the "*new sequence*" button of the software toolbar, involving the insertion of a new entry in the graphical representation, frame F in Fig. 1, of the image processing graph (corresponding to a new empty pipeline of operators in the effective graph reported in Fig. 2-right). This creates the appropriate connection in the effective image processing graph, between the user selected reference sequence and the new one.

The user can also define the sequence nature (properties editor, frame C in Fig. 1). As detailed in [16], the nature can either be a "target", for segmentation purpose (e.g. liver or heart), or a "zone", the goal being therefore to preprocess data so as to facilitate the further segmentation of targets (e.g. extracting air from which both lungs can be further segmented). This is required so that our system automatically

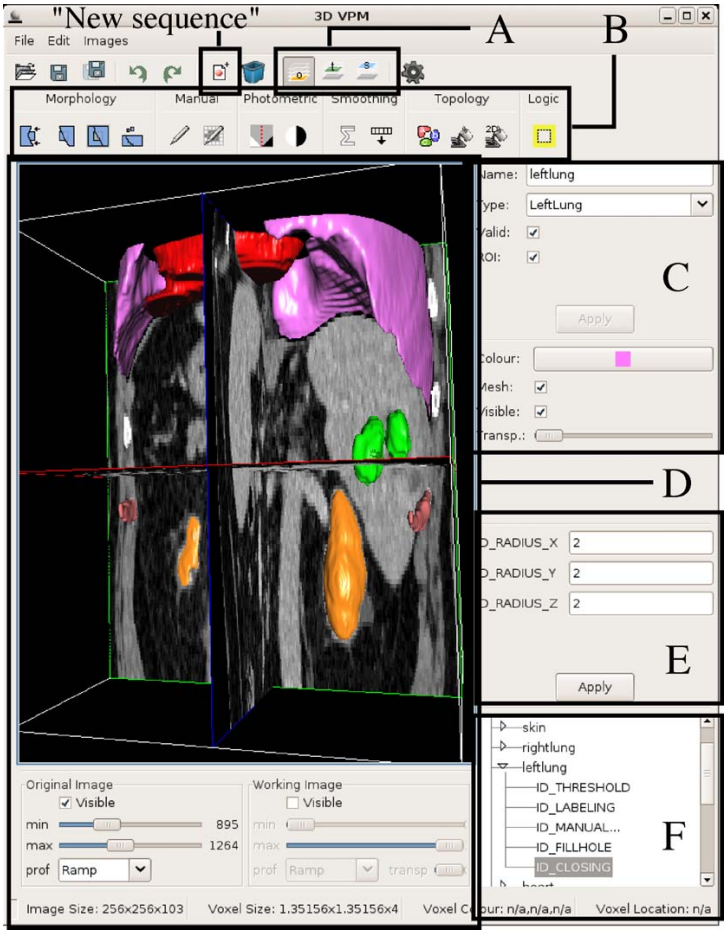

Fig. 1 – Snapshot of the developed patient modeling software. The part surrounded by frame B lists all available image processing operators which can be dynamically created and inserted within the image processing graph surrounded by frame F. This frame allows to visualize the processing line and to select a sequence or an operator. Frame E corresponds to the parameter editor of the operator selected in frame F. Frames A and D surround functionalities related to the visualization of the data associated with the current selection in the processing line (e.g. modifying the orientation of the negatoscope and the displayed slices). Frame C is dedicated to structure properties (type, name, color, etc.).

determines within which region of interest computations can be restricted, according to already segmented medical structures. For instance, if the liver has already been modeled and if the current structure type is a liver tumor, the system automatically determines that processings, for extracting the liver tumor, should be constrained within the liver area, which can strongly facilitate and accelerate segmentations.

After having defined the sequence nature and type, the user can insert (or remove) image processing operators from those available (icons within the frame B of Fig. 1), to build a processing sequence (OPi in Fig. 2-right). Data which is displayed (see frame D in Fig. 1) corresponds to the selected element (within frame F of Fig. 1). This allows to visually control the evolution of the processings. Finally, in case of "target", the user can validate the sequence (properties editor, frame C in Fig. 1) when he considers that the segmentation is finished. This allows, for instance, to generate its 3D surface representation (which can be superimposed on a 3D representation of the negatoscope, as illustrated in Fig. 1) and to register it within a database from which regions of interest associated with next structures will be determined [16].

The user can then start additional sequences, until all structures of interest are segmented. It must be pointed out that the patient model can be saved as a file to be later used for diagnosis, surgical planning or simulation purposes. This file also incorporates a description of the image processing graph interactively built during the modeling procedure.

2.2. *Major image processing functionalities*

Concerning image processing, two kinds of functionalities have been implemented: high-level ones, close to our medical application and context, and low-level ones, which are less specific. We point out this distinction because it

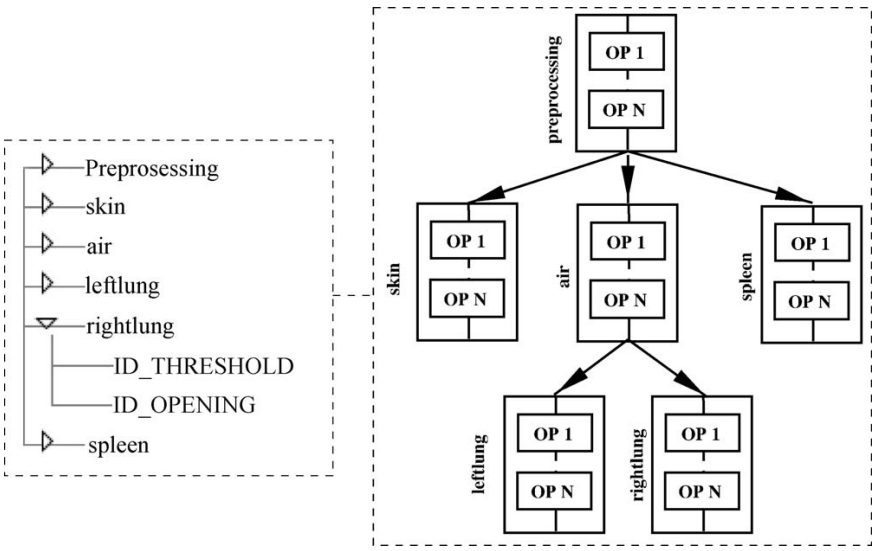

Fig. 2 – Effective image processing graph and graphical representation. (Left) Graphical representation (the one reported in frame F of Fig. 1). (Right) Associated effective image processing graph, describing the relationship between pipelines and underlying operators OPi.

has an impact on the definition of some of our components, as it will clearly appear in the next sections of this paper.

The first one concerns high-level implicit operator sequence construction and parameterization, according to user actions and medical information. Construction corresponds to the high-level mechanism translating user actions applied on the simplified graphical representation of the processing graph (Fig. 2-left) into the one effectively describing operator connections (Fig. 2-right). Parameterization concerns the determination of the regions of interest according to medical structure types defined by the user [16]. The second kind of image processing functionality deals with the smart mechanism implemented at low-level so that computations are correctly synchronized and optimized. It mainly concerns the automatic updating mechanism (see Fig. 3), similar to the one proposed in ITK [19] or VTK [20].

3. Structural description of the proposed software

We shortly describe the role of the different components, reported in Fig. 4, which are involved in our software.

The *ui* component represents the graphical user interface. This component is intended to manage the set of graphical user interfaces provided by the other components.

The *document* component is the central part of our software, as it will be further underlined in Section 5.2. It essentially maintains medical structures and an abstract description of the image processing graph built to achieve segmentations. Previously cited high-level implicit mechanisms (corresponding to high-level functionalities reported in Section 2.2) related to image processing are defined and managed by this component.

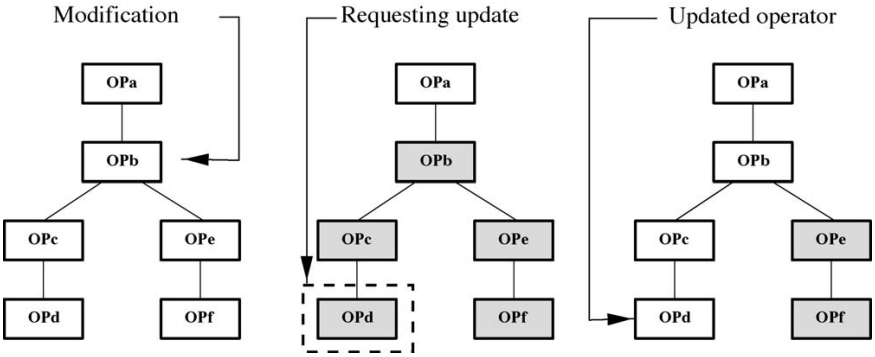

Fig. 3 – Updating mechanism on a processing graph. The modification of OPb (left graph) invalidates operators in grey boxes in the central graph. Requesting the update of OPd (surrounded by a dashed frame) involves minimal computations (only OPb, OPc and OPd of the right graph are automatically updated, OPe and OPf remaining invalid).

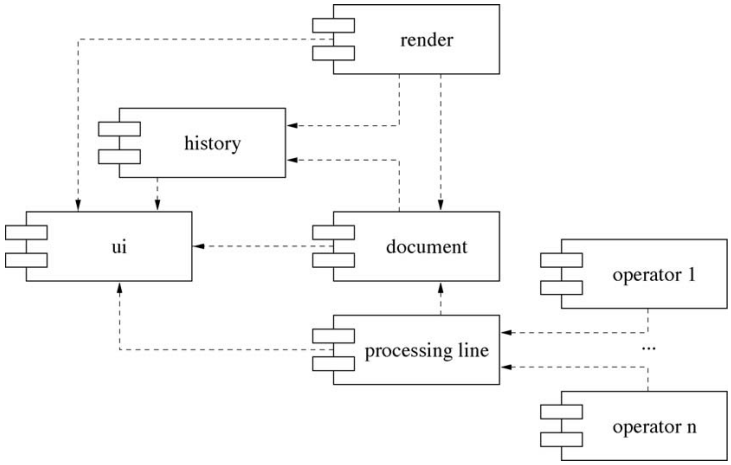

Fig. 4 – The different components of our software, including dependencies.

The *render* component manages everything concerning visualization of the data centralized by the *document* component. It provides functionalities for configuring the view mode (e.g. 2D or 3D negatoscope representation, number of simultaneous views, intensity windowing) and manipulating rendered objects (e.g. zoom, translation and rotation of the 3D objects).

The *history* component is used to store the sequence of user actions. This allows to undo and redo them, which is a highly convenient functionality from the user point of view. This component is not further detailed because it is not central enough in our system.

The *processing line* component encapsulates the image processing graph of operators. This graph is the implementation of the abstract description maintained in the *document* component. It synchronizes computations of operators and delivers processed and updated data to the *document* component. It also manages the low-level image processing mechanism that we shortly described in the previous Section 2.2.

To each operator is associated a specific *operator* component which provides, to the *processing line*, the core of the considered image processing algorithm. It can also embed its own graphical interface for entering parameters and is adapted to parameter specificity.

4. Structural description of components and contributions

We first propose to describe component content. Then the formal XML-based description of interfaces and contributions is given. Contributions concern either services or resources that components can provide to other ones, through interfaces, when they are combined. This allows to build the high-level software functionalities which have been previously described in Section 2. In the last part of this section, we focus on contributions related to services (i.e. effective code), achieved thanks to a specific design pattern [21].

4.1. *Component content*

A component is a bundle containing a XML file, a dynamically loadable library and resources:

- The XML file, which is further detailed in the next Section 4.2, describes component content and contributions.
- The dynamically loadable library implements services, assimilated to functionalities, which can be used to extend those of other components, through contributions. For instance, in the case of *operator* components, the library provides the image processing algorithm. Similarly, as illustrated by Fig. 5, the library contained in the *render* component contributes to the *ui* component one by providing functionalities to the main graphical interface (parts surrounded by frames A and C in Fig. 1). This graphical interface allows to manipulate the visualization scene managed by the *render* component. The technique used to implement such services is described in Section 4.3.
- Resources correspond to a set of files (related to icons, documentation, etc.) that a component can provide to the system. For instance, each *operator* component can embed a bitmap image to be incorporated within the graphical interface of the system (icons within the part surrounded by frame B in Fig. 1). The parameter list and default values of the *operator* component are also provided through textual information. Similarly, the *document* component provides a textual description of organ properties (i.e. type, color, etc.).

4.2. *A formal XML-based description of interfaces and contributions*

As previously said, the XML file describes contributions and interfaces the component provides. Contributions are formalized by specific a XML element that we called *extension*: the corresponding component proposes to extend or contribute to specific interfaces (associated to a XML element we called *extension point*) provided by other components.

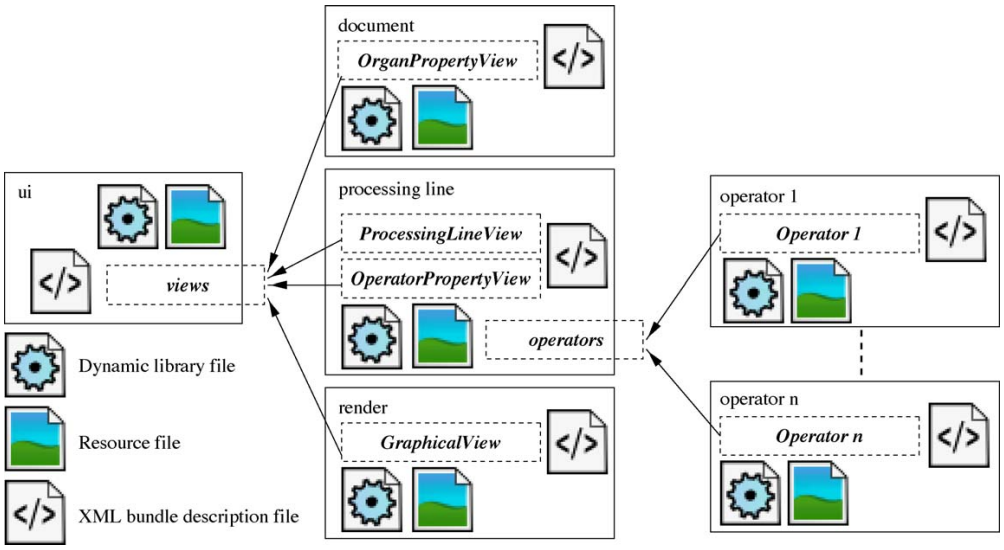

Fig. 5 – Graphical representation, associated to a formal XML-based description, of the contribution relationship between the various components (also called bundles) of the architecture. *Render*, *document* and *processing line* components provide graphical user interface contributions to the *ui* component thanks to *extensions* matching the *views extension point*. An extensible number of *operator* components (components *operator* 1 to *operator n*) can contribute to the *processing line* component through another kind of *extension point* differing from the one proposed by the *ui* component.

To facilitate the understanding and for illustration purposes, we propose to describe both notions of *extension* and *extension point* in a particular case associated to the contribution of a given *operator* to the *processing line*. This example is of great interest because it concerns the extension of system functionalities in terms of image processing operators, which is essential because the software is dedicated to medical image segmentation. This can be assimilated to a formal description of traditional plugins, similar to those described by Papajorgji [13] who directly focused on their technical implementation for another application field, without considering the proposed formal description.

The *processing line* component declares its intention to use external image processing operators. Thus, it defines an *extension point* (see Fig. 6), called `pl::operators` in our case, and a schema (see Fig. 7).

This schema, bundled to the *processing line* component as a resource file, follows the XML Schema specification [15] and imposes that an operator must provide, for instance, a human-readable name as well as an icon. The parameter list ("parameter-list" in Fig. 7), being optional ("minOccures = 0"), can contain an unbounded number of parameters ("maxOccures = unbounded"), each one having an identifier as well as a default value (could be extended to an acceptation range for instance). To contribute to this extension point, the given *operator* component declares an *extension* (see Fig. 8), which contains the required XML elements `icon` and `name`, as imposed by the schema file of the *processing line* component (verification being achieved when contributions are started by our *runtime* system, described in Section 5.1). Moreover, it provides an icon (which is a resource) and a dynamic library (associated to the XML element `library name`), which are bundled in the given *operator* component. The `class` XML attribute is related to a reference class type used to effectively incorporate the code embedded with the library, as detailed in Section 4.3. In our example, the given *operator* has one parameter (identified as "threshold" in Fig. 8. Parameters, initially considered as strings, are interpreted (appropriate types such as float or integer) by the instantiated operator at assignation (thanks to the identifier). Data and parameter management is further detailed in Section 5.2.

From this example, we clearly understand that an *extension point* is the formal declaration of an interface proposed by a component to allow others to extend its features. It can be seen as a contract establishing the requirements to contribute to the *extension point* (the schema). An *extension* must fulfill the contract defined by the corresponding *extension point*.

```
<plugin id="pl">
   <library name="pl"/>
   <extension-point id="pl::operators" schema="pl.operators.xsd"/>
</plugin>
```

Fig. 6 – An extension point declaration. *pl* is the short cut of *processing line*. This represents a part of the XML description file of the *processing line* component (or `plugin`). This file provides the library name (associated to the XML element `library name`) of this component and an *extension point*.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
...
 <xsd:element name="extension">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="icon" type="xsd:string" />
      <xsd:element name="name" type="xsd:string" />
      <xsd:element name="parameter-list" minOccurs="0">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="parameter" maxOccurs="unbounded">
              <xsd:complexType>
                <xsd:attribute name="id" type="xsd:string" use="required"/>
                <xsd:attribute name="default" type="xsd:string" use="required"/>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:all>
    <xsd:attribute name="point" type="xsd:string" use="required"/>
    <xsd:attribute name="id" type="xsd:string" use="required"/>
  </xsd:complexType>
 </xsd:element>
...
</xsd:schema>
```

Fig. 7 – Part of the XML Schema related to the `pl::operators` *extension-point* (file `pl.operators.xsd` considered in Fig. 6).

```
<plugin id="threshold">
  <library name="threshold"/>
  <extension point="pl::operators" class="Threshold">
    <icon>icon.png</icon>
    <name>Threshold</name>
    <parameter-list>
      <parameter id="threshold" default="0"/>
    </parameter-list>
  </extention>
</plugin>
```

Fig. 8 – An extension declaration (validated by the schema reported in Fig. 7).

It facilitates the management of component combination and configuration, as underlined in [14]. Indeed, the visualization of effective contributions becomes really clearer than when they are hard-coded, therefore facilitating the maintaining and the evolution of the structuration of the software. Moreover, contributions can be easily configured. For instance, as it will be detailed in Section 5.1, thanks to such textual descriptions, it becomes possible to easily define the set of components and connections we wish when starting the application (therefore configuring the set of high-level functionalities). The proposed software could therefore be easily converted into a single medical image viewer by disabling components related to image processing (ignoring contributions associated with image processing). Similarly, *extensions* and *extension points* can be easily configured. For instance, resources such as default parameter values of operators could be easily adapted, by simply configuring the XML description file.

4.3. *Service*

If contributing to an *extension-point*, a component can provide a library implementing a specified interface, finally leading to high-level software functionalities. Technically, this is managed by the use of the *Strategy* Design Pattern [21], as illustrated by Fig. 9 in the case of two particular contributions.

At build-level [22] (i.e. when compiling the source code leading to the library), the library only includes the abstract class (e.g. `IOperator`), but does not need to be linked with the library associated to *extension-point* (e.g. `pl::operators`). This ensures limited dependencies, only restricted to interfaces.

Both Figs. 5 and 9 report a simplified representation of the contribution network (respectively in a formal manner and in terms of effective service implementation) used to combine our components so that high-level functionalities and information are provided.

5. Behavioral description of the proposed software

In this section, we first describe how contributions are initialized, leading to high-level software functionalities. In particular, we explain how services related *operator* components are effectively started. Then, we shortly describe the technique used to manage communication between components, focusing on the *document* and *processing line* ones.

5.1. *Effective contributions: a formal XML-based management system*

Effective contributions are achieved by our component management system [14], which we called *runtime*, dedicated to the combination and the configuration of our components according to *extension points* and *extensions* they declare. It is a library using third party ones such as *libxml2* [23].

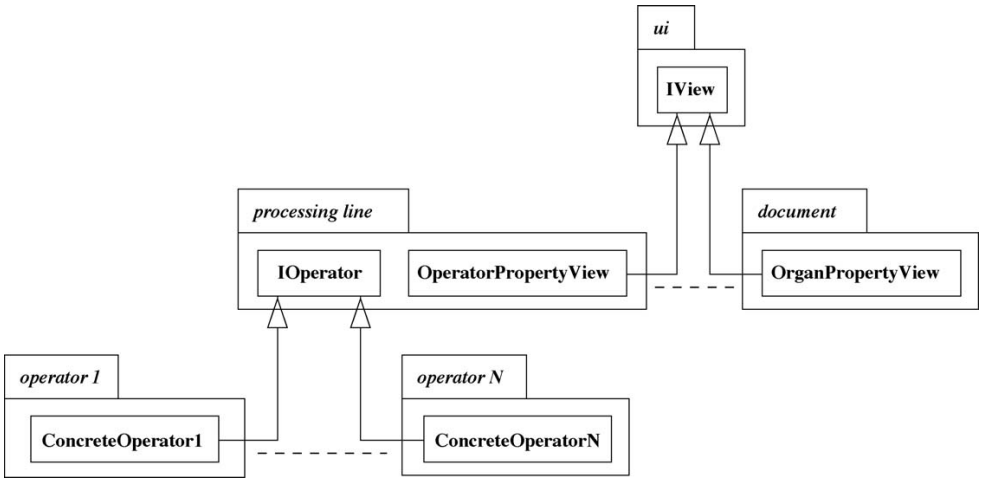

Fig. 9 – Simplified UML-like representation of the *Strategy* Design Pattern used to implement effective contribution, related to services, between components. `IOperator` and `IView` are abstract classes corresponding to component interfaces (related to *extension-points* `pl::operators` and `ui::views`). Inherited classes are concrete implementations extending component functionalities.

Fig. 10 shows the code invoked when the application starts. As we can see, *runtime* is asked to browse a given directory where components might be located (`m_bundlePath` in Fig. 10). Each time a component is found, it parses the description file and updates a knowlegde base of component *extension points* and *extensions*.

The *document* component is started first because next components (*processing line*, *render*, etc.) observe it (see next Section 5.2 for details).

Application's user interface component *ui* is started at last, because all the previous ones contribute to it. In this case, the *runtime* retrieves all *extensions* (declared by previously started components) associated to the `view` *extension-point* declared by *ui* and starts contributions (in terms of both services and resources), therefore providing views, toolbar buttons, menus, icons, etc., to *ui*.

Concerning *operator* components, the behavior slightly differs because contributions related to the code of algorithms are not started at application initialization. Indeed, the instantiation of a given operator is required only if the user explicitly inserts it when processing images. For these components, initialization only concerns resources such as the insertion of buttons (see Fig. 1, frame B), thanks to appropriate contributions to the *ui* component: the *processing line* queries *runtime* to retrieve all *extensions* contributing to its `pl::operators` *extension point*. Then, the *processing line* only uses the `icon` and `name` elements of each operator bundle (declared in *extensions*, as reported in 8) to create a button in the user interface. It will also associate each *extension*, in terms of service, to the relevant button for later use.

```
bool App::OnInit()
{
  ...
  runtime::loadBundles(m_bundlePath);
  runtime::startBundle("document");
  runtime::startBundle("pl");
  runtime::startBundle("render");
  runtime::startBundle("ui");
  ...
}
```

Fig. 10 – C++ code starting bundles. *pl* is the short cut of *processing line*.

After this initialization step, the application's graphical user interface is shown to the user who can start using the software (e.g. performing a patient modeling from an initial image, as described in Section 2).

Thanks to this *runtime* system, any new component is automatically detected and incorporated in our system, without describing it neither at code level nor in a configuration file. For instance, any new *operator* component located in the appropriate directory (browsed by the *runtime* system) becomes automatically available for the user in the toolbar of the application. This avoids the maintaining of a list of the available components. Indeed, the *processing line* component does not require to know the list of *operator* components which can contribute to it. As it has been partially discussed in the previous Section 4.2, our *runtime* system could be easily configured to filter contributions to start. For instance, if it is prefered to start the application in a basic viewer mode, contributions to `pl::operators` (see *extension* example 8) could be disabled by providing a simple XML configuration file to the *runtime*.

5.2. *Communication between components*

In this section, we propose to shortly describe how the communication between components is managed. When the application runs, most components communicate through an observation mechanism centered on the *document* component. Technically, the *Observer* Design Pattern is used [21]: the *doc-*

ument component is the subject and the other components observe it. In practice, most user actions are applied to *document* objects which are subjects that send notification messages. These messages can be received by objects (observers) belonging to the other components observing these *document* subjects. Then, these observers perform appropriate operations with respect to message nature (i.e. according to the initial user action). Therefore, the *document* component plays the role of a manager with respect to *render* and *processing line* components. Such a mechanism requires a preliminary registration step of components (initial attachment).

For illustration purposes, we propose to detail this communication mechanism in the particular case of the *processing line* component observing the *document* one.

As illustrated by Fig. 11, in the case of the *processing line* component, initial attachment requires the creation of a `Processing Manager` object which is then attached to the `Modeling Manager`. Both `Processing Manager` and `Modeling Manager` are singletons (*Singleton* Design Pattern [21], only one instance in the system). Singleton instantiation and attachment (if required) are managed by components encapsulating them, by executing an initialization code they embed. This execution is triggered by our *runtime* system (`startBundle()` in Fig. 10). This is the reason why the *document* component must be started before those observing it (in particular *processing line*, as illustrated by Fig. 10).

When the user starts modeling a set of medical structures from an initial image, it involves the creation of a `Modeling Procedure` object in the *document* component. The `Modeling Manager` notifies this user action (i.e. corresponding to the starting of a new modeling procedure). The `Processing Manager` receives the related signal and creates an `Effective Pipeline` which is automatically attached to the `Modeling Procedure` (observation *O1* in Fig. 11). The process is iterated according to user actions directly affecting *document* thanks to the *ui* component with which the user interacts (e.g. starting a new sequence or inserting a new operator). This is automatically notified to the *processing line* component in order to update the effective image processing graph.

Practically, each sequence (e.g. `Sequence 1`, `Sequence 2` and `Sequence 3` in Fig. 11) of *document* encapsulates a pipeline of operators as well as additional notions such as the sequence type shortly described in Section 2 (i.e. either a "target" such as `Organ 1`, `Organ 2` or a "zone" such as `Preprocessing`). When the user starts a new sequence, this involves the creation of a pipeline in *document* to be observed by its dual in *processing line* (e.g. observation *O2* in Fig. 11). Operator insertion (i.e. in a pipeline, associated with a given sequence, of *document*) is notified, thanks to this observation *O2*, to its dual in *processing line*, leading to the observation *O3* in Fig. 11. This also involves the instantiation of the effective image processing operator (according to the mechanism presented in Section 4.3), as well as its appropriate incorporation within the image processing graph.

Hereafter, we shortly describe how data and operator parameters are managed by our architecture and how the execution of an inserted operator is launched. When a parameter of an inserted operator is modified, a specific notification is sent by the considered operator of the *document* and received by its dual in *processing line* (e.g. observation *O3* in Fig. 11). This dual operator assignes the parameter to the effective algorithm provided by the appropriate *operator* component, which translates it into the appropriate type (according to the operator specificity), checks its validity (this can consist in a comparison with a range of acceptation) and raises an exception if relevant. The operator execution can also be requested using this communication mechanism. Input data is retrieved according to operator connections (i.e. the input data being the output one of the previous operator), and computations are started. Finally, the result (output data) is assigned by the dual operator (i.e. in *processing line*) to the one it observes in *document*. We use our own data structures (e.g. images supporting various point types) which are shared between our different components to avoid useless buffer copies. They are also shared according to image processing connections, similarly to the design considered in ITK [19] (i.e. the input port of an operator points to the appropriate output port of another operator, according to connections).

The major advantage of this observation based approach is that both image processing graphs (corresponding to the "high-level" and "low-level" functionalities reported in Section 2.2) are synchronized without building a too large component which would incorporate both of them. This avoids polluting the core component (*document* component) with too much code, which would make refactoring, correction and evolution more complex. Moreover, the *processing line* component, managing effective operator connections and execution (represented by "gears" in Fig. 11), can therefore be more easily reused for other softwares dealing with another context: marker detection and 2D/3D registration instead of organ segmentation, as in the case of the augmented reality based application presented in [7], for instance. In such a case, this *processing line* component would only require to be attached to another appropriate core component (similar to our *document* one) which would be specific to this new software.

6. Discussion and perspectives

The proposed architecture has been proved to be relevant in the sense that the software provides the required functionalities described in Section 2. This has been evaluated by performing several coarse modelings of sets of medical structures, as illustrated in [16]. Some image processing algorithms have been easily developed without any modification or adaptation of other components.

Our patient modeling software could be compared with modular systems such as MeVisLab [17], which seems to be highly modular and evolutive in terms of medical image processing algorithms. This appears to be the main focus of this system, as illustrated by publications involving medical image analysis (see [24,25] for instance). In our sense, one of the main originalities of our approach relies on the use of dynamic components for the various aspects (not solely restricted to image processing algorithms) of our application field: visualization, graphical user interface, processing graph and algorithms. We expect that the implemented foundations, including the XML-based contribution mechanism in particular, will facilitate the

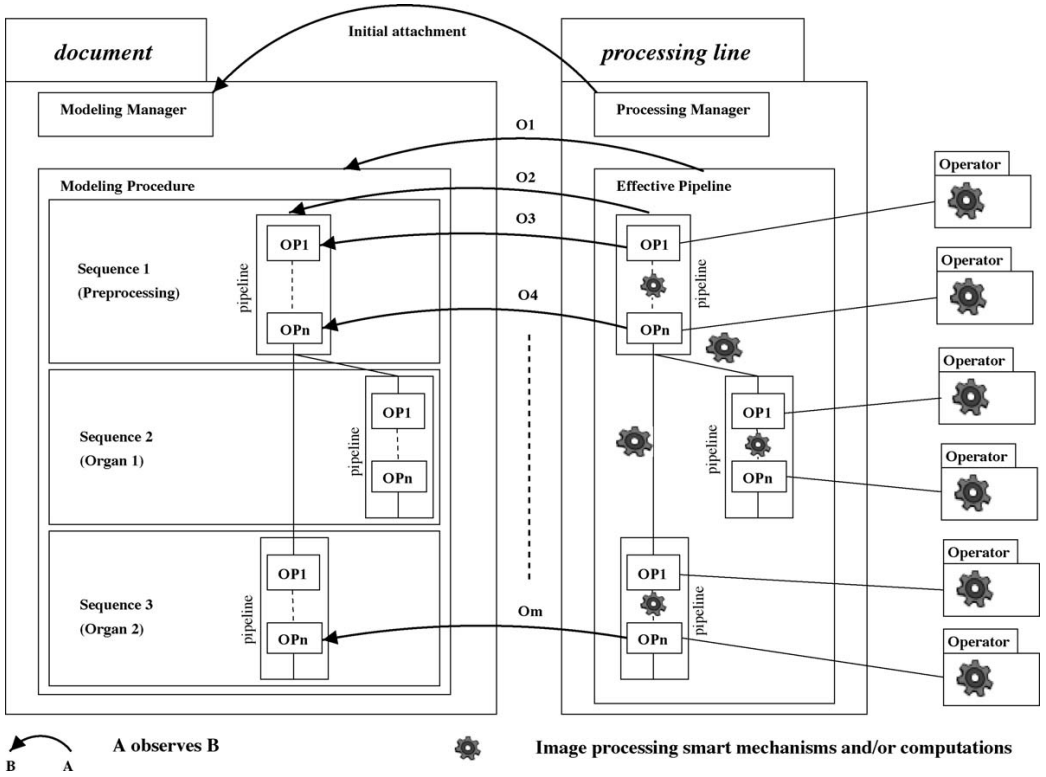

Fig. 11 – Observation based communication between *document* and *processing line* components. This illustrates our communication system, which is centered on *document* and based on the *Observer* Design Pattern.

development of additional softwares which do not necessarily involve image processing (e.g. surgical planning and simulation, as later discussed).

Concerning image processing operators, it is possible to integrate algorithms issued from third party libraries such as ITK [19] for instance. One only requires appropriate data structure converters. ITK-based algorithms we implemented process our data which are converted into ITK structures (the result being converted back to our data structure to deliver operator output data to our system). Although ITK pipelines can therefore be integrated, they are not directly connected to our pipeline as part of the graph (can basically be seen as "functions"). The advantage is that our image processing components do not depend on third party libraries. Dependencies can occur at algorithm level, resulting from the choice of the developer to use third party frameworks (requiring appropriate data structure converters), instead of directly interacting with our own data structures.

It would be useful to extend image processing functionalities by creating interfaces for interpreted languages, as considered by ITK [26] with CableSwig for instance. The fast integration and evaluation of algorithms would therefore be further facilitated, as well as the scripting of image processing sequences, as considered in MeVisLab [17]. In our sense, our framework is generic and modular enough to support such an extension.

Another interesting extension of our system would concern the integration of interactivity in image processing operators, with an approach similar to the one proposed in [27]. This would be useful to directly interact with the 3D data for guiding algorithms: selecting image points as seeds for propagation based algorithms, selecting image intensities, manually deforming 3D models ... Interactive image segmentation would be further facilitated. Our system is conceived to support such an extension by simply contributing to the *render* component using the *Strategy* Design Pattern. In such a case, this component must provide an *extension-point*, associated with a *Strategy* interface we could name `IInteractor`, to which some *operator* components could contribute (declaration of an *extension*).

Another point concerns the image processing graph shared between the *document* component and the *processing line* one. In our sense, the descriptive representation, close to the medical aspect, could be associated to a specific dedicated component (e.g. a *processing manager* component). Indeed, incorporating it within the general purpose *processing line* component does not seem pertinent, as previously underlined. Moreover, pure medical structures and associated general information (e.g. color, type) incorporated within the *document* should be independent from the processing aspect. For instance, for surgery planning, modeled medical structures are required while image processing is useless. This *processing manager*

could be dedicated to the translation of user actions and to implicit mechanisms which could be specific to patient modeling (similarly to the current one determining regions of interest according to medical structure types), transforming our software into an expert system (the "*expert*" being the *processing manager* component).

The next software to be developed with the proposed component oriented architecture will be dedicated to surgery planning. Functionalities related to such a software concerns, for instance, the incorporation of manipulation tools to interact with modeled medical structures (e.g. for liver resection [3]). These functionalities could be provided by a specific component contributing to the *ui* and *render* components. Both *Observer* and *Strategy* Design Patterns could be advantageously used for this. In our sense, most existing components could be kept, with minimal refactoring (e.g. splitting the *document* component). This is really more convenient than rewriting an entire software, including a graphical user interface, the management of the rendering of 3D structures, the definition of medical structures... Similarly, our current software architecture could be extended to incorporate virtual surgical tools for simulation purposes.

Concerning software engineering aspects, despite component construction [22] and modeling [22], combination/configuration is essential to reuse components for several softwares. In our sense, our *runtime* system provides the required foundations, allowing to easily make it evolve so that this aspect can be defined with a high level of abstraction, thanks to formal XML descriptions. For instance, we can define the set of required contributions according to the manipulation mode of the software, as underlined in Section 5: working in research mode (e.g. with an extended view of the processing graph), in basic view mode (only appropriate operators are started)... In our sense, a limited improvement effort is required to manage this evolution.

7. Conclusion

This paper presents a modular component oriented architecture of a software dedicated to medical image processing. Additionally to the efficient programming techniques involved, the proposed formal XML-based *runtime* system constitutes a powerful approach facilitating the management of contributions between components. This work has been proved relevant to easily extend software functionalities in terms of image processing algorithms.

Besides the considered application, an originality of this work is the application of efficient and modern methods, related to software engineering, to the emerging field of computer aided surgery, involving surgical planning, simulation, image guided intervention... In our sense, it is highly useful for computer scientists working in this field and trying to optimize the reusability of their work.

The next step will consist in extending our software functionalities so that operators can be directly configured or guided from the part dedicated to data visualization in order to further facilitate interactive modeling procedures. Additionaly, we expect to prove the evolutivity and the reusability of our component oriented architecture by developing, with minimal refactoring, new softwares dedicated to surgical planning and simulation.

Acknowledgements

First of all, we would like to Pamela Lhote for paper revision. We are very grateful to the French National League against Cancer, the Alsace Region, the French Research Ministry and the French Agency for Innovation for their financial support. We would also like to thank all our medical partners, who have provided us medical images of patients, and in particular Prof. Afshin Gangi and Prof. Catherine Roy from the radiological department of the Civil Hospital in Strasbourg. We also thank the surgical team of Prof. Marescaux for their help in our works, in particular Prof. Mutter, Prof. Leroy, Dr. Vix.

REFERENCES

[1] J. Duncan, N. Ayache, Medical image analysis: progress over two decades and the challenges ahead, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 85–106.
[2] M. Brown, M. McNitt-Gray, Handbook of Medical Imaging, vol. 2, Spie Press, 2000.
[3] H.-P. Meinzer, M. Thorn, C.E. Cardenas, Computerized planning of liver surgery—an overview, Comput. Graphics 26 (2002) 569–576.
[4] C. Forest, H. Delingette, A.N., Surface contact and reaction force models for laparoscopic simulation, in: Proceedings of the Second International Symposium on Medical Simulation, Lect. Notes Comput. Sci. 3078 (2004) 168–176.
[5] R. Taylor, D. Stoianovici, Medical robotics in computer-integrated surgery, IEEE Trans. Robotics Automat. 19 (5) (2003) 765–781.
[6] R. Ginhoux, J. Gangloff, M. De Mathelin, L. Soler, M. Arenas, J. Marescaux, Active filtering of physiological motion in robotized surgery using predictive control, IEEE Trans. Robotics 21 (2005) 67–79.
[7] S. Nicolau, A. Garcia, X. Pennec, L. Soler, N. Ayache, An augmented reality system to guide radio-frequency tumor ablation, Comput. Animat. Virtual Worlds 16 (2005) 1–10.
[8] M. Fowler, Refactoring. http://www.martinfowler.com/bliki/DefinitionOfRefactoring.html.
[9] C. Szyperski, Component Software: Beyond Object-Oriented Programming, 2nd ed., Addison-Wesley, 2002.
[10] I. Crnkovic, M. Larsson, Challenges of component-based development, J. Syst. Softw. 61 (2002) 201–212.
[11] W.B. Frakes, K. Kang, Software reuse research: status and future, IEEE Trans. Softw. Eng. 31 (7) (2005) 529–536.
[12] R. Van Ommering, Software reuse in product populations, IEEE Trans. Softw. Eng. 31 (7) (2005) 537–550.
[13] P. Papajorgji, A plug and play approach for developing environmental models, Environ. Model. Softw. 20 (2005) 1353–1357.
[14] U. Zdun, Tailorable language for behavioral composition and configuration of software components, Comput. Lang. Syst. Struct. 32 (2006) 56–82.
[15] W3C, XML Schema specification. http://www.w3.org/XML/Schema.
[16] J.-B. Fasquel, V. Agnus, J. Moreau, S.L.J. Marescaux, An interactive medical image segmentation system based on the optimal management of regions of interest using topological medical knowledge, Comput. Meth. Programs Biomed. 82 (2006) 216–230.

[17] MeVis-Center for Medical Diagnostic Systems and Visualization, Medical image processing and visualization. http://www.mevislab.de.
[18] Mercury Computer systems, Advanced 3D visualization and volume modeling. http://www.amiravis.com/.
[19] National Library of Medicine, Insight segmentation and registration toolkit. http://www.itk.org.
[20] The Visualization Toolkit. http://www.vtk.org.
[21] E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns, 1st ed., Addison-Wesley, 1995.
[22] M. Jonge, Build-level components, IEEE Trans. Softw. Eng. 31 (7) (2005) 588–600.
[23] The XML Library for Gnome. http://www.xmlsoft.org.
[24] D. Keller, S. Wildermuth, T. Boehm, T. Boskamp, D. Mayer, H. Schuster, B. Marincek, H. Alkadhi, CT angiography of peripheral arterial bypass grafts: accuracy and time-effectiveness of quantitative image analysis with an automated software tool, Acad. Radiol. 13 (5) (2006) 610–620.
[25] T. Kalteis, M. Handel, T. Herold, L. Perlick, C. Paetzel, J. Grifka, Position of the acetabular cupaccuracy of radiographic calculation compared to CT-based measurement, Eur. J. Radiol. 58 (2) (2006) 294–300.
[26] National Library of Medicine, Insight segmentation and registration toolkit, CableSwig. http://www.itk.org/HTML/CableSwig.html.
[27] I. Wolf, M. Vetter, I. Wegner, T. Bottger, M. Nolden, M. Schobinger, M. Hastenteufel, T. Kunert, H.-P. Meinzer, The medical imaging interaction toolkit, Med. Image Anal. 9 (2005) 594–604.

The Journal of Systems and Software 84 (2011) 847–863

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

A design pattern coupling role and component concepts: Application to medical software

Jean-Baptiste Fasquel [a,*], Johan Moreau [b]

[a] LISA - Angers University, 62 avenue Notre Dame du Lac, 49000 Angers, France
[b] IRCAD, 1 place de l'hôpital, 67091 Strasbourg, France

ARTICLE INFO

Article history:
Received 9 July 2009
Received in revised form 12 January 2011
Accepted 12 January 2011
Available online 22 January 2011

Keywords:
Design pattern
Dynamic composition
Collaboration
Role
Component
Medical software

ABSTRACT

One of the challenges in software development regards the appropriate coupling of separated code elements in order to correctly build initially expected high-level software functionalities. In this context, we address issues related to the dynamic composition of such code elements (i.e. how they are dynamically plugged together) as well as their collaboration (i.e. how they work together). We also consider the limitation of build-level dependencies, to avoid the entire re-compilation and re-deployment of a software when modifying it or integrating new functionalities. To solve these issues, we propose a new design pattern coupling role and component concepts and illustrate its relevance for medical software. Compared to most related works focusing on few role concepts while ignoring others, the proposed pattern integrates many role concepts as first-class entities, including in particular a refinement of the notion of collaboration. Another significant contribution of our proposal concerns the coupling of role and component concepts. Roles are related to the functional aspects of a target software program (composition and collaboration of functional units). Components correspond to the physical distribution of code elements with limited build-level dependencies. As illustrated in this paper, such a coupling enables to instantiate a software program using a generic main program together with a description file focusing on software functionalities only. Related code elements are transparently retrieved and composed at run-time before appropriately collaborating, regardless the specificity of their distribution over components.

1. Introduction

This paper proposes a design pattern addressing three challenging issues, corresponding to the dynamic composition of separated code elements, their collaboration and the reduction of build-level dependencies.

Dynamic composition (i.e. at run-time) is a useful feature for plugging software components together (Bass et al., 1998; Zdun, 2006). Unfortunately, composition is often static (i.e. at compile time). Such lack of dynamicity concerns the aggregation or composition with specific classes or objects, traditional C++ and Java inheritance relationships, generative programming approaches with hard coded template parameterization (Czarnecki and Eisenecker, 1999). The consequence is that a large part of a program must often be adapted (re-written) to composition or API (*Application Programming Interface*) changes.

Many approaches have been proposed for dynamic composition. Aspect oriented programming (AOP) aims at injecting some code elements at run-time within class methods (class-level composition), changing the behaviour of these methods for all instances of a given class (Villazon et al., in press). Mixin-based composition techniques also focus on run-time composition at class-level, except in the recent proposal coupling both class-level and object-level composition (Zdun et al., 2007). Some other approaches favour composition at object-level, as in our case. The well-known *decorator* pattern (Gamma et al., 1995) aims at changing the behaviour of object methods at run-time without subclassing. Delegation-based approaches (Lieberman, 1986; Bettini et al., in press) regard dynamic composition as a way to dynamically change or define the behaviour of object methods. Role-based approaches address object-level composition using many concepts being out of the scope of other composition alternatives (Kristensen and Osterbye, 1996; Steimann, 2000; Herrmann, 2007; Baldoni et al., 2007; Colman and Han, 2007). As recently underlined (Herrmann, 2007), due to the richness of these concepts, existing role-based implementation proposals often focus on some of them while ignoring others.

Collaboration represents a challenging issue dealing with the coordination or chaining of the execution of separated code elements. This notion is sometimes called method chaining, method resolution order, linearization of method invocation (Zdun et al., 2007). In many cases, this issue is addressed using a dedicated

* Corresponding author. Tel.: + 33 0241226556; fax: + 33 0241226561.
E-mail address: Jean-Baptiste.Fasquel@univ-angers.fr (J.-B. Fasquel).

0164-1212/$ – see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2011.01.026

entity (e.g. *behavioural link* (Tanter et al., 2008) or coordination aspect (Fuentes and Sanchez, 2007) in AOP, a mixin list for mixin-based composition (Zdun et al., 2007), a *Team* (Herrmann, 2007) or a coordination role (Colman and Han, 2007) in recent role-based proposals). As underlined in this paper, the use of a dedicated entity (explicit chaining) is appropriate when the chaining can be foreseen. Unfortunately, this is not necessarily the case in event-based software programs involving a graphical user interface (gui), as we consider in this paper, where end-user interactions are unforeseen and therefore may become hard to model into a dedicated entity.

The limitation of build-level dependencies also represents a challenging issue. Build-level dependencies can, for instance, result from a link dependency between the main program and dynamic libraries providing functionalities. To integrate any newly developed code element (embedded in new dynamic library), the re-compilation as well as the re-deployment of the entire program can therefore be required, involving the need of a development environment. Component-based development, widely used for software composition (Szyperski, 2002), represents an interesting paradigm to reduce build-level dependencies (i.e. link dependencies). Practically, it can enable to integrate a new dynamic library (provided by a component) without recompiling the main program. A difficulty is to appropriately define components, characterized by well defined interfaces and explicit dependencies (Szyperski, 2002; Lau and Wang, 2007), so that the main program can retrieve, at run-time, compiled code elements which are subject to composition and collaboration with other code elements.

In this paper, we propose a design pattern coupling role and component concepts, for dynamic composition at object-level and collaboration, with limited build-level dependencies. As further detailed in Section 4, we claim four contributions: The integration of many role concepts as first-class entities of the pattern, including the notion of collaboration generally ignored by related role-based approaches (Herrmann, 2007). The refinement of the notion collaboration, involving two types of chaining (*explicit* and *implicit*) as first-class entities of the proposed pattern. The integration of the notion of component, where interfaces and dependencies are directly mapped on role entities. A particular component description formalism, involving a graph-based representation, is proposed for the transparent retrieval of functional units of composition (i.e. roles), regardless their distribution over components providing them. To our knowledge, the coupling of both role and component concepts has never been considered, except in the context of a development methodology (Parka et al., 2004), this being out of the scope of this paper. The application of the proposed pattern for developing a set of software programs dedicated to medical applications. These applications are built using a generic main program which, according to a description file, retrieves and composes functionalities provided by software components so that they appropriately collaborate at run-time. As illustrated, composition and collaboration can evolve at run-time, according to end-user interactions.

Section 2 presents the proposed design pattern. Section 3 focuses on its application to medical software. Before concluding, Section 4 compares our proposal with some recent related works.

2. Design pattern

The design pattern is described according to three essential elements considered (Gamma et al., 1995) for specifying a pattern. The first element concerns the problem addressed by the pattern (Section 2.1), indicating when to apply it. The second element, detailed in Section 2.2, regards the proposed solution (i.e. the different elements of the design). The third element is related to the consequences (i.e. the result of applying the pattern), being summarized in Section 2.3.

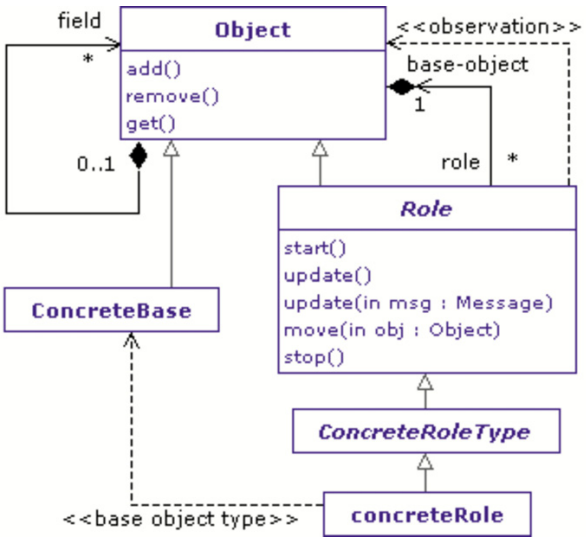

Fig. 1. Design pattern, participants and relationships (class diagram).

Note that, according to the classification considered (Gamma et al., 1995), the proposed design pattern has both structural and behavioural purposes, corresponding to composition and collaboration. Moreover it focuses on object (*scope* criterion; Gamma et al., 1995), because dynamic composition is considered at object-level.

2.1. Problem

The problem regards issues addressed in this paper (i.e. regarding dynamic composition, collaboration and build-level dependencies).

The proposed pattern aims at being applied for developing applications involving many various functionalities that may be unknown to each other and which should be composed at run-time and should correctly collaborate in order to provide initially expected high-level software functionalities. Moreover, this pattern concerns situations where code elements are embedded in components (limitation of build-level dependencies) and should be transparently retrieved, regardless the identity and location of components providing them.

Note that this pattern aims at being particularly appropriate for developing a set of software programs (rather than a single one) sharing common functionalities and data structures to be reused from one application to another. This is illustrated in Section 3 for a set of medical software programs involving various functionalities (e.g. graphical user interface, data reading/writing, 3D visualization and data processing) together with several kind of data structures (e.g. 3D image, video), all of them being provided by components.

2.2. Solution

Fig. 1 provides a class diagram reporting participants involved in the proposed design pattern (Gamma et al., 1995). The `ConcreteBase` class correspond to data structures. The `Role` class regards the notion of functionality. The `ConcreteRoleType` class is related to a functionality type (e.g. notion of data reading or 3D visualization) and the `ConcreteRole` class represents a particular implementation of a given functionality type (e.g. a specific data reader). The `Object` class enables to bind functionalities and data structures.

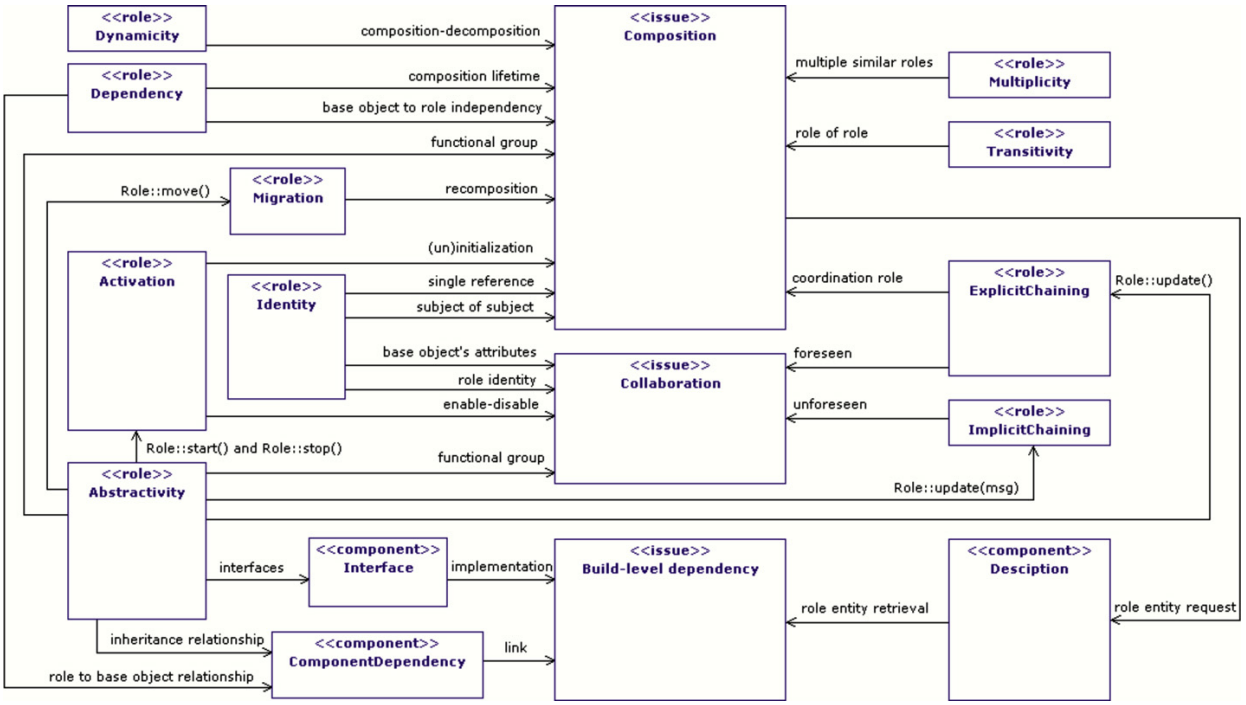

Fig. 2. Design pattern, concepts and issues. Class diagram providing an overview of role (`«role»`) and component (`«component»`) concepts involved in the proposed design pattern for issues (`«issue»`) regarding composition (`Composition`) and collaboration (`Collaboration`) with limited build-level dependencies (`Build-level dependency`). Relationships (explained in Sections 2.2.1 and 2.2.2) summarize how each concept interacts with (or participate to) issues and/or other concept(s).

Details of the proposed pattern, including in particular those related to Fig. 1, are given in next Sections 2.2.1 and 2.2.2, focusing respectively on role and component concepts on the top of which the pattern is built. Role concepts essentially correspond to the part of the proposed pattern solving dynamic composition and collaboration issues. Component concepts pattern mainly focus on the limitation of the build-level dependencies.

The way these concepts, representing the foundations of the proposed design pattern, contribute to solving considered issues is reported in Fig. 2, being also detailed in Sections 2.2.1 and 2.2.2.

2.2.1. Role concepts

First, we present role concepts mainly focusing on dynamic composition. The *Identity* feature concerns composition units and result. The *Dependency* feature regards dependencies between composition units, as well as composition lifetime. Other features also related to dynamic composition correspond to *Dynamicity* (dynamic composition and de-composition), *Multiplicity* (composition of several similar functionalities), *Transitivity* (composition of composition) and *Migration* (dynamic re-composition).

Then, we consider the two role concepts, namely *Explicit chaining* and *Implicit chaining*, focusing on collaboration.

Finally, we present the two last role concepts involved in the proposed pattern. They concern both composition and collaboration issues, and are related to *Activation* and *Abstractivity*.

Note that role concepts commonly considered are *Identity*, *Dependency*, *Dynamicity*, *Multiplicity*, *Abstractivity* and *Transitivity* (Kristensen and Osterbye, 1996). *Activation* and *Explicit chaining* have been recently proposed for collaboration (Herrmann, 2007). *Implicit chaining* concerns our proposal for managing unforeseen collaborations.

2.2.1.1. Identity. An object (or subject; Kristensen and Osterbye, 1996) is initially considered as a base-object (`ConcreteBase` in Fig. 1), being restricted to intrinsic methods and attributes (Kristensen and Osterbye, 1996; Herrmann, 2007). As traditionally considered (Herrmann, 2007), composition regards the fact that the base-object can acquire role(s), including methods and (dynamic) attribute(s) (`field` in Fig. 1). Acquired methods correspond to methods of `Role`, `ConcreteRoleType` and `ConcreteRole` classes in Fig. 1. The composition result is illustrated by the example of subject reported in Fig. 3.

A dynamically composed subject appears as a monolithic unit (base-object with attached roles, including intrinsic and dynamic attributes), having its own identity, which can present different perspectives or roles to other objects (or subjects) manipulating it or interacting with it. An object has or plays a given role if one instance of such a role is attached to it. As a consequence, a single reference on the base-object is enough to retrieve composition units (`get()` method of the `Object` class in Fig. 1), corresponding to attached roles (Fig. 2 ("single reference")).

Within this monolithic entity, the union of all attributes (intrinsic base-object and role attributes, as well as dynamic attributes) constitutes the overall state of the subject (Herrmann, 2007). Base-object's attributes (both intrinsic and dynamic ones) are used by attached roles to collaborate (Fig. 2 ("base object's attributes")): if the role execution depends on attributes of another role, this other role assigns these attributes to its base-object, as dynamic attributes. This is further discussed in next paragraphs focusing on collaboration-based role concepts (*Implicit chaining* and *Explicit chaining*). It must be pointed out that dynamic attributes can be seen as base-objects (similarly to `ConcreteBase`) to which roles

Fig. 3. Design pattern and composition: subject example (object diagram).

can be attached: a subject can therefore be composed of subject(s) (Fig. 2 ("subject of subject")).

As discussed in (Steimann, 2000; Herrmann, 2007), each role keeps its own identity (e.g. the concrete role entity appears distinct from the base-object one in both Figs. 1 and 3). Although appearing contradictory as roles belong to a single monolithic entity (Herrmann, 2007), this is required for collaboration (Fig. 2 ("role identity")), as underlined in next paragraphs regarding collaboration-based role concepts.

2.2.1.2. Dependency. As reported in Kristensen and Osterbye (1996), this feature regards the fact that a role cannot exist without its base-object, and, to ensure the separation of concerns, a (base) object is independent of any role (Fig. 2 ("base object to role independency")).

The role to base-object dependency remains strong because a role interacts with its base-object and therefore depends on its methods and attributes. In Fig. 1, this is represented by the dependency between `ConcreteRole` and `ConcreteBase` (`«base object type»` dependency) and the navigability of the association between `Role` and `Object` classes. This feature has an impact on the management of object lifetime (Fig. 2 ("composition lifetime")): base-object destruction involves the implicit detachment and destruction of attached roles (together with dynamic attributes). This is represented by composition relationships in Fig. 1.

2.2.1.3. Dynamicity. This feature regards both dynamic composition and decomposition (Fig. 2 ("composition–decomposition")): roles (and related dynamic attributes) can be attached (composition) and detached (decomposition) during the lifetime of the object (`add()` and `remove()` methods of the `Object` class in Fig. 1).

2.2.1.4. Multiplicity. This feature corresponds to the ability to attach several instances of the same concrete role to a given base-object (Fig. 2 ("multiple similar roles")), each role instance having its own state (i.e. value of its attributes). In Fig. 3, this could be represented by two instances of `ConcreteRole` attached to the same `ConcreteBase` instance.

2.2.1.5. Transitivity. This notion regards the ability to attach a role to another role, being considered as a base-object (Fig. 2 ("role of role")). In the proposed pattern, a role can be seen a base-object as the `ConcreteRole` inheritates from `Object` similarly to the `ConcreteBase` class (see Fig. 1).

2.2.1.6. Migration. This feature regards the ability to move (`move` method in Fig. 1) one role from a base-object to another (the role being temporarily unattached). This can be seen as a way to recompose (Fig. 2 ("recomposition")) base-objects and roles without destruction and (explicit) detachment. Compared to the *Dynamicity* feature (i.e. composition/decomposition), *Migration* avoids losing the role state.

2.2.1.7. Explicit chaining. This feature regards the collaboration of roles using a specific entity, as considered in (Herrmann, 2007; Colman and Han, 2007). The proposed pattern manages it using a dedicated role (coordination role, being a particular `ConcreteRole`) having a reference on coordinated roles, as illustrated in Fig. 4(a). The coordination role manages the invocation of `update()` method (see `Role` class in Fig. 1) on collaborating roles according to a given order, as illustrated by Fig. 4(b) reporting interactions between involved entities. Role execution (`update()` communication message) involves the modification (`modify()` communication message) of the value some base object's attributes (`ConcreteBase` instance).

Note that for collaboration purposes, a role can dynamically assign dynamic attributes to its base-object, as previously underlined (paragraph related to the *Identity* role feature). In Fig. 4(a), `ConcreteField` represents a dynamic attribute which could be assigned and modified by `ConcreteRoleA` (modification resulting from the `update()` method execution) and then used by `ConcreteRoleB` so that both roles appropriately collaborate.

As underlined in Section 1 and further illustrated in next Section 3, such chaining regards situations where collaboration can be foreseen (Fig. 2 ("foreseen")). Note that this feature also participates to composition through the use of a dedicated coordination role (Fig. 2 ("coordination role")).

2.2.1.8. Implicit chaining. This concept concerns situations where collaboration cannot be foreseen (Fig. 2 ("unforeseen")) and therefore not easily modelled using a dedicated entity, this being illustrated and discussed in Section 3.

In the proposed design pattern, *Implicit chaining* is based on an observation mechanism (`«observation»` relationship in Fig. 1, related to the *Observer* design pattern (Gamma et al., 1995)). As reported in terms of objects by Fig. 4(c), an observing role has an `observation` link with its base-object.

The idea is to define how a role behaves according to base-object changes, independently from the composition context (i.e. other attached roles). Roles naturally and implicitly collaborate after attachment: a role performs some computations, which may involve the modification of some base-object attributes (intrinsic and/or dynamic ones). This is represented by the `modify()` communication message sent by the `ConcreteRoleA` entity in Fig. 4(d). These modifications are notified to observing roles (`update(msg)` communication message in Fig. 4(d), and `update(msg)` method in Fig. 1). Then, these observing roles can perform some other tasks according to modifications reported in `msg` (`modify()` communication message sent by the `ConcreteRoleB` entity in Fig. 4(d)).

2.2.1.9. Activation. This feature is ignored except in Herrmann (2007) to trigger the attachment of roles. Compared to Herrmann (2007), we assume that roles are already attached. In the proposed pattern, this feature is considered for (un)initialization (Fig. 2 ("(un)initialization")) such as (de)allocation of some resources ((un)initialization of a composition unit).

As reported in Fig. 2 ("enable–disable"), it is also used to enable (or disable) a role so that it processes (or ignores) requests (invocation of `update()` or `update(msg)` method, depending on the considered chaining). Practically, this allows to (temporarily) disable the participation of a role to any collaboration without detachment (and therefore destruction), which could otherwise involve the loss of its state (value of its attributes). Conceptually, a subject can have a role without playing it (temporarily). In the proposed pattern, this feature corresponds to both `start()` and `stop()` methods of the `Role` class reported in Fig. 1.

2.2.1.10. Abstractivity. This feature regards the notion of role type (Zdun et al., 2007; Kristensen and Osterbye, 1996), corresponding to the classification of roles in functional groups. In the proposed pattern, this feature corresponds to three levels of the class hierarchy, namely `Role`, `ConcreteRoleType` and `ConcreteRole` (see Fig. 1).

Abstractivity at `Role`-level enables to summarize several role concepts, considered as first-class entities of the pattern: *Migration* (Fig. 2 ("Role::move()")), *Explicit chaining* (Fig. 2 ("Role::update()")), *Implicit chaining* (Fig. 2 ("Role::update(msg)")), *Activation* (Fig. 2 ("Role::start() and Role::stop()")).

This facilitates the manipulation of roles in an abstract manner, regardless their type (`ConcreteRoleType`-level) and particular implementation (`ConcreteRole`-level).

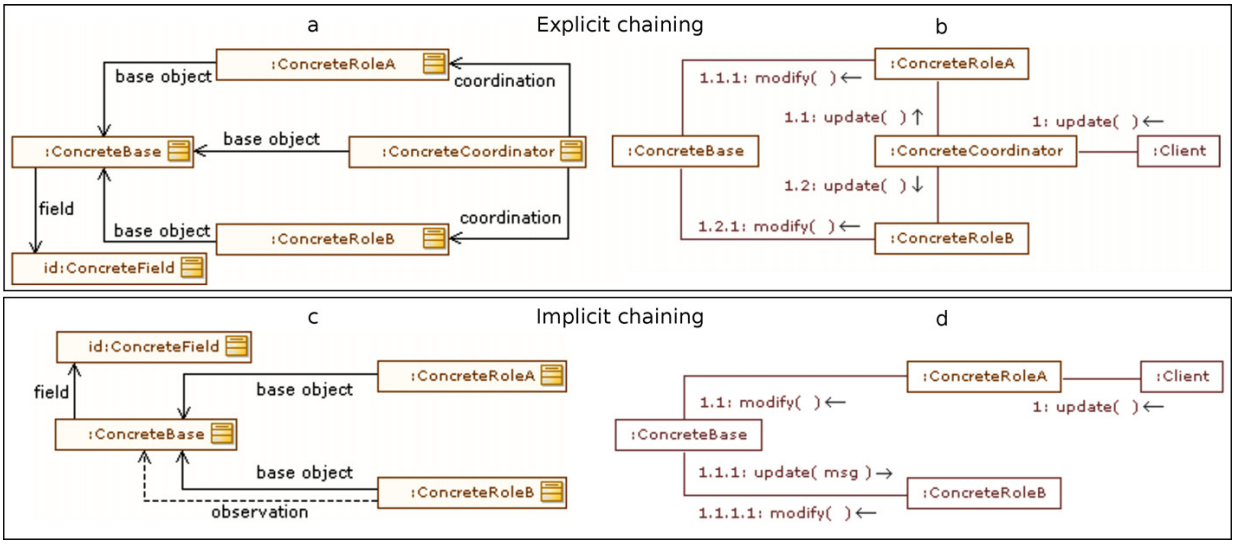

Fig. 4. Design pattern and collaboration. Explicit chaining (a and b diagrams) and implicit chaining (c and d diagrams) of two roles represented using two object diagrams (a and c) and two communication diagrams (b and d). The considered chaining concerns the execution of a role of type `ConcreteRoleA` first and then the execution of a role of type `ConcreteRoleB`, both of them being attached to a base object of type `ConcreteBase`. In communication diagrams, the `Client` entity represents a code element in charge of triggering chainings. The `modify()` communication message regards modifications of a base-object by one of its roles.

Abstractivity at `ConcreteRoleType`-level regards the management of role concepts for a specific type of functionality (this being illustrated in Section 3). It also concerns interactions with functional groups (Fig. 2 ("functional group")) regardless concrete roles. For instance, this can correspond to the detachment or (de)activation of an entire functional group, consisting in several roles of given role type (i.e. inheriting from the same `ConcreteRoleType`).

2.2.2. Component

Role entities reported in Fig. 1 are split over components (see Fig. 5) in order to reduce build-level dependencies. A component is a entity embedding at least a description of its content (description file) and a dynamic library. `Object` and `Role` entities are provided by a central `layer` component. Role types, concrete roles and base-objects are provided by other components, without any particular constraints regarding their distribution.

Details regarding this aspect of the proposed pattern are given hereafter in terms of component interfaces and dependencies (Szyperski, 2002; Lau and Wang, 2007) as well as in terms of description constraints. These three component features participate to the reduction of build-level dependencies, as reported in Fig. 2 by relationships between the three `«component»`-typed classes and the `Build-level dependency` one.

2.2.2.1. Interface. A component may require and/or provide interfaces (Lau and Wang, 2007), depending on embedded role entities. In the proposed pattern, interfaces are related to `Role`, `ConcreteRoleType` and `ConcreteRole` entities, as reported in Fig. 5(a).

The `Role` entity is associated with a required interface defined by a set of methods invoked by `Role` class methods (see Fig. 1). For instance, the `update()Role` method invokes a interface method to be implemented by a component providing a concrete role type (`ConcreteRoleType` entity). This can be seen as a translation or a mapping of some general role concepts (related to `Role` methods) at role type level (regarding `ConcreteRoleType` methods).

Any role type is also associated with a required interface. This interface defines a contract to be fulfilled by any component providing a concrete role compliant with a given concrete role type. This can be seen as a translation of role concepts at concrete role level (`ConcreteRole` entity).

Fig. 6(b) illustrates how components can be wired for a given composition (Fig. 6(a)).

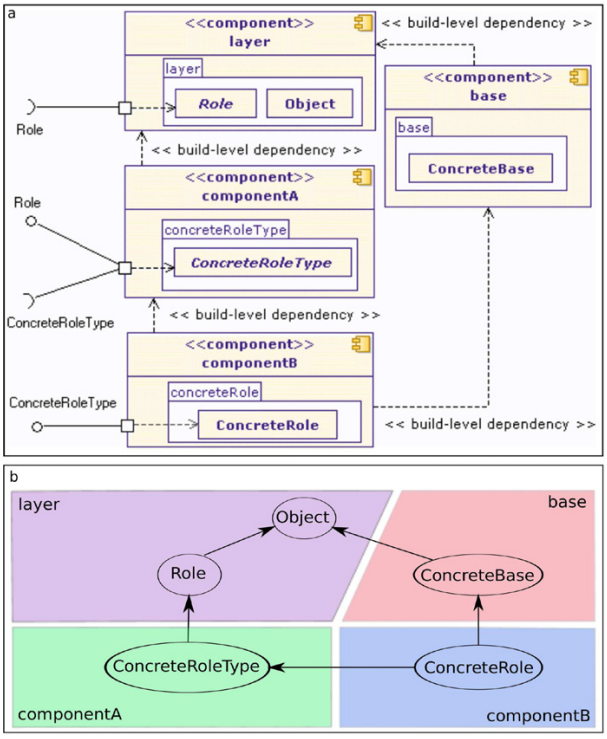

Fig. 5. Design pattern and components: structure, interfaces and dependencies (a) and graph-based descriptions (b).

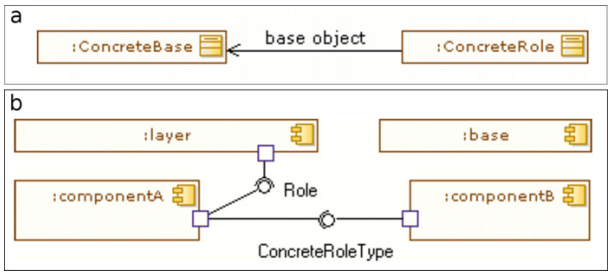

Fig. 6. Design pattern, components and composition. Example of a subject (a) with related components (b), according to the component distribution reported in Fig. 5.

As a result, component interfaces are associated with the *abstractivity* role feature, as they correspond to the same three levels in the class hierarchy (i.e. `Role`, `ConcreteRoleType` and `ConcreteRole`). This relationship between both concepts is indicated by Fig. 2 ("interfaces"). Fig. 2 ("implementation") represents interfaces to be implemented, involving build-level dependencies resulting from the required compatibility between declared interfaces and related implementations.

2.2.2.2. Dependency. A component may have dependencies (Lau and Wang, 2007) with respect to other components (`Component-Dependency` in Fig. 2), depending on embedded role entities. In the proposed pattern, dependencies result from inheritance relationships and `ConcreteRole` to `ConcreteBase` dependency (see Fig. 1).

In Fig. 5(a), the central `layer` component has no dependencies with respect to other components. Both `componentA` and `base` components have a dependency with respect to the `layer` component, and `componentB` depends on both `componentA` and `base`.

Fig. 6(b) illustrates components involved in the considered composition, where underlying dynamic libraries have to be loaded, taking their dependencies into account (e.g. `componentB` depends on `base`), so that both `ConcreteBase` and `ConcreteRole` can be instantiated and finally linked to compose the expected subject (Fig. 6(a)).

In terms of concepts, Fig. 2 indicates that both *dependency* and *abstractivity* role concepts contribute to component dependencies (Fig. 2 ("inheritance relationship") and ("role to base object relationship")). Fig. 2 ("link") regards link dependencies between dynamic libraries embedded in components.

2.2.2.3. Description. As previously underlined, except `Object` and `Role` entities provided by a central component (`layer` component in Fig. 5(a)), other role entities can be spread over components in different ways. For instance, a single component could be considered in Fig. 5(a) instead of three ones to embed `ConcreteRoleType`, `ConcreteRole` and `ConcreteBase` entities.

A part of the proposed design pattern concerns the abstraction of the component distribution with respect to embedded base-objects, role types and concrete roles that are involved in compositions. The underlying motivation is to be able to manage compositions (base-objects with attached roles) transparently, regardless the specificity of the distribution of components. A composition request can therefore remain focused on base-objects and roles, without explicitly indicating required components.

The considered design pattern proposes to support this abstraction by imposing a particular component description model where a description file declares each provided entity (being either a base-object, a concrete role type or a concrete role) as a node of a graph (see Fig. 5(b)). The resulting graph structure binds role and component entities, as reported in Table 1.

Table 1
Graph nodes and edges in terms of role and component entities. Considered graph and component distribution are those reported in Fig. 5. *dep* and *int* terms stand respectively for *dependency* and *interface*. `Object` represents the graph root. Although *inheritance* is reported in the *role* column, it regards a standard object oriented mechanism rather than a specific role concept.

	Role	Component
Graph node		
`ConcreteBase`	base-object	–
`Role`	role	required int
`ConcreteRoleType`	role type	provided int required int
`ConcreteRole`	concrete role	provided int
Graph edge		
`ConcreteBase` →`Object`	inheritance	build-level dep
`Role` →`Object`	inheritance	build-level dep
`ConcreteRoleType` →`Role`	inheritance	build-level dep
`ConcreteRole` →`ConcreteRoleType`	inheritance	build-level dep
`ConcreteRole` →`ConcreteBase`	base-object type	build-level dep

On one side, according to nodes and edges associated with a given component (each node knows its associated component), we can deduce its required interfaces, those it provides, as well as its dependencies with respect to other components. On the other side, we can deduce whether a given component provides a role type, a concrete role, a base-object or even a mix of all.

In Fig. 6(a), the composition involves a base-object of type `ConcreteBase` and a concrete role of type `ConcreteRole`. From graph analysis (see Fig. 5(b)), one can easily retrieve components whose dynamic libraries embed `ConcreteBase` (`base`) and `ConcreteRole` (`componentB`). According to graph edges, build-level dependencies are identified and appropriate dynamic libraries are loaded (i.e. those provided by `layer` and `componentA` components). Additionally, components are appropriately wired (see Fig. 6(b)).

This part of the design pattern is associated with the `Description` entity in Fig. 2. Related relationships describe the fact that, according to a given composition request (Fig. 2 ("role entity request")), the retrieval of a role entity (Fig. 2 ("role entity retrieval")) is performed through the analysis of the graph built from component descriptions.

2.3. Consequences

As further illustrated in next Section 3, using the proposed design pattern has several consequences: An entire application can be defined as a subject being recursively composed of subjects, each subject corresponding to a base-object (data structure) with attached roles (functionalities). This is illustrated in next Section 3, where an application is declared by specifying data structures and functionalities (i.e. subject(s)) using a simple tree-like structure provided as a configuration file to a generic main program. As further underlined in next Section 3, the composition can evolve at run-time. The generic main program has no build-level dependencies with respect to components providing data structures and functionalities. A functionality (or a data structure) can be integrated in an application by simply ensuring that the related component can be accessed by the application (i.e. reading its content and, in particular, loading the dynamic library). Thanks to the graph-based description, the functionality (or the data structure) can be transparently retrieved without explicitly indicating the related component. A functionality can implicitly collaborate with other functionalities by appropriately defining how it reacts according to possible modifications of the state (i.e. value of attributes) of the data structure it is attached to. A specific functionality can be used to manage the explicit collaboration of functionalities (i.e. how they work together). This specific

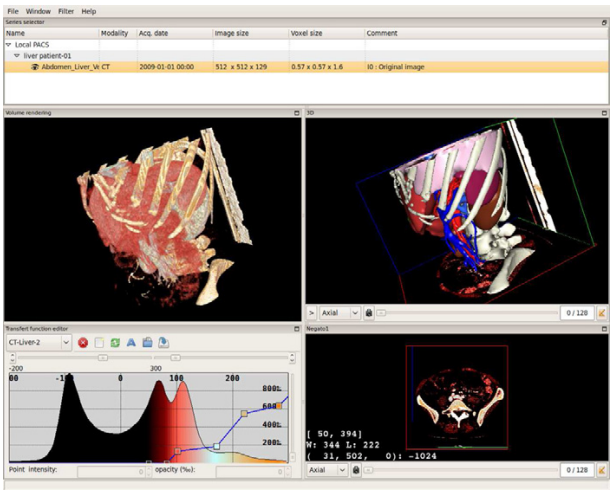

Fig. 7. Snapshot of the gui of the VR-Render freeware, rendering an image (middle-left and bottom-right rendering areas) and 3D surface models (middle-right rendering area), according to the end-user selection (top panel).

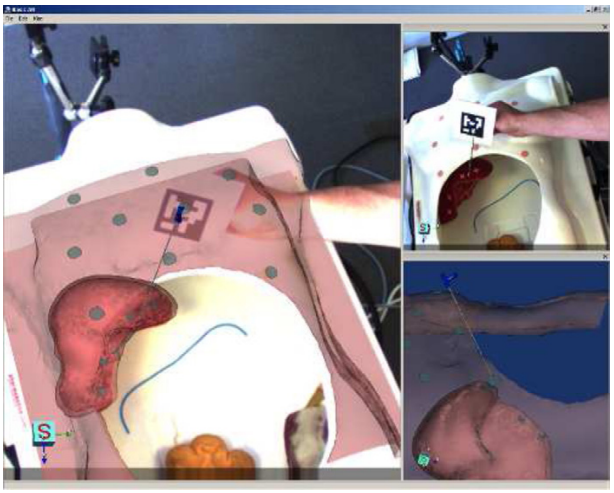

Fig. 8. Snapshot of the gui of the basicAugmentedReality software, mixing both video rendering and 3D surface model rendering. A role is used to track the position of the needle carrying a tag, involving transparent rendering refresh (see Fasquel et al., 2008 together with the provided video for functional details).

functionality (coordination role) is considered as any other functionality (i.e. can be integrated as any other functionality). The major drawback regards the required understanding of role concepts and their appropriate translation into a specific application domain (i.e. mapping general role concepts to functionality types and particular implementations). Next Section 3 provides examples of such a translation between these concepts and practical functionalities.

3. Case study

The proposed design pattern has been applied to medical software. Section 3.1 describes the application domain and the set of developed software programs. Section 3.2 presents an overview of the implementation of the proposed design pattern and three illustrating examples regarding functionalities which have been widely used in considered software programs. Section 3.3 discusses the relevance of the proposed pattern coupling both role and component concepts for developed applications.

3.1. Application domain and developed software programs

This application domain concerns a large community of computer scientists developing medical systems and software for computer aided diagnosis and therapy (DiMaio et al., 2007; Brown and McNitt-Gray, 2000; Duncan and Ayache, 2000; Taylor and Stoianovici, 2003). Considered software programs are often expected to integrate many kind of data structures (e.g. patients, images, videos and 3D surface models) and to combine various functionalities such as graphical user interface (gui), data reading/writing (io), 3D rendering (visualization), processing algorithms (e.g. medical image analysis for diagnosis purposes), video-based tracking (e.g. tracking of surgical tools in computer aided surgery) and medical robotics (see Fasquel et al., 2008, 2009 for examples).

Several medical applications have been developed on the top of this pattern. The VR-Render[1] freeware (see Fig. 7) is an advanced software focusing on visualization, together with gui, io and some processing functionalities. Involved data structures are patients, medical images and 3D surface models. The term "advanced" means that the application provides a large number of sophisticated functionalities with a specific effort on usability (International Organization for Standardization, 2010).

In addition to this advanced software, several basic applications (i.e. without specific effort on usability (International Organization for Standardization, 2010)) have also been developed. They were prototypes developed for research purposes or for testing many functional compositions covering our application field: BasicViewer focuses on the visualization of 3D medical images, together with gui and io functionalities (illustrated in next Section 3.2.3), BasicProcessor concerns medical image processing, together with BasicViewer's functionalitites (illustrated in next Section 3.2.4), BasicAugmentedReality involves 3D surface models, video processing, io, gui, visualization and tracking (see Fig. 8). BasicRobot aims at driving a medical robot (see Fig. 9), including several data structures (video, 3D surface models and robot) and functionalities (video processing, io, gui, visualization, tracking and command).

3.2. Design pattern implementation: overview and examples

3.2.1. Implementation overview

The proposed design pattern has been implemented using the C++ language, and represents the core of the fw4spl[2] open source project. The long-term purpose of this project is to provide a software product line for medical applications (fw4spl stands for *framework for software product line*). Note that some implementation details provided in next sections may slightly differ from the code available on the fw4spl website, either for clarity purposes or because fw4spl's code regularly evolves.

Several concrete base-objects (`ConcreteBase` in Fig. 1) have been developed, for the different data structures previously mentioned.

Several concrete role types (`ConcreteRoleType` in Fig. 1) have been implemented to manage the different functionalities supported in developed software programs. Next sections provide

[1] http://www.ircad.fr/softwares/vr-render/Software.php (January 2011).

[2] http://code.google.com/p/fw4spl/ (January 2011).

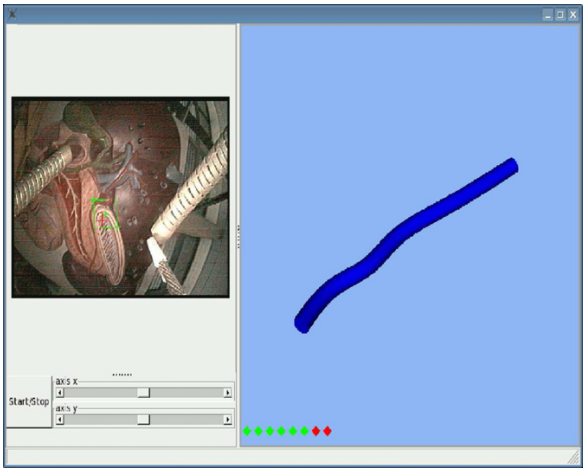

Fig. 9. Snapshot of the gui of the basicRobot software (see Fasquel et al., 2009 together with the provided video for functional details). Left panel: part dedicated to the robotized system, including an endoscopic view, an endoscope head control (both sliders) and a button to start/stop the automatic motion rejection. Right panel: part dedicated to 3D rendering of endoscope deformations (using a electromagnetic tracking of the endoscope head), for monitoring purposes.

examples regarding reading (Section 3.2.2), rendering (Sections 3.2.3 and 3.2.4), processing (Section 3.2.4) and gui (Section 3.2.4).

Several concrete roles (`ConcreteRole` in Fig. 1) have been developed for the different concrete role types and concrete base-objects. The largest number of concrete role per concrete role type have been developed for image, as it represents a central data structure of our application domain.

In the proposed implementation, a component is a directory containing a XML-based description file (examples are provided in next Section 3.2.2), a dynamic library as well as optional elements such as icons.

A central `layer` component, required whatever the medical software, has been implemented (similar to the `layer` component reported in Fig. 5). The underlying dynamic library (represented by the `layer` package in Fig. 5) implements all previously presented concepts regarding roles and components. For practical reasons, it also provides a container keeping a reference on all instantiated subjects to facilitate the retrieval of the instantiated roles, regardless their related base-object.

Other components provide the different concrete base-objects, concrete role types and concrete roles.

A software instance is defined by a root subject instance, declared in a XML-based description file provided to a generic launcher (or generic main program) depending on the `layer` package only.

3.2.2. Example: reading

This example regards the reading, from the filesystem, of a 3D medical image. Medical images are often stored in DICOM[3] format (DICOM stands for *Digital Imaging and Commmunications in Medicine*). Practically, a 3D medical image is stored in a repository as a set of 2D image files, as illustrated in Fig. 10. In addition to pure image information, these files provide medical information about the patient, the hospital and the acquisition protocol for instance. Due to the variability of the information and the specificity of imaging devices, many various reading algorithms exist.

[3] http://medical.nema.org/ (January 2011).

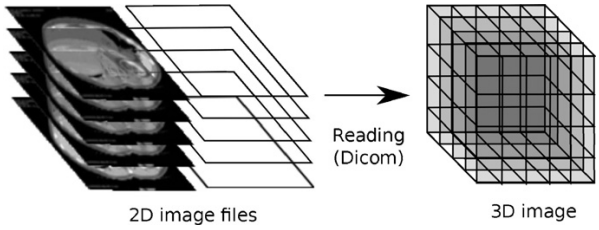

Fig. 10. Example combining a 3D medical image and a reading functionality: principle.

Components involved in this example are reported in Fig. 11. The `io` component regards the role type (the `ConcreteRoleType` corresponds to the `IReader` class) dedicated to data reading (and writing although not reported for clarity). The associated library (corresponding to the `io` package) embeds common functionalities required by concrete reading roles (e.g. management of interactions with the filesystem regardless the specificity of both concrete reading roles and concrete base objects). It provides an interface compliant with the `Role` one. It also declares a required interface as a contract to be fulfilled by components providing specific reading roles. The `dicom` component provides a concrete reading role and therefore an interface fulfilling the `IReader` required one declared by the `io` component. This interface is associated with the internal `Dicom` class, which depends on the appropriate base-object class (`Image` class, provided by the `data` component, because the DICOM-based reading role is dedicated to images).

As previously said, interfaces map role concepts related to `Role` methods at both concrete role type and concrete role levels. For instance, the `updateRole` invokes its pure virtual `updating()` method (part of the required interface) if the role is activated (`start` must have been called). The `IReader` class implements this `updating()` method (part of the provided interface) which, in turns, invokes its pure virtual `read()` method to be implemented by the `Dicom` class (part of the provided interface). The `read()` method is called only if, in this case, the location of 2D image files is valid (this being checked by the `IReader::updating()` method).

The `dicom` component depends (build-level dependency) on the `data` component as well as on the component regarding its associated concrete role type (i.e. `io` component in this case).

Component description files are also reported in Fig. 11. As illustrated, a description file may report several information: `<plugin id="..."/>` regards the component identifier (e.g. `io`). `<library name="..."/>` concerns the name of the C++ dynamic library (e.g. `libio`). `<point.../>` focuses on the declaration of `Object`, `Role`, concrete base-object type (e.g. `Image`), concrete role type (e.g. `IReader`) or concrete role (e.g. `Dicom`). Static inheritance relationships and role to base-object dependency are declared using `implements` XML elements.

The graph-based representation recommended by the proposed design pattern results from `point` XML elements, as illustrated by Fig. 12. `Point` XML elements are the nodes of the graph, where edges are defined by `implements` element values. As previously said, nodes could be spread over components in different ways, according to the component distribution. For instance, the `io` component could also provide a default image reader (default concrete role), which would lead to an additional node in the graph.

An example of root subject declaration, to be provided to the generic launcher, is reported in Fig. 13(a). This example involves a base-object of type `Image` to which one wishes to attach one role (instance of the `Dicom` class). The reading functionality is seen as a role played by an image, as illustrated by the object diagram reported in Fig. 13(b), resulting from the instantiation of the subject. Fig. 13(c) provides an additional diagram showing components involved in the composition, including their appropriate wiring.

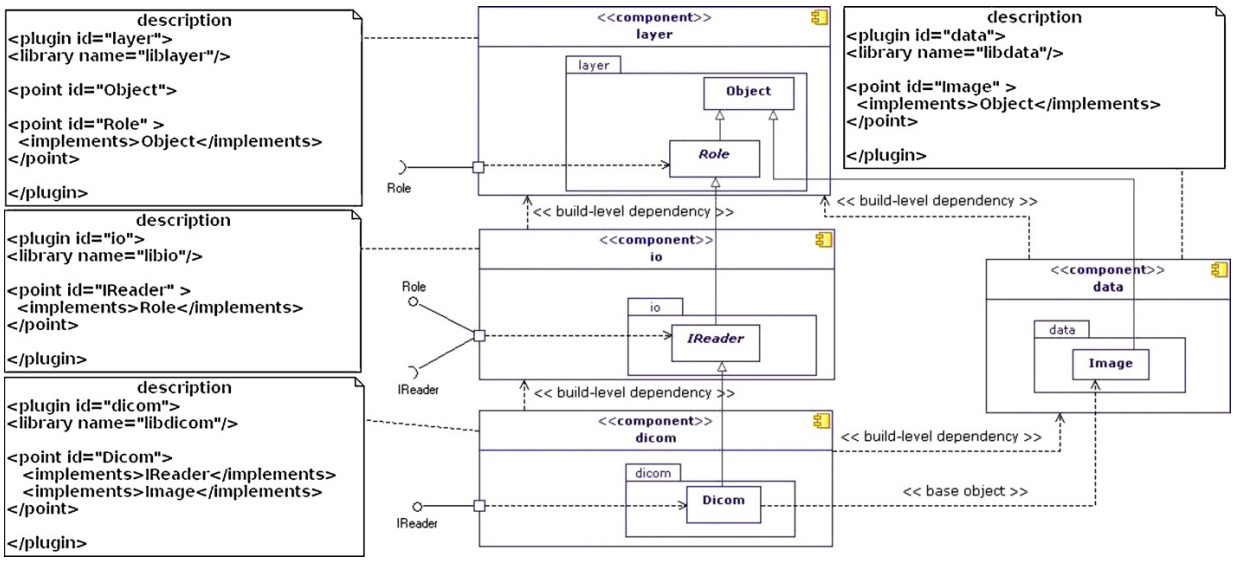

Fig. 11. Example dedicated to the reading functionality: components.

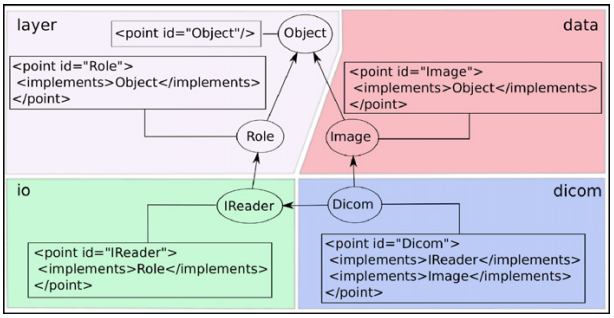

Fig. 12. Example dedicated to the reading functionality: graph-based description of components.

Hereafter, we propose to shortly describe the instantiation procedure of a software in the case of this example. Before instantiating the subject, the graph representation previously described is built from the analysis of the component description files, where components are organized as a set a directories on a filesystem. From the analysis of the graph, build-level dependencies are identified in order to start appropriate components. In this example, it concerns the `data` component because the subject's concrete base-object type is `Image` and the `dicom` component because the concrete role is of type `Dicom`. Note that starting a component means loading a dynamic library and performing some registrations in a factory maintained by the `layer` component, so that classes declared by components can be instantiated (e.g. an image can be instantiated after having started the `data` component). As the graph indicates the (build-level) dependency of the `dicom` component with respect to the `io` one (`Dicom` to `IReader` edge), the `io` component is started first (the `layer` component being already started). After having started appropriate components, both base-object and concrete role can be instantiated and attached.

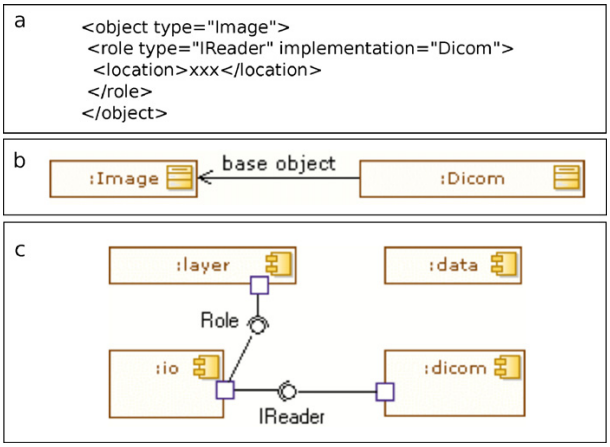

Fig. 13. Example dedicated to the reading functionality: subject declaration (a), subject instance (b) and involved components (c).

Note that the state or configuration (value of attributes) of base-objects and roles can be specified in root subject declarations. This is illustrated in Fig. 13(a) for the reading role with `location` XML element indicating the location of medical image files to be read. Although not detailed in this paper, such a configuration is performed by passing an abstract parameter to the `Role` entity.

An image can finally be read (according to the specified `location`) by invoking the `update()` method of the reading role: this leads to the updating of image attributes such as the data buffer.

3.2.3. Example: reading and rendering

This example concerns the combination of image reading and rendering functionalities, as illustrated in Fig. 14. In this application domain, different techniques can be used to display 3D medical images. The multi-planar rendering (*Mpr* for short) is one of the most classical technique. It consists of rendering slice(s) (i.e. 2D image(s)) of the full set of 3D points, each rendered slice being defined by a slice index the end-user can interactively modify to get across the full 3D data set. Other rendering techniques exist such as the volume rendering one (*Vr* for short), for displaying an entire 3D medical image. Note that both rendering techniques are supported by the VR-Render freeware. In this example, it is

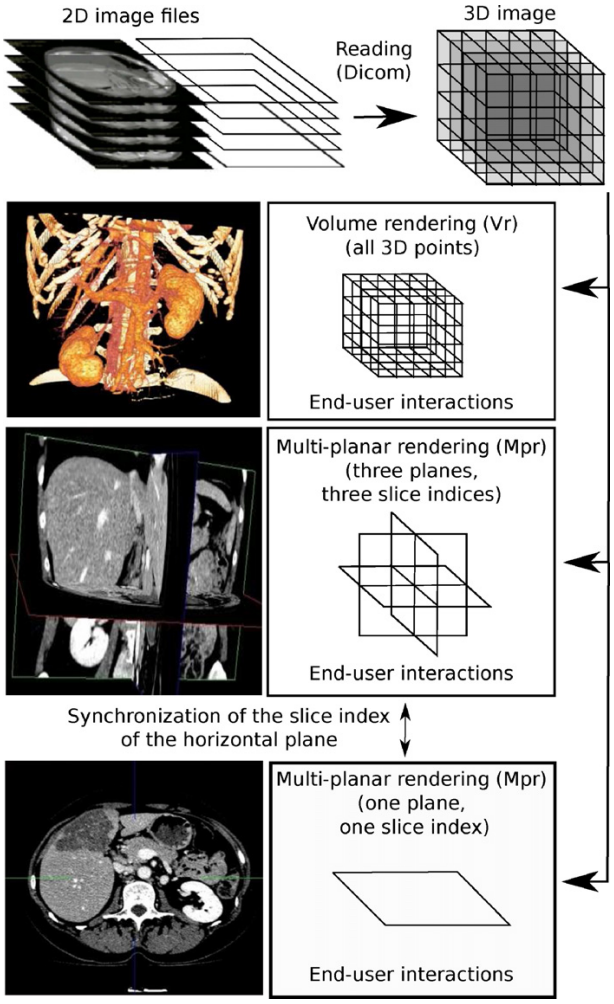

Fig. 14. Example combining reading and rendering functionalities: principle. Rendering functionalities (roles) encapsulate end-user interactions allowing to change rendering properties such as the slice index of the horizontal plane of `Mpr` roles.

expected that rendering areas automatically refresh after reading. As illustrated in Fig. 14, it is also expected that both *Mpr*-based rendering roles are synchronized with respect to the horizontal plane which can be modified through end-user interactions. If the end-user modifies the slice index of the horizontal plane to be rendered in one area, it is expected that the associated plane in the other area automatically changes according to this modification. Note that the end-user can perform modifications either in the *Mpr*-based rendering area configured (role state) to render three planes or in the one configured to render the horizontal plane only.

Fig. 15 reports the XML declaration of the subject (Fig. 15(a)), together with the resulting object diagram (Fig. 15(b)). Observation links are set using the `com` XML element (`com` stands for communication). These links allow to automatically refresh rendering areas after image reading: an event is raised by the reading role and caught by rendering ones which can refresh according to the new image buffer content. Observation links also enable to synchronize rendered horizontal planes according to end-user interactions. Planes are defined by a common slice index being stored in an `Integer` attribute assigned by *Mpr*-rendering roles

a

```
<object type="Image">
 <role type="IReader" implementation="Dicom">
  <location>xxx</location>
 </role>
 <role type="IRender" implementation="Vr" com="yes"/>
 <role type="IRender" implementation="Mpr" com="yes" ... />
 <role type="IRender" implementation="Mpr" com="yes" ... />
</object>
```

b

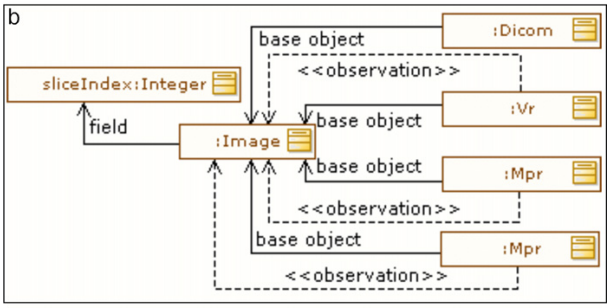

Fig. 15. Example combining reading and rendering functionalities: subject declaration (a) and instance (b).

to the base-object with a common identifier (`sliceIndex` in Fig. 14(b)).

Note that, besides *implicit chaining*, this example also illustrates the *multiplicity* feature as two similar rendering roles (`Mpr` ones, differently configured) are attached to the same base-object.

3.2.4. Example: reading, rendering, processing and graphical user interface

This example concerns the combination of a graphical user interface with 3D medical image reading, rendering and processing. In the considered application domain, many various image processing algorithms exist. An image processing algorithm aims at extracting or transforming image information in order to facilitate its understanding and therefore make a more efficient diagnosis for instance. The considered combination of functionalities is illustrated in Fig. 16. The algorithm considered in the example performs a morphological erosion of the image (Pratt, 2007), which involves some noise removal and an enlargement of dark areas (not detailed as out of scope of the present paper).

As illustrated by Figs. 17 and 18, the algorithm is seen as a role (`Erode`, identified by `algo`) to be played by the root base-object of type `Object`. This algorithm is configured to process the `input0` image (Fig. 16 (“Initial 3D image”) and `input0` in Figs. 17 and 18). This image is a dynamic attribute of the root base-object. This dynamic attribute has a `Dicom` role for loading the image to be processed, and a `Mpr` role for rendering purposes. This situation illustrates subject composition (i.e. a subject being composed of subject(s)). The `output` image (Fig. 16 (“Processed 3D image”) and `output` in Figs. 17 and 18) has also a rendering role.

An editor (`Editor` role in Figs. 17 and 18) embeds text controls to enter algorithm’s parameters (dynamic attributes of type `Float`). This editor corresponds to the left panel appearing in the snaptshot of the gui in Fig. 16.

The `Aspect` role (see Figs. 17 and 18) manages gui elements such as windows to be used by the `Editor` and both `Mpr` roles to install the gui-based functionalities they provide (`win` XML elements reported in Fig. 17).

`Action` defines an additional role appearing as a menu item in the gui (see the snapshot of the gui in Fig. 16). This role is in

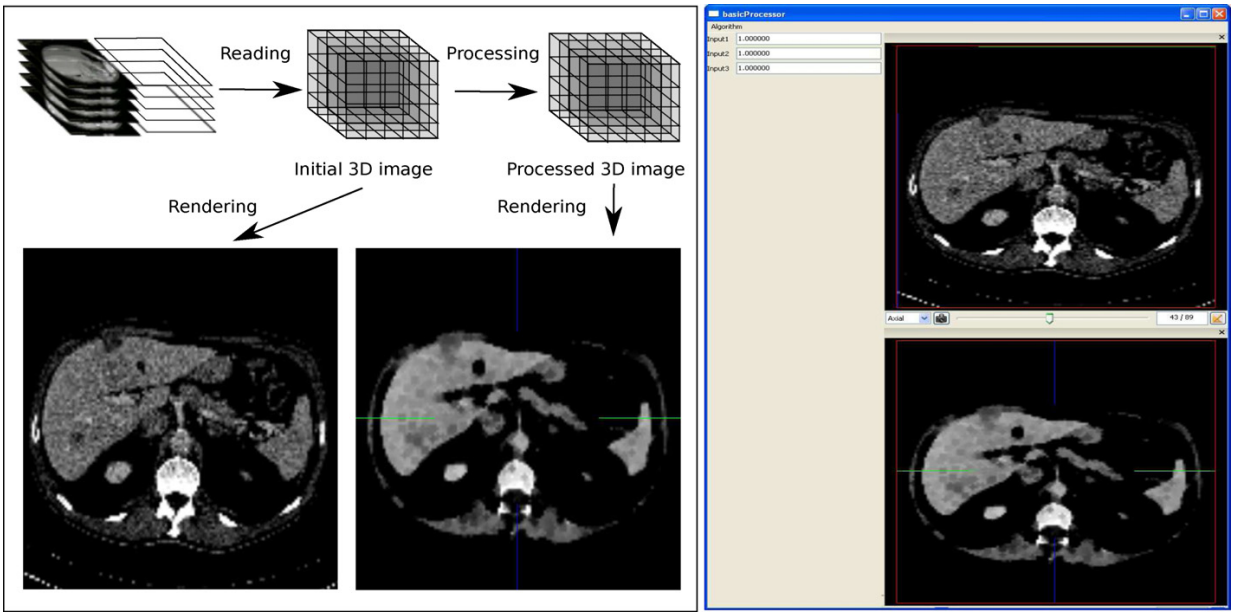

Fig. 16. Example combining a graphical user interface (gui) with reading, rendering and processing functionalities: principle (left area) and snapshot of the gui (right area).

charge of triggering the processing: when the end-user presses this menu item, the `Action` invokes the `update()` method of the `algo` role. This procedure illustrates the concept of explicit chaining in a basic situation where the `Action` role is a coordination role configured to trigger `algo` (see subject declaration reported in Fig. 17). After having run `algo`, the output image is updated, and its rendering role naturally refreshes (implicit chaining through observation).

Note that optional information can also be provided together with the root subject declaration, corresponding to `start` and `stop` XML tags in Fig. 17. The `start` tag indicates role(s) to be activated (specifying the concrete role type), according to a given order chosen to take functional dependencies into account. For instance, `IAspect` must be activated before `IRender`. Indeed, the layout of windows, within which rendering is expected to be performed, must be initialized first. The `stop` tag allows to specify the deactivation order of roles when one quits the application. Note that, when quitting a software, roles are naturally deactivated and destroyed consequently to automatic (root) subject destruction, even if not explicitly specified in the subject declaration.

3.3. Discussion

The purpose of the proposed pattern is to facilitate dynamic composition and collaboration of separated code elements without code rewriting or glue code. In particular, this includes the ability to reuse code elements developed for one software to another, as summarized in Section 3.3.1. Section 3.3.2 focuses on role aspects. Additionally to observed advantages of considered role features, this section details the meaning of some of them in this application domain. Section 3.3.3 summarizes observed advantages regarding the integration of the component orientation for developed applications.

3.3.1. Software

In the case of basic applications, no specific code (i.e. glue code) has been required for the various compositions. Moreover, the code has not been modified to reuse roles from one application to another. Only specific XML descriptions (root subjects) have been written, in order to appropriately combine and configure roles as well as base-objects, all of them being transparently provided by components.

In the case of the advanced VR-Render software, roles developed for basic applications have been reused. No adaptation (i.e. (partial) rewriting of the code) has been necessary to meet the specific functional requirements of VR-Render. Only few additional concrete roles have been specifically developed, concerning some specific gui elements. For instance, an image selector panel has been developed for choosing the medical image to be visualized among a set of loaded images (provided by patients). The amount of specific code (specific roles) appeared almost negligible compared to the code reused (the one developed for basic applications).

```
<object type="Object">
 <object id="input0" type=Image >
  <role type="IRender" implementation="Mpr" com="yes" win="900"/>
   ....
 </object>
 </object id="input1" type="Float" />
 ....
 <role id="algo" type="IProcessor" implementation="Erode" >
    <InputImage id="input0"/>
    ....
 </role>
 <role type="IEditor" implementation="Editor" com="yes" win="901"/>
 <role type="IAction" implementation="Action" target="algo" ... />
 <role type="IAspect" implementation="Aspect" >
   <win id="900" ... />
   <win id="901" ... />
   ...
 </role>
 <start     type="IAspect" />
 ...
 <start     type="IRender" />
 <stop      type="IRender"/>
 ...
 <stop      type="IAspect"/>
</object>
```

Fig. 17. Example combining a graphical user interface (gui) with reading, rendering and processing functionalities: subject declaration.

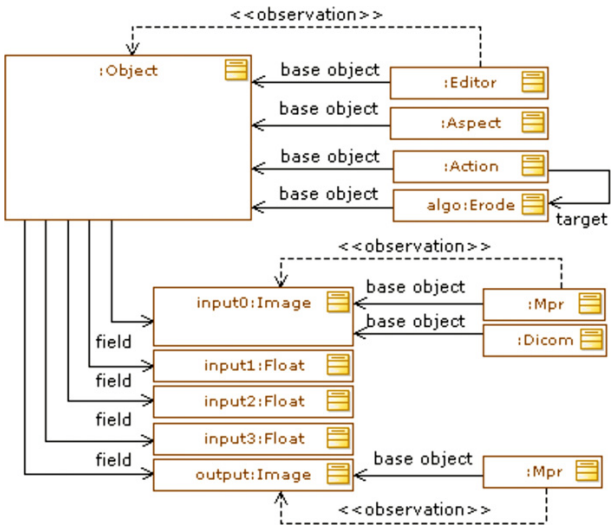

Fig. 18. Example combining a graphical user interface (gui) with reading, rendering and processing functionalities: subject instance.

To summarize, many code elements has been reused from one software to another, with few specific developments. In particular, no glue code has been required to built software programs, as only XML-based description files have been used. Although a quantitative evaluation of the degree of reuse would be interesting (e.g. amount of code reused from one application to another), the variety and the functional complexity of generated software programs underline the overall relevance of the proposed pattern for composition and collaboration.

3.3.2. Role

Observed advantages are discussed for each role concepts introduced in Section 2. The meaning of some of them for the considered application domain is also underlined.

3.3.2.1. Identity. This feature enabled to associate data structures and functionalities using a single reference on one entity only (data structure), without requiring a dedicated container which would be in charge of handling functionalities independently from the (base) objects to be processed. This facilitated the use of simple description files declaring subjects in a simple tree-like structure (functionalities appearing as a simple list of sub-elements of base-objects). Besides, the fact that each role keeps its own identity has been useful for explicit chaining with coordination roles retrieving collaborating roles using their identifier. The use of dynamic attributes appeared crucial to prevent unwanted inter-relations between functionalities (limited to dynamic attributes), and to ensure a good separation of concerns.

3.3.2.2. Dependency. This feature enabled to preserve the traditional separation between data (base-objects) and functionalities (roles) (Szyperski, 2002; Sridhar, 2007), although the composition result was seen and manipulated as a single entity. It also ensured the implicit detachment and destruction of functionalities attached to a given base-object, without requiring glue code. Besides, as roles directly depend on their base-objects, functionalities have a direct access to data structure, which appeared useful in terms of runtime performances, this being crucial in our application domain. For medical image and video processing as well as for rendering purposes, this regards the fact that the data buffer (of images or videos) can be directly accessed. As base-objects do not depend on functionalities (no a priori knowledge about possible functionalities to be attached to base-objects), any new initially unforeseen functionality has been integrated without requiring the modification of data structures (code rewriting).

3.3.2.3. Dynamicity. This feature enabled to achieve dynamic composition and decomposition of functionalities. In some cases, dynamic composition appeared similar to load-time composition (or early run-time composition), as for root subject instantiation (software instantiation). Nevertheless, real run-time composition and decomposition has also been considered. It has been used for processing, rendering or reading functionalities which could be interactively changed by the end-user after software instantiation (i.e. changing the processing, reading or rendering role). For instance, the reading algorithm could be changed each time the end-user decided to read a new image or video, by pressing the appropriate menu item (typically the "*File/Open*" menu). The list of available reading roles for the considered data structure (e.g. image or video) was first proposed to the end-user. The concrete role was then selected by the end-user, instantiated and finally attached to the appropriate image before being run. Note that any already attached reading role was removed (decomposition) if different from the selected one.

3.3.2.4. Multiplicity. This feature has been widely used in developed applications, in particular for assigning multiple similar concrete rendering roles to the same image, each one being specifically configured (e.g. example related to Fig. 14).

3.3.2.5. Activation. This feature has been used for all applications, as for the initialization of gui-based roles (e.g. preparation of a window or installation of menu items) and rendering-based ones (allocation of some resources for 3D rendering). It has also been used for temporarily enabling/disabling rendering roles depending on windows which could be hidden or shown according to end-user interactions (as illustrated for VR-Render). For instance, the `Mpr` rendering was disabled if the window was hidden (not visible by the end-user). In such a case, any attempt for refreshing the window content (invocation of `update` or `update(message)` methods) was ignored, therefore saving computation time and/or avoiding system failure. This feature also facilitated the management of the control flow at software instantiation time or when quitting applications. For instance, when quitting an application, systematically deactivating roles (in the appropriate order, as illustrated by the example related to Fig. 17) before detaching and destroying them preserved from unforeseen behaviour or software failure (e.g. deleting a role managing a window while a rendering role using it still receives refreshing requests).

3.3.2.6. Migration. This feature has been used in VR-Render to manage image selection changes, involving the migration of attached roles to the newly selected image. Migration appeared useful for rendering roles in particular, to avoiding losing their state (e.g. rendering windows within which renderings were performed) as well as to save computation time (no need to reallocate rendering resources).

3.3.2.7. Explicit chaining. This feature appeared useful to explicitly control the collaboration between roles for some specific functionalities. For instance, it has always been used to trigger the execution (invocation of the `update()` method) of particular roles (e.g. processing, reading or rendering) from buttons and menu items (being themselves roles). Such gui controls were seen as coordination roles (e.g. Fig. 18 ("Action")). Explicit chaining also appeared a relevant alternative for ordering particular sequences of tasks avoiding indirections which would result from observations (*implicit chaining*),

in order to ensure a limited response time. For `basicRobot` (see Fig. 9) in particular, explicit chaining enabled to completely control the execution flow (video tracking and command application) so that the control loop cycle was shorter than 40 ms (Fasquel et al., 2009). In such a case, a coordination role was in charge of the control loop, triggering video rendering only if enough time was remaining.

3.3.2.8. Implicit chaining. This feature appeared absolutely required to manage collaboration between functionalities in case of unforeseen end-user interactions as well as in case of unforeseen compositions encountered in considered software programs. This has been observed for the collaboration of rendering functionalities with reading and processing ones (illustrated in Fig. 18), the number of rendering roles involved in the composition being a priori unknown (unforeseen composition). This also appeared convenient for the collaboration, according to (unforeseen) end-user interactions, between several rendering roles (synchronization illustrated in Fig. 14).

3.3.2.9. Abstractivity. At `Role`-level, this feature appeared crucial to manipulated roles (i.e. functionalities) in an abstract manner. For instance, this facilitated the development of an application using a simple description file, involving a limited number of key-works, corresponding, in particular, to the well-identified (i.e. mapped on well-identified role concepts) and limited number of methods provided by this `Role` entity.

At `ConcreteRoleType`-level, this facilitated the manipulation of groups of roles, corresponding to concrete functionalities classified into functional groups. Thanks to a central container registering all instantiated subjects (and therefore functionalities), it was possible to access and interact with all functionalities of a given type. For instance, this enabled to activate all rendering roles only after those managing windows (e.g. Fig. 17 ("IAspect")) within which renderings were expected to be performed, regardless their related data structure (rendering roles being attached to two distinct images in Fig. 16).

3.3.2.10. Transitivity. This feature has been used to configure image readers (being a role) using a graphical editor (i.e. selection of filenames to be read) acting as a gui-based edition role (similar to Fig. 18 ("Editor")) attached to the reading role.

3.3.3. Component

The integration of the component orientation, coupled with role concepts, appeared useful for the reduction of build-level dependencies, as any new functionality or functionality type has been incorporated without entirely recompiling and redeploying an application.

Thanks to the graph-based component description in particular, one remained focused on functional composition (i.e. composition of base-objects and roles, corresponding to data structures and functionalities) regardless the specificity of the distribution of both data structures and functionalities over the different components. For load-time (or early run-time) composition, subject declarations did not explicitly indicate components to use and wire: only base-objects and roles were reported (see Figs. 13(a), 15(a) and 17). Required components were automatically and transparently retrieved through the analysis of the graph resulting from their descriptions. For run-time composition, this also appeared useful to integrate new functionalities without taking care of the identity and the location of components providing them. In the case of reading functionality previously discussed, the graph enabled to easily propose to the end-user the list of reading role available for images for instance: this list was automatically built from graph analysis by searching for nodes connected simultaneously to both `IReader` and `Image` nodes. After selection, the appropriate component was started if necessary, to finally instantiate and attach the concrete role.

4. Related works

Hereafter, we propose to compare our proposal to some well-known approaches addressing issues considered in this paper. Considered alternatives are those mentioned in Section 1, namely those based on aspect oriented programming (AOP), role, mixin and delegation. Section 4.1 focuses on the composition issue. The collaboration issue is discussed in Section 4.2. Section 4.3 is dedicated to roles and Section 4.4 focuses on the reduction of build-level dependencies through the use of components. Table 2, the content of which being detailed hereafter, aims at providing an overview of the proposed comparison.

4.1. Composition

For comparison purposes, composition is considered in terms of composition type (i.e. composition at class-level or object-level), composition time (i.e. run-time or compile-time) and composition context. Composition context regards the context of use of the composition for the different alternatives.

4.1.1. Composition type

Class-level composition involves that all instances of a given class are concerned by composition, compared to object-level composition focusing on instances. The proposed design pattern considers composition at object-level. For instance, in our application domain, a rendering or a reading role is attached to an image, and not to all images. Table 2 reports the type of composition favoured by the different considered related works, this being detailed hereafter.

As recently underlined in Bettini et al. (in press), aspect oriented programming (AOP) focuses on class-level composition. This paradigm, representing an active research area (Tanter et al., 2008), aims at inserting some code into class methods of a target program, the position in the target program being defined by *jointpoints* (Bettini et al., in press; Tartler et al., 2010). The inserted code is an *advice* and a *pointcut* identifies or captures *jointpoints* in the program flow. *Advices* and *pointcuts* are declared using *aspects*, and code insertion corresponds to *aspect weaving*. AOP is widely used for cross-cutting concerns (i.e. common tasks independent from the business logic) such as logging, monitoring, persistence and security purposes (Villazon et al., in press; Fuentes and Sanchez, 2007).

As recently underlined in Baldoni et al. (2007), mixin-based approaches mainly focus on class-based composition (mixing separated class hierarchies), rather that on object-base one, although the ability to consider composition at both object-level and class-level has been recently studied (Zdun et al., 2007).

Approaches based on roles (Herrmann, 2007; Baldoni et al., 2007) and delegation (Gamma et al., 1995; Lieberman, 1986; Bettini et al., in press) focus on composition at object level. As an example of delegation-based object composition approach, a recent proposal introduces the notion of *incomplete objects*, which focuses on the dynamic assignment of method implementation to objects (objects becoming *complete* after composition) (Bettini et al., in press). Delegation-like design patterns such as *decorator* aims to changing object method behaviour through dynamic composition (Gamma et al., 1995). Recent role-based proposals such as *ObjectTeams/Java* (*OT/J* for short) (Herrmann, 2007) and *powerJava* (Baldoni et al., 2007) aim at modifying the behaviour of some existing object methods through role attachment (e.g. notion of *callin* in *OT/J* (Herrmann, 2007)).

Table 2
Comparison with some recent works (see references therein for earlier works). *"Out of scope"* means that, to our knowledge, no specific study has been performed to evaluate whether (or detail how) the approach could support the notion.

Notion	Approach				
	Proposed pattern	Role (Herrmann, 2007; Baldoni et al., 2007; Colman and Han, 2007)	AOP (Tanter et al., 2008; Fuentes and Sanchez, 2007; Bachara et al., 2010; Tizzei et al., in press)	Mixin (Zdun et al., 2007)	Delegation (Gamma et al., 1995; Lieberman, 1986; Bettini et al., in press)
Composition type (Section 4.1.1)	Object-level	Object-level	Class-level	Class-level (Class/Object-level; Zdun et al., 2007)	Object-level
Composition time (Section 4.1.2)	Run-time	Run-time	Run-time	Run-time	Run-time
Composition context (Section 4.1.3)	Adding behaviour (Entire program)	Changing behaviour	Changing behaviour (existing program)	Adding behaviour	Changing behaviour
Collaboration (Section 4.2)	Implicit/explicit	Explicit (team; Herrmann, 2007), (coordinator; Colman and Han, 2007)	Explicit (behaviour links; Tanter et al., 2008), (coordinator; Fuentes and Sanchez, 2007)	Explicit (mixin list; Zdun et al., 2007)	Out of scope
Role (Section 4.3)	Many supported concepts, advanced functionalities	Few supported concepts, many supported concepts (Herrmann, 2007), basic functionalities (Herrmann, 2007; Baldoni et al., 2007)	Out of scope	Out of scope	Out of scope
Build-level dependencies (Section 4.4)	Component	Out of scope	Component (Tizzei et al., in press)	Out of scope	Out of scope, component (Gamma et al., 1995)

4.1.2. Composition time

As underlined in Table 2, all considered related works support run-time (dynamic) composition (see e.g. Bettini et al., in press for delegation and Zdun et al., 2007 for mixins). AOP supports dynamic weaving, even though static weaving remains widely used (e.g. AspectJ extension of the Java language for AOP) (Bachara et al., 2010).

Recent proposals for role-based composition consider composition at run-time (Herrmann, 2007; Baldoni et al., 2007). Note that an earlier work considered the implementation of roles in C++ (VanHilst and Notkin, 1996), the language we consider in the case study. Their proposal was based on the use of the C++ template mechanism, therefore limited to compile-time composition (not reported in Table 2 as recent role alternatives consider run-time composition).

4.1.3. Composition context

This aspect regards the context of use, or the intention, of considered proposals with respect to dynamic composition.

In our sense, considered related works regard dynamic composition as a way to modify or provide the behaviour of existing method at class-level (e.g. AOP; Villazon et al., in press) or object-level (e.g. *incomplete object*, Bettini et al., in press; *OT/J*, Herrmann, 2007; and *PowerJava* Baldoni et al., 2007). In our case, one focuses on adding functionalities, initially unknown to the base-object. Practically, instead of changing the behaviour of, for instance, a `read` method which could have been declared for images, one favours the ability to dynamically add the notion of reading. In our case, base-objects have no knowledge about the kind of functionalities which could be attached to it after composition. An advantage is that it is not required to update the declaration of a base-object in order to integrate a new type of functionality. Note that, similarly, mixin-based approaches allow to add methods (mainly at class-level). The counterpart is that the behaviour of existing methods are not modified in our case, as it is not the intention of our proposal. This difference is reported in Table 2 (*Adding behaviour* and *Changing behaviour*).

In our opinion, a motivation of approaches focusing on the modification of existing methods is to adapt the behaviour (at object or class level) of these methods from the point of view of the client program invoking them. This clearly appears for AOP approaches focusing on cross-cutting concerns rather than on business logic. This underlines that AOP-based composition is used to extend an existing program (*existing program* in Table 2) and not to build an entire one. In our case, the entire program (*entire program* in Table 2) is concerned by the composition (the entire program being defined by the *root subject*).

In our sense, due to this difference regarding the composition context, considered related works appear complementary to our proposal.

4.2. Collaboration

In AOP-based approaches, collaboration regards the execution order of pieces of code (*advices*) injected at a given point of the target program. Controlling the chaining, and therefore interactions between advices woven to a given *jointpoint*, remains a challenging issue (Tanter et al., 2008). In approaches comparable to dynamic inheritance (Zdun et al., 2007), chaining corresponds to the navigation across object or class methods (notion of navigation path), according to the composition. For instance, in the case of *incomplete objects* (Bettini et al., in press), the implementation of some methods can be provided through composition and are therefore dynamically integrated in the navigation path. For mixin-based composition, navigation depends on methods dynamically added at class level, or at both class and object levels in Zdun et al. (2007). For role-based composition, the notion of chaining regards the coordination of roles (Herrmann, 2007; Colman and Han, 2007).

As underlined in Section 1, considered related works favour the management of collaboration using a dedicated entity. For instance, in AOP, a *behavioural link* has been recently proposed to control the program flow, according the configuration of links (e.g. activation condition) (Tanter et al., 2008). Another recently studied approach is to consider a specific aspect in charge of the coordination (Fuentes and Sanchez, 2007). For mixins, this issue has also been addressed by providing a mixin list (Zdun et al., 2007). The recent *OT/J* proposal, based on role concepts, manages role collaboration using a dedicated entity (Herrmann, 2007), the *Team* which encapsulate collaborating roles. In Colman and Han (2007), a specific role is in charge of the coordination.

In our case, as previously underlined, we propose to consider two collaboration modes (see Table 2). The first one regards the use of a coordination role, as in Colman and Han (2007) (*Explicit chaining*). The second collaboration mode corresponds to the notion of *implicit chaining*, being, in our sense, ignored by related works addressing dynamic composition, and in particular by role-based approaches. Note that a strength of our proposal is that both collaboration modes appear as first-class entities of the proposed pattern. For explicit chaining, this regards the `update()` method of the `Role` class together with a coordination role. For implicit chaining, this regards the `update(msg)` method of the `Role` class coupled with the observation link. A perspective is to facilitate the reuse of a given role in different collaboration modes. For instance, by removing the observation link of a given role with respect to its base-object (*implicit chaining* mode), the given role could be used in an *explicit chaining* mode, where coordination would be managed by a dedicated role, as in Colman and Han (2007).

It must be pointed that *implicit chaining* (similar to an *endogeneous coordination model*; Fuentes and Sanchez, 2007) may present a limitation resulting from the underlying observation mechanism. In such a case, roles manage both computations (i.e. the functionality (Fuentes and Sanchez, 2007), associated with the `update()` method of the `Role` class in the proposed pattern) and interactions (according to received messages, associated with the `update(msg)` method the `Role` class in the proposed pattern). As a consequence, any modification of the coordination protocol (e.g. changing the identifiers of dynamic attributes indicating base-object modifications in our case) may involve some rewriting of concrete roles (regarding the `update(msg)` method). In the case of *explicit chaining* (similar to an *exogeneous coordination model*; Fuentes and Sanchez, 2007), only the coordination role has to be modified, therefore preserving concrete roles.

4.3. Role

As recently underlined (Herrmann, 2007), there is no well accepted standard for applying roles to a programming language, as it relies on concepts (Kristensen and Osterbye, 1996; Steimann, 2000) rather than on well established implementation constraints or rules. Therefore, a difficulty is to find an appropriate mapping between these high-level concepts and low-level programming languages, as addressed in our case with the proposed design pattern. A consequence is that, as underlined in Herrmann (2007), almost all existing implementation proposals focus on some concepts while ignoring others (see recent papers Herrmann, 2007; Baldoni et al., 2007, and references therein).

As underlined in Table 2, we claim that a strength of the proposed pattern regards the fact that we support many role concepts. In our sense, the set of supported concepts is comparable to those considered in the recent *OT/J* proposal which covers more aspects than previous proposals (Herrmann, 2007) (*Few supported concepts*, except for Herrmann, 2007 in Table 2).

Concepts being supported in our case as well as by *OT/J* are *identity*, *dependency*, *dynamicity*, *multiplicity*, *abstractivity* and *transitivity*. Notions of *activation* and *chaining* are associated with the notion of collaboration managed by the *Team* entity in *OT/J*. A first difference regards the notion of *activation* which has a different meaning in our case, as underlined in Section 2.2.1. A second difference concerns the concept of *visibility*, regarding the ability to restrict the access to some roles (Kristensen and Osterbye, 1996). This feature is supported by *OT/J* but not in our case because all attached roles can be accessed without any restriction. A third difference regards the *migration* feature, being supported in our case but not by *OT/J*. In our sense, a major limitation of *OT/J* regards the fact that roles are hard-coded inner classes of *Team* entities. This does not facilitate the reuse of a role in different contexts (a given role being constrained to a particular *Team*). This also affects the ability to easily support the *multiplicity* feature, as underlined in Herrmann (2007), roles of similar types and attached to the same base-object instance must be provided by different contexts (i.e. different *Team* instances). Note that differences between our proposal and *OT/J* are not detailed in Table 2 because it aims at providing an overview not limited to role concepts. For clarity, only the overall degree of coverage of role concepts is reported in Table 2 (i.e. many or few supported concepts), details being provided in this section.

Among other recent alternatives supporting few role concepts (Herrmann, 2007), *powerJava* (Baldoni et al., 2007) breaks the *dependency* feature at code level, because roles are declared as inner classes of the base-object, as also underlined in Herrmann (2007). The recent *ROAD* proposal (Colman and Han, 2007) focuses on collaboration, rather than on other concepts, using a dedicated role, similarly to our proposal of *explicit chaining* (discussed in the previous Section 4.2), and does not suffer the limitation of *Team* as roles are independent for the coordination role.

In our sense, additionally to the good coverage of role concepts, another contribution of our proposal, compared to related role-based approaches, regards the refinement of the notion of collaboration by considering two complementary modes (discussed in previous Section 4.2).

Another strength of our pattern is that several concepts are considered as explicit first-class entities of the pattern. This regards the fact that the `Role` class defines an interface (to be implemented by entities related to both `ConcreteRoleType` and `ConcreteRoleType` in Fig. 1) consisting in five abstract methods corresponding to well identified role concepts (*activation*, *implicit chaining*, *explicit chaining* and *migration*). This guides the developer in declaring new concrete role types as well as new concrete roles, while respecting associated role concepts appearing as contract to be fulfilled (interface to be implemented). This also facilitates the manipulation of roles in an abstract manner, through the concepts related to `Role` class methods regardless the specificity of role types and concrete roles.

Note that another interesting aspect of our work is the proposed mapping between some role concepts and a particular application domain which has never been considered in related works regarding roles. It illustrates both the meaning and the relevance of role concepts in a new practical situation, involving complete software programs with advanced functionalities (*Advanced functionalities* in Table 2) compared to related works such as (Herrmann, 2007; Baldoni et al., 2007), limited to basic use cases (*Basic functionalities* in Table 2).

4.4. Build-level dependencies

As reported in Table 2, coupling dynamic composition and component has been considered in the case of AOP, where aspects can be provided by a component to another (Tizzei et al., in press) (see references therein for other works coupling both notions). To our knowledge, there is no specific study focusing on the cou-

pling of component orientation with other dynamic composition approaches such as mixin and specific delegation alternatives. In Table 2, we only consider that approaches based on standard design patterns such as *decorator* (Gamma et al., 1995) have been widely used for software development involving components.

To our knowledge, there is no proposal coupling both role and component concepts. Nevertheless, it is interesting to mention the work of Parka et al. (2004), focusing on a methodology, coupling both notions, for the organization of the development of collaborative commerce systems. The goal is to reduce the development cycle time. The role perspective focuses on capturing business requirements for the c-commerce community business functions. The result is the logical-level specification of reusable components (for storing code elements). This work does not consider roles for composition purposes at programming level as in our case. For this reason, the coupling of roles with the issue related to the reduction of build-level dependencies using components is considered as *out of scope* in Table 2. Although out of the scope, their idea can also be found in our case. Indeed, the role orientation focuses on the composition of functionalities (at implementation level in our case), hiding the specificity of their (physical) distribution over components. In our sense, this observation underlines the relevance of coupling both role and component concepts, although the context of use differs from Parka et al. (2004).

5. Conclusion

The proposed design pattern offers a good coverage of role concepts for dynamic composition and collaboration purposes. Compared to related works, it integrates a refinement of the notion collaboration by considering two complementary approaches, depending on whether collaborations can be foreseen or not. The presented design pattern also proposes a solution for coupling both role and component concepts, where the component orientation aims at reducing build-level dependencies. As a result, the proposed design pattern proposes a solution for managing composition by focusing on functionalities only, components providing them being transparently retrieved.

The pattern has been successfully implemented in C++ and used to develop a set of medical software programs, involving many complex and advanced functionalities. Each role concept appeared perfectly appropriate to manage considered functionalities. The proposed coupling with the component orientation enabled to build an entire software using a generic program, independent from dynamic libraries embedding data structures and functionalities, together with description file focusing on functionalities regardless the specificity of the component distribution.

The perspective and future steps of that work is to further evaluated the relevance of the proposed design pattern for managing a reliable software product line, involving even more advanced medical software programs, as well as for other application domains.

Acknowledgements

A part of this work has been achieved in the context of the eHealth PASSPORT project of the 7th Framework Program of the European Community, funded by the ICT program.

We would like to thank all the research and development teams of IRCAD. In particular, we thank Guillaume Brocker and Vincent Agnus for their technical contributions to the implementation of the proposed design pattern. We also thank all medical partners who have provided us the medical images of patients, and in particular, Pr. Afshin Gangi and Pr. Catherine Roy from the radiological department of the University Hospital of Strasbourg. Finally, we thank the entire digestive and endocrine surgical team of Prof. Marescaux and in particular Prof. Didier Mutter, Prof. Bernard Dallemagne, Prof. Jol Leroy and Dr. Michel Vix.

Authors thank reviewers for their helpful suggestions regarding concepts and models.

References

Bachara, P., Blachnicki, K., Zielinski, K., 2010. Framework for application management with dynamic aspects J-EARS case study. Information and Software Technology 52, 67–78.

Baldoni, M., Boella, G., van der Torre, L., 2007. Interaction between objects in power Java. Journal of Object Technology 6, 5–30.

Bass, L., Clement, P., Kazman, R., 1998. Software Architecture in Practice. Addison-Wesley.

Bettini, L., Bono, V., Venneri, B., Delegation by object composition, Science of Computer Programming, in press.

Brown, M., McNitt-Gray, M., 2000. Handbook of Medical Imaging, vol. 2, Spie Press.

Colman, A., Han, J., 2007. Using role-based coordination to achieve software adaptability. Science of Computer Programming 64, 223–2485.

Czarnecki, K., Eisenecker, U., 1999. Generative Programming: Methods Techniques and Applications. Addison-Wesley.

DiMaio, S., Kapur, T., Cleary, K., Aylward, S., Kazanzides, P., Vosburgh, K., Ellis, R., Duncan, J., Farahani, K., Lemke, H., Peters, T., Lorensen, W.B., Gobbi, D., Haller, J., Clarke, L.L., Pizer, S., Taylor, R., Fichtinger, R.G.G., Hata, N., Lawson, K., Tempany, C., Kikinis, R., Jolesz Jr., F., 2007. Challenges in image-guided therapy system design. NeuroImage 37, 144–151.

Duncan, J., Ayache, N., 2000. Medical image analysis: progress over two decades and the challenges ahead. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (1), 85–106.

Fasquel, J.-B., Waechter, J., Goffin, L., Nicolau, S., Agnus, V., Soler, L., Marescaux, J., 2008. A XML based component oriented architecture for image guided surgery: illustration for the video based tracking of a surgical tool. In: Insight Journal, Workshop on Systems and Architectures for Computer Assisted Interventions, 11th International Conference on Medical Image Computing and Computer Assisted Intervention, http://hdl.handle.net/10380/1497.

Fasquel, J.-B., Chabre, G., Zanne, P., Nicolau, S., Agnus, V., Soler, L., De Mathelin, M., 2009, M.J. A role-based component architecture for computer assisted interventions: illustration for electromagnetic tracking and robotized motion rejection in flexible endoscopy. In: Insight Journal, Workshop on Systems and Architecture for Computer Assisted Intervention, the 12th International Conference on Medical Image Computing and Computer Assisted Intervention, http://hdl.handle.net/10380/3069.

Fuentes, L., Sanchez, P., 2007. Aspect-oriented coordination. Electronic Notes in Theoretical Computer Science 189, 87–103.

Gamma, E., Helm, R., Johnson, R., Vlissides, J., 1995. Design Patterns. Addison-Wesley.

Herrmann, S., 2007. A precise model for contextual roles: the programming language ObjectTeams/Java. Applied Ontology 2, 181–207.

International Organization for Standardization, 2010. ISO/IEC 9126–1:2001, Software Engineering – Product quality – Part 1: Quality model. http://www.iso.org/.

Kristensen, B., Osterbye, K., 1996. Roles: conceptual abstraction theory and practical language issues. Theory and Practice of Object Systems 2, 143–160.

Lau, K.-K., Wang, Z., 2007. Software component models. IEEE Transactions on Software Engineering 33 (10), 709–724.

Lieberman, H., 1986. Using prototypical objects to implement shared behavior in object oriented systems. In: Proceedings of the Conference on Object-oriented Programming Systems, Languages, and Applications , pp. 214–223.

Parka, H., Suhb, W., Lee, H., 2004. A role-driven component-oriented methodology for developing collaborative commerce systems. Information and Software Technology 46, 819–837.

Pratt, W.K., 2007. Digital Image Processing, 4th edition. Wiley-Interscience.

Sridhar, N., 2007. Serfs: dynamically-bound parameterized components. Journal of Systems and Software 80, 736–749.

Steimann, F., 2000. On the representation of roles in object-oriented and conceptual modelling. Data and Knowledge Engineering 35, 83–106.

Szyperski, C., 2002. Component Software: Beyond Object-oriented Programming, 2nd edition. Addison-Wesley.

Tanter, E., Toledo, R., Pothier, G., Noyeb, J., 2008. Flexible metaprogramming and AOP in java. Science of Computer Programming 72, 22–30.

Tartler, R., Lohman, D., Scheler, F., Spinczyk, O., 2010. AspectC++: an integrated approach for static and dynamic adaptation of system software. Knowledge-Based Systems 23, 704–720.

Taylor, R., Stoianovici, D., 2003. Medical robotics in computer-integrated surgery. IEEE Transactions on Robotics and Automation 19 (5), 765–781.

Tizzei, L., Dias, M., Rubira, C., Garcia, A., Lee, J., 2010. Components meet aspects: assessing design stability of a software product line. Information and Software Technology 53 (2), 121–136.

VanHilst, M., Notkin, D., 1996. Using role components to implement collaboration-based designs. In: Proceedings of the Conference on Object-oriented Programming Systems, Languages, and Applications , ACM Press, pp. 359–369.

Villazon, A., Binder, W., Moret, P., Ansaloni, D., Comprehensive aspect weaving for java, Science of Computer Programming, in press.

Zdun, U., Strembeck, M., Neumann, G., 2007. Object-based and class-based composition of transitive mixins. Information and Software Technology 49, 871–891.

Zdun, U., 2006. Tailorable language for behavioral composition and configuration of software components. Computer Languages, Systems and Structures 32, 56–82.

Jean-Baptiste Fasquel was born in 1975. He received his diploma from the general engineering school ENSPS in France, and his diploma for advanced studies in applied optics and computer vision from the University of Strasbourg in 1998. He obtained his PhD in computer science from the University of Strasbourg, France, in 2002. From 2002 to 2009, he has been a permanent researcher in computer sciences at the Research Institute Against Digestive Cancer IRCAD, Strasbourg. Since September 2009, he has been an Associate Professor at the University of Angers. His main areas of interest are software engineering and medical image processing for computer-aided diagnosis and surgery.

Johan Moreau received his M.S. in computer science from the University of Rennes in 2001. Since 2001, he has been a permanent engineer in computer sciences at the Research Institute Against Digestive Cancer IRCAD, Strasbourg. His main areas of interest are network and software engineering for computer-aided diagnosis and surgery.

Optical Engineering 46(10), 108202 (October 2007)

Optical implementation of the filtered backprojection algorithm

Morgan Madec
Institut d'électronique du Solide et des Systèmes
23, rue du Loess
67037 Strasbourg, France
and
Micro Module
38, rue Jim Sévéllec
29200 Brest, France

Jean-Baptiste Fasquel
Hôpital Civil
Institut de Recherche sur le Cancer de l'Appareil Digestif
67091 Strasbourg, France

Wilfried Uhring
Institut d'électronique du Solide et des Systèmes
23, rue du Loess
67037 Strasbourg, France

Pascal Joffre
Micro Module
38, rue Jim Sévéllec
29200 Brest, France

Yannick Hervé
Institut d'électronique du Solide et des Systèmes
23, rue du Loess
67037 Strasbourg, France

Abstract. Filtered backprojection (FBP) is the basic operation of image reconstruction algorithms in tomography. It is widely used but very time-consuming. We propose a new implementation, based on an optoelectronic architecture, providing a speedup of about two orders of magnitude over a classical digital implementation. The realization of the optical core, based on a rotated Dove prism, requires careful attention in order to ensure good image quality. This aspect has been studied in simulation with a suitable model of the architecture, and in practice with an experimental setup. Results are very encouraging. © *2007 Society of Photo-Optical Instrumentation Engineers.* [DOI: 10.1117/1.2786450]

Subject terms: optical processing; tomographic image reconstruction; filtered backprojection.

Paper 070117R received Feb. 7, 2007; revised manuscript received Mar. 23, 2007; accepted for publication Mar. 23, 2007; published online Nov. 1, 2007.

1 Introduction

The Radon transform (RT) is the fundamental mathematical tool of computerized tomography (CT). For most imaging systems using x-ray or nuclear emission techniques, the final image can be reconstructed by computing the inverse of the Radon transform on the acquired data. This operation can be carried out through many methods, but the filtered backprojection (FBP) algorithm seems to be the most commonly used.[1]

The FBP algorithm can be decomposed into two steps: the filtering of the acquired data, and the backprojection. This operation is very time-consuming, with a complexity of $O(N^{3/2})$, as discussed in Sec. 3.2, where N is the number of points in the reconstructed image. The huge computation power needed to perform this operation will probably be a strong limitation on the development of future real-time applications such as image-guided robotic surgery.[2,3] Some clever methods have been introduced in order to reduce the complexity of backprojection. One of the most famous ones is the linogram method, developed in 1987.[4] Recently, some other methods, based on the divide-and-conquer principle, have been published,[5–7] and in 2003, a new inversion method based on the wavelet transform was introduced.[8] Those methods offer lower computation time at the cost of a decrease of processing quality.

In this paper, we choose to keep the original formulation of the FBP and try to improve the hardware. Backprojection and filtering are both attractive operations for optical implementation. Filtering can be performed through a classical optical architecture such as the Vander Lugt optical correlator.[9] Some optical implementations of the backprojection have already been demonstrated. In 1978, Nishimura et al. put forward an architecture adapted to an alternative formulation of the FBP using the Hilbert transform.[10,11] Due to its complexity, such a system is nowadays obsolete. In 1995, an American patent[12,13] described an optical system designed for the implementation

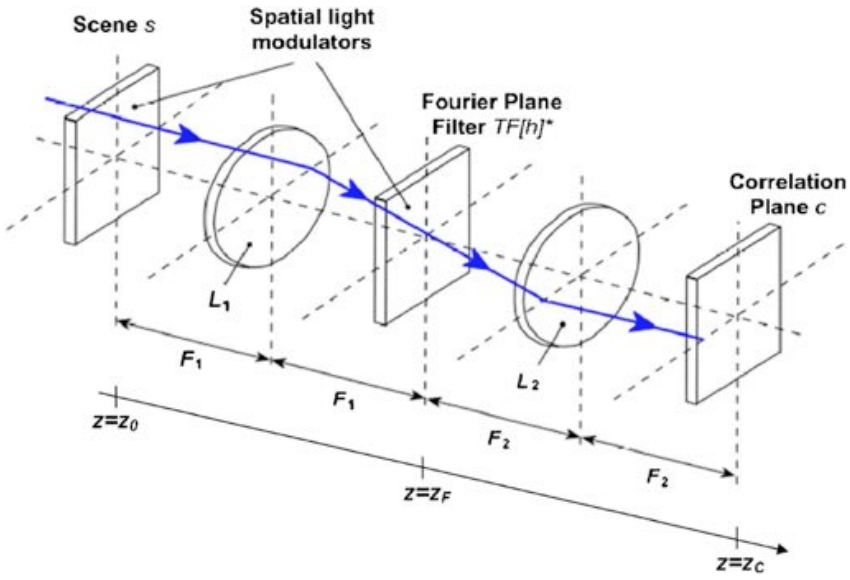

Fig. 1 The 4f Vander Lugt correlator.

of iterative reconstruction algorithms like ART and PIRT.[1] For our purpose, however, the iterative nature of this implementation precludes fast reconstruction, and it is therefore unsuitable.

Our system, based on the FBP algorithm, combines a filtering core with an optical backprojection core using a rotated Dove prism. The presentation begins with a summary of the FBP algorithm, before describing the proposed optical architecture. Then, run-time aspects are analyzed. In the last section, image quality aspects are considered, first in simulation, and then with experimental results. Finally, the results are used to discuss the potential of our architecture.

2 The FBP Algorithm

The RT of a 2-D function is the set of integrals of the function along all the directions of the plane. A point of the RT is defined by two coordinates: The direction of the integration (angle θ) and the distance between the integral line and the center of the system (radial coordinate ρ):

$$R\{f\}(\theta,\rho)=\int_{x=-\infty}^{+\infty}\int_{y=-\infty}^{+\infty} f(x,y)\cdot\delta(\rho-x\cos\theta - y\sin\theta)\mathrm{d}x\,\mathrm{d}y, \quad (1)$$

where $\delta(u)$ is the 1-D Dirac function. This expression can be inverted by considering the Fourier-slice theorem, which states that the 1-D Fourier transform of the projection function $R\{f\}(\theta,\rho)|_{\theta=\varphi}$ is equal to the 2-D transform of the image f evaluated on the line that the projection was taken on (the line whose equation is $x\cos\varphi+y\sin\varphi=0$):

$$F\circ R\{f\}(\varphi,\nu_\rho)=F_2\{f\}(\nu_x,\nu_y)\big|_{\substack{\nu_x=\nu_\rho\cos\varphi\\ \nu_y=\nu_\rho\sin\varphi}}. \quad (2)$$

Let us consider the 2-D inverse Fourier transform of f:

$$f(x,y)=\int_{\nu_x=-\infty}^{+\infty}\int_{\nu_y=-\infty}^{+\infty} F_2\{f\}(\nu_x,\nu_y)\exp[2i\pi(\nu_x x+\nu_y y)]\mathrm{d}\nu_x\,\mathrm{d}\nu_y. \quad (3)$$

With a polar coordinate system and the Fourier-slice theorem, we can write

$$\begin{aligned} f(x,y)&=\int_{\theta=0}^{2\pi}\int_{\nu_\rho=-\infty}^{+\infty} F_2\{f\}\\ &\times(\nu_\rho\cos\theta,\nu_\rho\sin\theta)\exp[2i\pi\nu_\rho(\cos\theta x+\sin\theta y)]\\ &\times|\nu_\rho|\mathrm{d}\nu_\rho\,\mathrm{d}\theta=\int_{\theta=0}^{2\pi}\int_{\nu_\rho=-\infty}^{+\infty} F\circ R\{f\}(\theta,\nu_\rho)\\ &\times|\nu_\rho|\exp(2i\pi\nu_\rho\rho)\mathrm{d}\nu_\rho\,\mathrm{d}\theta. \end{aligned} \quad (4)$$

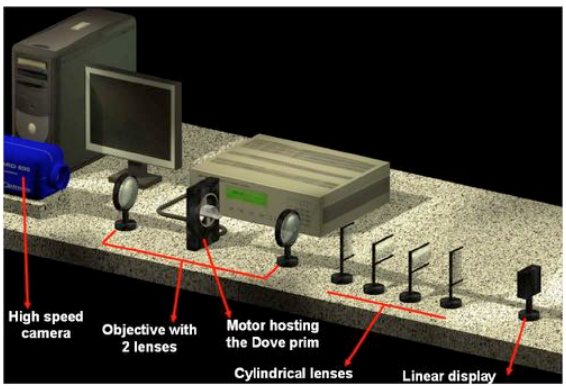

Fig. 2 The optical backprojection core.

This expression corresponds to the basic equation of the FBP algorithm. We can recognize the projection filtering step,

$$V{:}\mathbb{R}^2 \to \mathbb{R}^2, \quad f(\rho,\theta) \to f(\rho,\theta) * h(\rho,\theta) \quad \text{with } F\{h\}(\nu_\rho,\theta) = |\nu_p|, \tag{5}$$

and the backprojection operator,

$$B{:}\mathbb{R}^2 \to \mathbb{R}^2, \quad f(\rho,\theta) \to B\{f\}(x,y) = \int_0^{2\pi} f(x\cos\theta + y\sin\theta,\theta)\mathrm{d}\theta. \tag{6}$$

3 Proposed Optical Implementation

3.1 *Filtering*

Filtering of projections can be obtained through a Vander Lugt optical correlator.[9] This setup uses two spatial light modulators (one to display the image and the second one to display the filter), two thin lenses, a camera to acquire the filtered image, and a coherent light source (see Fig. 1).

The image acquired by the camera is the squared modulus of the convolution integral between the image generated by the first display and the filter whose Fourier transform is displayed on the second display. This is a standard result in optical processing:

$$g(-x,-y) = |f(x,y) * h(x,y)|^2. \tag{7}$$

3.2 *Backprojection*

The backprojection operator is generally implemented on a digital processor through the three-loop algorithm described in the following:

```
Algorithm: BP_classic
  For each line (m=1 to M),
     For each column (n=1 to N),
        For each projection (l=1 to L),
          a=i*cos(l)j=i*sin(l),
          F(m,n)=F(m,n)+Rf(a,l)
        End
     End
   End
End Algorithm
// Rf is the Radon transform of f. F is the reconstructed image
// associated with Rf
```

The numbers of rows, columns, and projections are of the same order of magnitude. For this reason, this algorithm is said to be $O(LMN)$ (where L is the number of points per projection, N the number of columns in the reconstructed image, and M the number of rows) or $O(A^{3/2})$, where A is the number of pixels in the reconstructed matrix (number of rows times number of columns).

An alternative to this implementation is to consider a

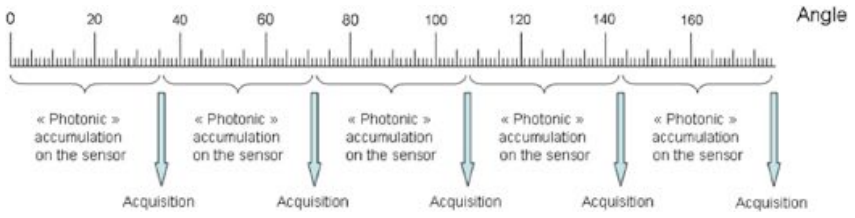

Fig. 3 Illustration of the hybrid accumulation system. Each graduation of the ruler corresponds to a subframe to add. In this example, K=180 and N_{acq}=5. There are 36 photon accumulations before each acquisition.

geometrical interpretation of the equation (6). In such a case, the reconstructed image can be seen as the sum of subframes constituted by one filtered projection extended in the perpendicular direction and rotated by the angle that the projection was taken at. Let us consider the equation of one subframe in the coordinate system (O,x',y') obtained by rotating the original one (O,x,y) through an angle θ:

$$f_\theta(x',y') = V \circ R(x',\theta). \tag{8}$$

In the original coordinates, one can write

$$x' = x\cos\theta + y\sin\theta \quad \Rightarrow$$
$$f_\theta(x,y) = V \circ R(x\cos\theta + y\sin\theta,\theta), \tag{9}$$

where $f_\theta(x,y)$ is the integral term of the FBP expression. This way to compute the backprojection is summarized by the following algorithm:

```
Algorithm: BP_alternative
  For each projection (k=1 to K),
    For each point of the projection (l=1 to L),
      Temp (1 to l,l)=Rf(l,k)
      Rotate (Temp, k)
      F=F+Temp
    End
  End
End Algorithm
```

This implementation can be achieved by the optical setup described on Fig. 2. It is composed of a noncoherent uniform light source, a linear spatial light modulator (to display the projection), a beam expander, an image rotator, and an acquisition system. The beam expander is composed of cylindrical lenses, whereas the image rotator consists of a Dove prism mounted on a motor. It should be noticed that the linear SLM and the expander can be replaced by an array SLM. However, the passive display system coupled with the noncoherent light source can also be replaced by an active system (an LED array, for example).

4 Potential of the Optical Architecture

In this section, we aim at evaluating the potential, in terms of computation power, of the optical architecture presented in the previous section. We introduce the four following parameters: K is the number of points per projection, L is the number of projections, and M and N are the numbers of rows and columns in the reconstructed image.

First, we focus on the filtering processor. We consider that:

- Display of image i and readout of image $i-1$ are performed in parallel.
- The filter is fixed for a large sequence of images, so that its display rate can be neglected.
- The exposure time can be neglected thanks to the use of high-power light sources.

In this condition, the computation time for one filtering is given by the longer of the readout time t_{rea} and the image display time t_{disp-m}:

$$t_{filt} = \max(t_{disp-m}, t_{rea}). \tag{10}$$

This result remains true as long as the resolution of the optoelectronic components is sufficient in comparison with K and L.

Second, the backprojection processor is analyzed. We consider that:

- Display and readout of camera and rotation of the motor are performed in parallel.
- The exposure time can be neglected.
- A hybrid accumulation system is used: The L subframes are accumulated optically by packets on the sensor, and then the packets are digitally added by postprocessing. The number of packets is N_{acq}, and each one contains L/N_{acq} subframes (see Fig. 3). The method of accumulation allows one to reach a compromise between the computation time and the dynamics of the reconstructed image (both parameters increase with N_{acq}). This point is discussed in the next section.

In this case, the computation time for the backprojection core is given by

$$t_{bp} = \max(Lt_{disp-l}, t_{mot}, N_{acq}t_{acq}), \tag{11}$$

where t_{disp-l} is the display time for a linear spatial light modulator and t_{mot} is the rotation time of the motor. This

Table 1 Typical performance of up-to-date optoelectronic and electronic equipment.

Description	Symbol	Value	Remarks
High-speed array LCD screen (FLC)	t_{disp_m}	1 ms	512^2, Boulder Nonlinear Systems[14]
High-speed array SLM (MQW)	t_{disp_m}	20 μs	256^2, Lenslet EnLight 256[15]
High-speed linear VCEL display	t_{disp_l}	8 ns	256 points, Lenslet EnLight 256[15]
Motor rotation: time speed	V_{mot}	2400 rpm	Newton, with a reduction ratio of 20[16]
	t_{mot}	6.25 ms	A full backprojection requires a quarter of revolution.
High speed CMOS sensor	$T_{rea\text{-}512}$	200 μs	Micron Web site[17]
	$T_{rea\text{-}1k}$	2 ms	Micron MT9M413 sensor[17]
Filtering on DSP	$t_{filt\text{-}1k}$	15.64 μs	Analog Devices TS201[18]
	$t_{filt\text{-}2k}$	34.40 μs	
Backprojection on GPU	$t_{bp\text{-}512}$	808 ms	V. Vlcek[19]
	$t_{bp\text{-}1k}$	6256 ms	

FLC: ferroelectric liquid crystal; MQW: multiple quantum well; VCEL: vertical cavity emitting laser; DSP: digital signal processor; GPU: graphics processing unit.

Table 2 Comparison of computation times between optical and digital operation in both cases.

	Time			
Description	Digital Filtering	Optical Filtering	Digital Back-projection	Optical Back-projection
	Case 1			
One projection	40 μs			20 μs[a]
Full reconstruction	14.1 ms	1 ms	808 ms	7.2 ms[b]
	Case 2			
One projection	88 μs			20 μs[a]
Full reconstruction	63.3 ms	1 ms[d]	6256 ms	16 ms[c]

[a]Limited by the rotating system.
[b]36 acquisitions per reconstruction.
[c]8 acquisitions per reconstruction.
[d]Adapted SLM (1 Mpixel, 1 kHz) does not exist on the market yet. Estimations are based on the limitation of the FLC technology (frame rate $>$ 1 kHz) and the image transfer data rate (up to 10 Gbit/s for high-speed image sensors).

result remains true as long as the resolution of the optoelectronic components is sufficient in comparison with K, M, and N.

Our analysis is based on the performance of up-to-date optoelectronic devices, which are described in Table 1. Two cases are considered: The first one corresponds to the typical requirements in medical imaging ($L=729$, $K=360$, $M=512$, $N=512$), whereas the second one corresponds to the need of the next generation of scanners, which is expected to manage larger images ($L=1453$, $K=720$, $M=1024$, $N=1024$).[20] Results are given in Tables 2–4.

The speedup is between one and two orders of magnitude for current systems and will be between two and three for future image sizes. The major improvement is given by the optical backprojection core. The use of optical filtering involves a limited speedup factor between 2 and 4. Taking degradations introduced by this technique into account (especially because of the use of coherent light), it is more realistic to consider standard digital filtering components and a standard optimization (parallelization, pipelining, etc.) instead. Such a hybrid implementation will improve the quality of the reconstruction without increasing the computation time.

The parameter N_{acq} has been chosen to have the greatest value allowing $N_{acq}t_{acq}$ to be nondominant in Eq. (11). The values, 36 and 8, are high enough to have a good coding depth for the reconstructed images: 13 to 15 bits with a 10-bit image sensor (which is the current standard value[17]).

The last point concerns the data rate at the input and the output of the optical system (including the digital accumulation of images). They are given by following equations:

- At the input: $D_{in}=(d_{disp}n_{disp})/t_{disp}$, (where d_{disp} is the display coding depth and n_{disp} the number of pixels of the system). The maximum data rate required is about 555 Mbit/s (for a 1453-point 8-bit display with a 20-μs refresh rate).
- At the output: $D_{out}=dNM/t_{fbp}$ (where d is the coding depth of reconstructed images and t_{fbp} is the computation time of the FBP algorithm). The maximum data rate required is about 1000 Mbit/s (for a

Table 3 Speedup according to the chosen configuration in case 1.

Filtering	Backprojection:	Digital	Optical
Digital		1	56
Optical		1	112

Table 4 Speedup according to the chosen configuration in case 2.

Filtering	Backprojection:	Digital	Optical
Digital		1	98
Optical		1	390

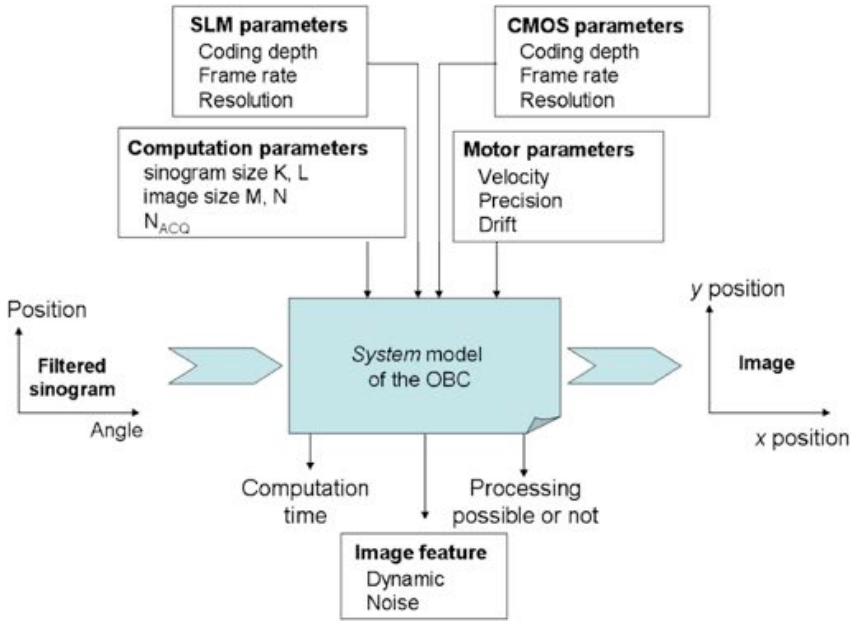

Fig. 4 The system model of the OBC.

1024×1024-point 16-bit reconstructed image with a 16-ms computation rate).

These rates appear reachable in comparison with standard data rates.

5 Simulation of the Optical Backprojection Core

This analysis tends toward proving that the optical backprojection is very promising with respect to computation time. We now aim at estimating distortions introduced by this method and their effect on the quality of the reconstructed image. First, a model of the optical backprojection core (OBC) has been established in order to study this point with simulations.

5.1 *System Model of OBC*

First, a very simple model of the OBC has been established to map features of the components of the system to OBC performance (Fig. 4). It has been partly used in the computation-time analysis in the previous section. Concerning degradation sources, only quantization effects and motor positioning accuracy (motor angular positioning noise, or MAPN) have been taken into account. The model of MAPN is the following:

$$\theta(t) = (V + \Delta\theta)t + \delta\theta, \tag{12}$$

where V is the rotation velocity of the motor, $\Delta\theta$ is an angular drift (low-frequency noise), and $\delta\theta$ is the positioning noise (high-frequency noise). The optical setup is supposed to be ideal.

The quantitative criterion chosen to compare the images is the signal-to-noise ratio (SNR), which is the normalized mean squared error (NMSE) obtained by considering the digitally reconstructed image as a reference. This parameter is only relevant if each image pixel can be considered as one realization of independent random variable. Since this condition is not fulfilled by synthetic images—because of textures—natural images are preferred for this analysis. We have

$$\mathrm{SNR}_{\mathrm{dB}} = 20\log(\mathrm{NMSE}) = 10\log\left(\sum_{i=1}^{N}\sum_{j=1}^{M}\frac{|I_{\mathrm{opt}}(i,j) - I_{\mathrm{num}}(i,j)|^2}{I_{\mathrm{num}}(i,j)^2}\right). \tag{13}$$

Figures 5–8 illustrate the effects of various parameters of the OBC model. In two Figs. 5 and 6, variations of SLM and CMOS parameters are considered; in Figs. 7 and 8 the motor parameters.

We can point out that the spatial distribution of the error is uniform for an 8-bit and a 4-bit display system (some edge effects appear with a 4-bit system). The binary reconstruction (with a 1-bit system) works, but the image is not usable. The noise is naturally filtered, as the algorithm includes a summation process. This can be observed in Fig. 6 (left): With quantized projections having predominant noise ($\mathrm{SNR} < 0$ dB), one can reconstruct an image that exhibits a SNR up to 30 dB higher. This improvement is to be linked with the number of projection used for the reconstruction—$10\log(720) \approx 28.5$ dB.

The number of acquisitions by reconstruction, N_{acq}, does not allow one to improve the SNR significantly, especially when the input is quantified, because the SNR on a single acquisition (60 dB for a 10-bit camera) is superior to the maximum SNR that can be obtained with a 8-bit quantified display system (55 dB according to Fig. 6). On the other hand, increasing N_{acq} permits one to reduce the distortion

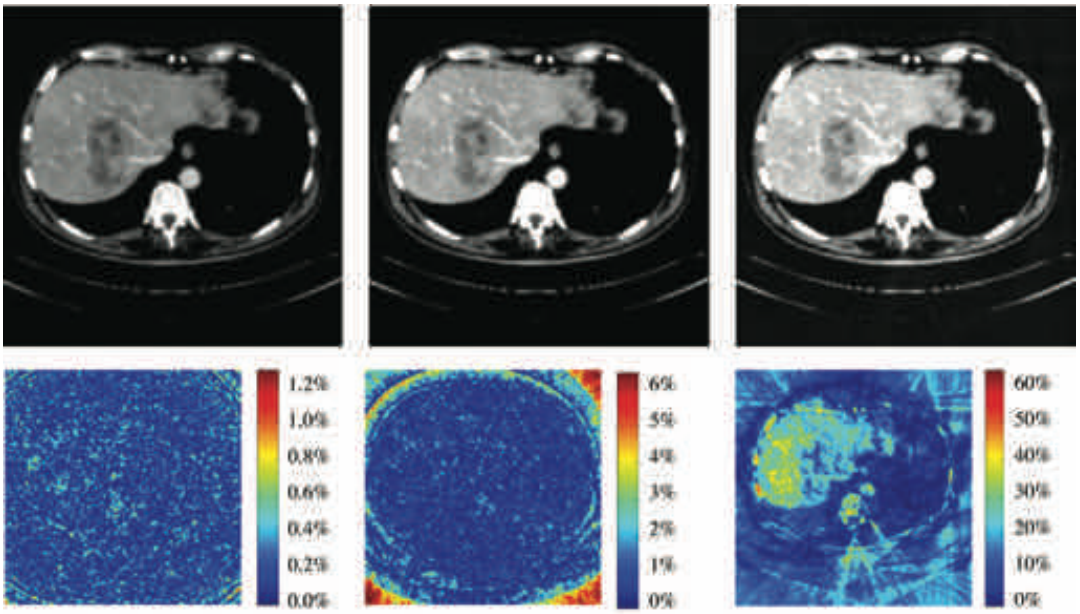

Fig. 5 Simulation: Effect of the quantization. Examples of optically reconstructed images (top) and the difference between those images and the digitally reconstructed ones (bottom) are represented respectively for a 8-bit (left), 4-bit (middle), and 1-bit coding depth (SLM).

due to the saturation on the camera. This point is not illustrated in this study, but can be critical in practice.

Positioning noise leads to a noise with a uniform spatial distribution, whereas drift alters the position of the image and leads to a progressive shift. The reconstructed image is blurred, and geometrical distortion appears. Simulation reveals that the noise resulting from quantization is weak (the expected SNR is about 50 dB with standard material) compared respect to the noise related to motor positioning accuracy. The constraints that ensure that the SNR on the reconstructed image remains higher than 20 dB are that rotation velocity should be adjusted to 0.2% and the instantaneous positioning noise should be less than 0.1 deg.

The simulation results obtained with this model are encouraging. Nevertheless, one should remember that the optical setup introduces additional degradations.

5.2 *Optical Model of the OBC*

In this subsection, we focus on the effect of Dove-prism misalignment on reconstructed image quality.

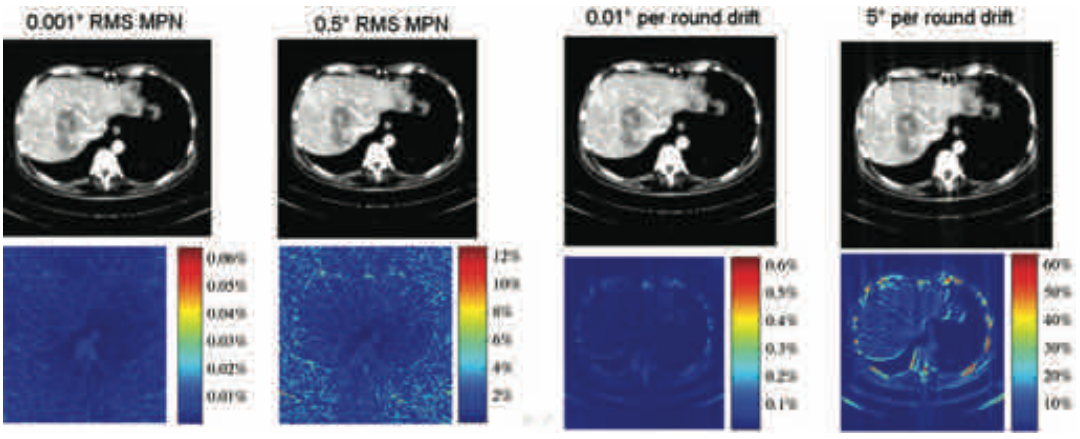

Fig. 7 Simulation: Motor positioning accuracy. Examples of optically reconstructed images (top) and the difference between those images and the digitally reconstructed one (bottom) for different motor parameters.

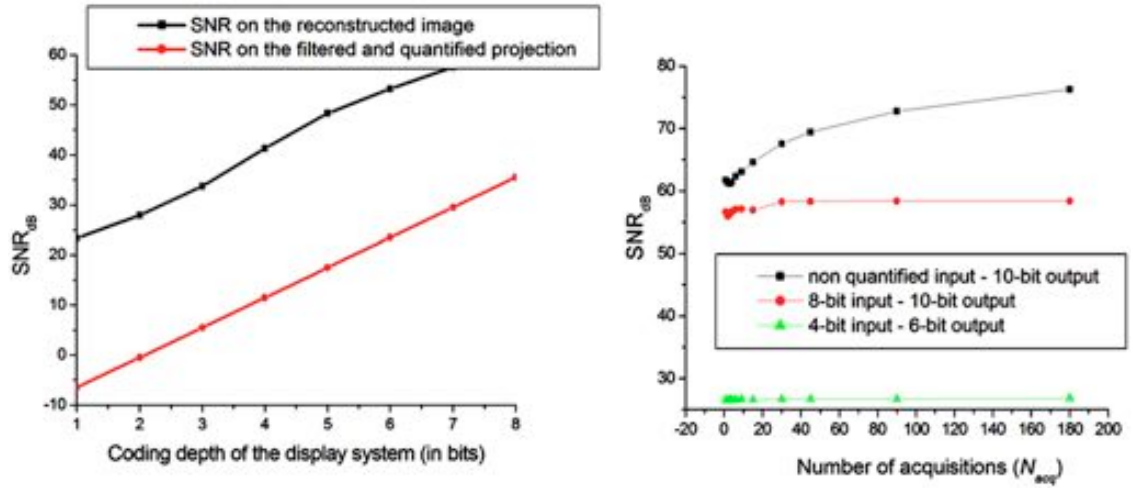

Fig. 6 Simulation: Effect of the quantization. The simulated SNR of the optically reconstructed image is studied according to the input display dynamic (on the left) with an output dynamic fixed to 10 bits, and according to the output camera dynamic and the parameter N_{acq} (on the right).

We consider the OBC setup where the prism is located between the last spherical lens and the sensor (which is slightly different from the setup presented in Fig. 2, but equivalent). The position of the camera is adjusted so that the image in plane M is focused on the sensor. We first assume that the prism is well aligned and the rotation angle is 0 deg (Fig. 9). Hereafter, we describe the effect of the prism on a ray, parallel to the optical axis, emerging from the input plane. Let $M(x,y)$ be the coordinates of the emerging ray in the input plane. Its location N in the intermediate plane has coordinates $(-\gamma x,-\gamma y)$ where γ is the optical magnification factor. After passing through the Dove prism, the position P of the parallel ray in the output plane corresponds to $(\gamma x,-\gamma y)$.

We now suppose that the prism is turned through an angle θ around its axis. We define a new coordinate system (CS), the so-called prism CS (O,x',y',z), obtained by the rotation of the lab CS through an angle $-\theta$. In this CS, the configuration is the same as in Fig. 9. The coordinate of the point N in the new coordinate system is

$$N:\begin{pmatrix} x'_n \\ y'_n \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} x_n \\ y_n \end{pmatrix}. \tag{14}$$

As previously, the coordinates of the point P reached by the ray emerging from N in the new coordinate system are given by

$$P:\begin{pmatrix} x'_p \\ y'_p \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x'_n \\ y'_n \end{pmatrix}. \tag{15}$$

If we return to the lab CS,

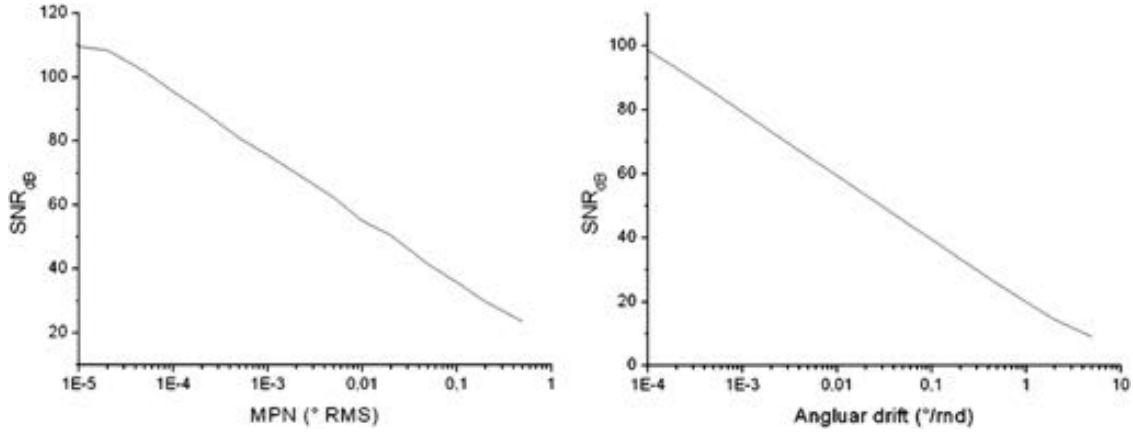

Fig. 8 Simulation: Motor positioning accuracy. The simulated SNR on the optically reconstructed image is studied according to the motor parameters.

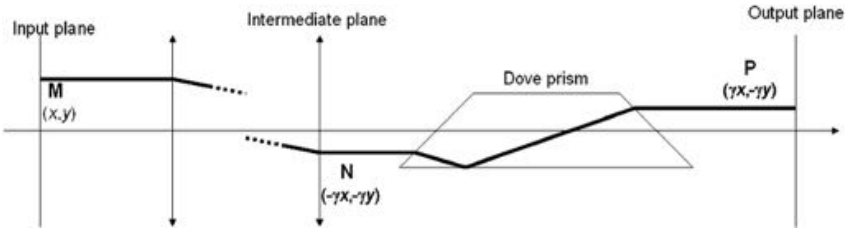

Fig. 9 The Dove prism in its basic ideal configuration.

$$P:\begin{pmatrix} x_p \\ y_p \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} x'_p \\ y'_p \end{pmatrix}. \tag{16}$$

By combining Eqs. (14) and (16), we can write

$$\begin{aligned} P:\begin{pmatrix} x_p \\ y_p \end{pmatrix} &= \begin{pmatrix} -\cos 2\theta & -\sin 2\theta \\ -\sin 2\theta & \cos 2\theta \end{pmatrix}\begin{pmatrix} x_n \\ y_n \end{pmatrix} \\ &= \begin{pmatrix} \gamma\cos 2\theta & \gamma\sin 2\theta \\ \gamma\sin 2\theta & -\gamma\cos 2\theta \end{pmatrix}\begin{pmatrix} x_m \\ y_m \end{pmatrix}. \end{aligned} \tag{17}$$

The image in the output plane corresponds to the image in the input plane rotated by an angle 2θ and mirrored.

5.2.1 *Equations of the Dove prism for any incident ray*

We now consider a ray that is not parallel to the optical axis. Dimensions and angles are defined in Fig. 10. We first focus on the (O',x,z) plane.

The origin is associated with the lower left corner of the prism. In this configuration, the coordinates of the input point of the incident ray are (a',a'). Using the Snell-Descartes principle, we can write

$$n\sin\left(\frac{\pi}{4}-\beta\right) = \sin\left(\frac{\pi}{4}+\alpha\right). \tag{18}$$

Hence,

$$\beta = \frac{\pi}{4} - \arcsin\left[\frac{\sin(\pi/4+\alpha)}{n}\right]. \tag{19}$$

The ray is then reflected by the base of the prism and reaches the output face, where it is refracted for the second time:

$$n\sin\left(\frac{\pi}{4}-\beta\right) = \sin\left(\frac{\pi}{4}-\gamma\right). \tag{20}$$

We deduce from Eqs. (18) and (20) the first relation for the Dove prism:

$$\gamma = -\alpha. \tag{21}$$

The equation of the straight line AB is

$$x' = \frac{\tan(-\beta)}{(z'-a')} + a'. \tag{22}$$

That is to say that the coordinates of the point B are

$$(b',0) \qquad \text{with} \quad b' = a'\left(1+\frac{1}{\tan\beta}\right). \tag{23}$$

In the same way, the equation of BC is

$$x' = \frac{\tan\beta}{(z'-b')}. \tag{24}$$

As the point C belongs to both the straight line BC and the output face of the prism, its coordinates are

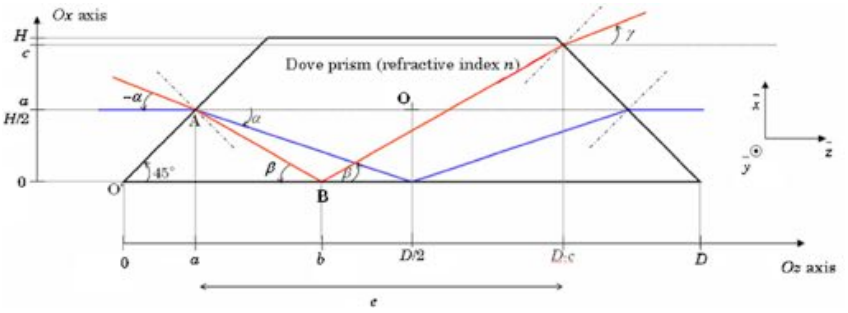

Fig. 10 Configuration in the *Oxz* plane and definitions of constants used to calculate the input-to-output equation of the Dove prism for any incident ray.

$$(D-c',c') \quad \text{with} \quad c'=\frac{\frac{\tan\beta}{(D-b')}}{1+\tan\beta}. \tag{25}$$

With Eq. (25), we can also write the second equation for the prism:

$$c'=-a'+\frac{D\tan\beta}{\tan\beta+1}. \tag{26}$$

If the small-angle approximation (i.e., rays are weakly tilted with respect to the $O'z$ axis, or $\alpha \ll 1$) can be used, the equations can be linearized. The expressions for β and $\tan\beta$ can be approximated by the first-order Taylor expansion,

$$\beta=\frac{\pi}{4}-\left[\arcsin\left(\frac{\sqrt{2}}{2n}\right)+\frac{\alpha}{(2n^2-1)^{1/2}}\right]+O(\alpha^2). \tag{27}$$

In the same way, the first-order Taylor expansion of c is given by

$$c'=-a'+D\left(\frac{K}{K+1}-\alpha\frac{1+K^2}{(K+1)^2(2n^2-1)^{1/2}}\right)+O(\alpha^2), \tag{28}$$

where K is a constant defined by

$$K=\tan\left[\frac{\pi}{4}-\arcsin\left(\frac{\sqrt{2}}{2n}\right)\right]. \tag{29}$$

If a increases, the ray is reflected further along the base of the prism and exits the prism at a lower position (c decreases). If α increases, the refracted ray is deviated more from the normal. It is reflected further along the base of the prism, and c decreases. Both these assertions are confirmed by Eq. (28).

The origin is now set as the center of the prism, O, also being the origin of the lab's CS. The values of c and a in this CS are obtained by a translation of half the height H of the prism. Thus,

$$c=-a+D\left(\frac{K}{K+1}-\alpha\frac{1+K^2}{(K+1)^2(2n^2-1)^{1/2}}\right)-H+O(\alpha^2). \tag{30}$$

H and K are linked. Let us consider the case where $\alpha=0$. According to the geometrical properties of the Dove prism, the ray entering the prism at the center of the input face ($a=0$) exits at the center of its output face ($c=0$). Therefore, the relation between H and K can be deduced from

$$H=\frac{DK}{K+1}. \tag{31}$$

The glass thickness the ray goes through is given by

$$e=D-H-(c+a)=D\left(\frac{1}{1+K}+\alpha\frac{1+K^2}{(K+1)^2(2n^2+1)^{1/2}}\right). \tag{32}$$

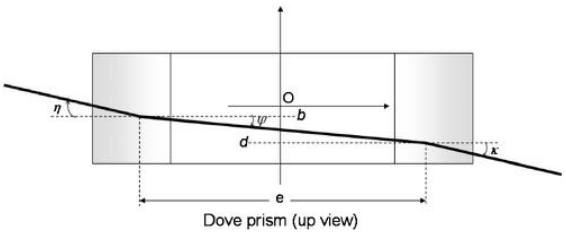

Fig. 11 Configuration in the Oxz plane and notation used to calculate the input-to-output equation of the Dove prism for any incident ray.

Now, we focus on the (O,y,z) plane, where a double refraction occurs (Fig. 11). According to geometrical considerations, one can write

$$\kappa=\eta,$$

$$\sin\eta=n\sin\psi \quad \Rightarrow \quad \psi=\frac{\eta}{n},$$

$$\tan\psi=\frac{d-b}{e} \quad \Rightarrow \quad d=b+e\psi. \tag{33}$$

Finally, the full equation set for the Dove prism for any ray under the small-angle approximation can be summarized as

$$c=-a-\alpha D\frac{1+K^2}{(K+1)^2(2n^2-1)^{1/2}},$$

$$d=D\left(\frac{1}{1+K}+\alpha\frac{1+K^2}{(K+1)^2(2n^2+1)^{1/2}}\right)\frac{\eta}{n}+b,$$

$$\gamma=-\alpha,$$

$$\kappa=\eta. \tag{34}$$

5.2.2 *Nonideal model of the OBC*

The nonideal model of the OBC uses the equation set of the Dove prism. The configuration under study is described by Fig. 12. Three axes are considered: The prism axis, the optical axis, and the rotation axis, which is taken as a reference. The origin of the system is the center of the Dove prism. The prism axis is tilted with respect to the rotation axis (angles φ and ψ in Fig. 12) and shifted (distances r and s). At the same time, the optical axis (not represented in the figure) is also tilted and shifted (α,η,t,u).

The model computes directly the reconstructed image $g(x,y)$ according to the projection set. For a given configuration (at a given backprojection angle θ), we first find, for each pixel of the input image, the trajectory of the ray emerging from that point and parallel to the optical axis. This is done in six steps:

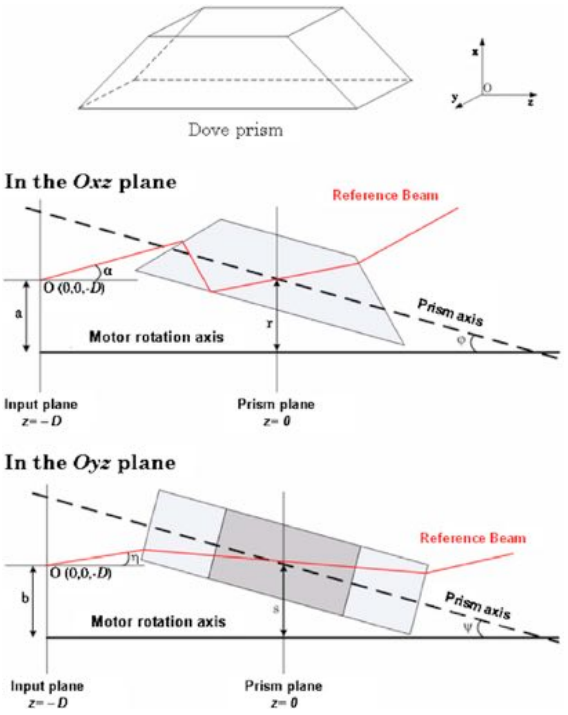

Fig. 12 Notation for the nonideal Dove-prism model.

- Calculation of the parameters (a, b, α, η) of the ray with respect to the rotation axis, as described in Fig. 12. Here a, b depend on the position (x, y) of the pixel in the input plane and on the parameters t, u.
- Change of CS so that the prism is in the same configuration as in the previous study (Figs. 10 and 11). Three CS rotations (φ, ψ, and θ) and two translations (s and r) are carried out. The parameters of the ray must also be affected by this operation. Let (a', b', α', η') be the new parameters in this CS.
- Calculation of the input point on the Dove prism.
- Use of the Dove-prism equation set to compute the output point.
- Calculation of the parameter of the ray that emerges from the Dove prism, first in the Dove prism CS $(c', d', \gamma', \kappa')$, and then in the reference CS (c, d, γ, κ).
- Search for the intersection between the emerging ray and the sensor plane.

Due to the misalignment, the image is only partially focused on the sensor. The image of one point in the input image is a spot on the output plane whose radius R depends on the f number of the system and the difference of optical paths, Δl. We consider the following approximation:

$$R = \frac{2\,\Delta l}{F}. \tag{35}$$

A bilinear interpolation and a weighting are used to calculate the effect of each input pixel value on targeted output ones. A resulting image is computed for each backprojection angle and added to obtain the simulated reconstructed image.

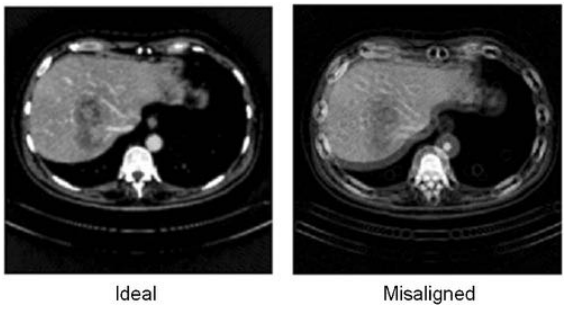

Fig. 13 Example of a reconstructed image (on the right) with a misalignment between the optical axis and the rotation axis of 5 pixels and 0.1 deg. The ideal image is displayed on the left.

5.3 *Results*

This model is first used to estimate the accuracy needed with respect to the alignment of optical elements. Second, it will also be very useful for the experimental alignment task to understand the observed phenomena and then to facilitate their compensation.

Figure 13 gives an example of a reconstructed image with a misalignment between the optical axis and the rotation axis of 5 pixels and 0.1 deg. Each point of the image seems to have traced out along an ellipse.

Figure 14 illustrates the effect of misalignment between the Dove prism axis and the rotation axis. Each curve corresponds to the trajectory, in the output plane, of the center of the input plane in different configurations. In the case of a perfect alignment, the trajectory is reduced to the center of the output plane. Such trajectories have been observed with the experimental prototype during the alignment stage.

A quantitative analysis of the distortion has also been carried out. The signal-to-noise ratio is computed according to misalignment parameters (Fig. 15). Their effect seems to follow a first-order behavior with a SNR decrease of 20 dB per decade. We can estimate that the angular misalignment should be less than 0.01 deg and the radial one less than 1 pixel in order to keep the SNR above 20 dB. Such strong constraints make the realization of the OBC very tricky. In addition, this study seems to confirm that quantization noise can be neglected in comparison with misalignment noise.

6 Experimental Results

We have built an experimental prototype of the OBC. For practical reasons, our setup incorporates an array SLM instead of a linear one (Fig. 16).

6.1 *Optical Setup*

For practical reasons, our experimental prototype of OBC differs slightly from the ideal one we have considered in previous sections. First, an array SLM has been considered

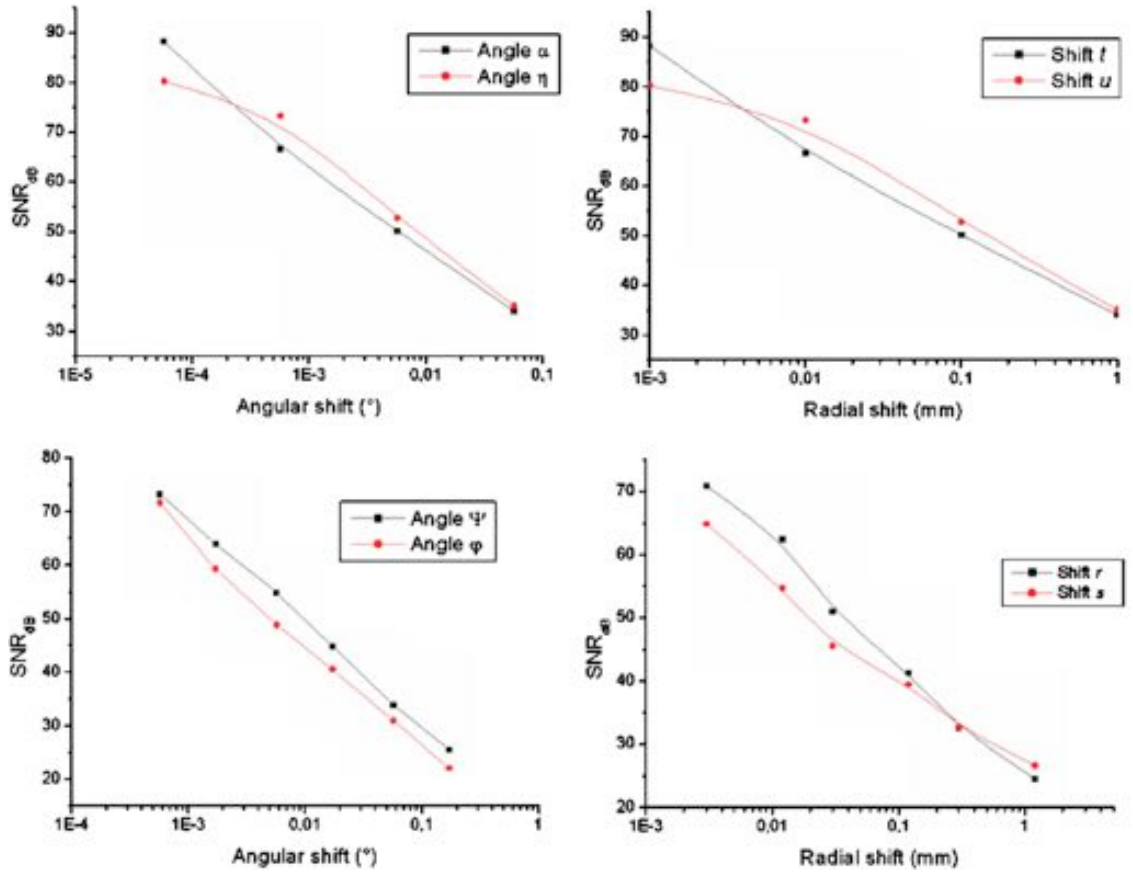

Fig. 15 Simulation: signal-to-noise ratio according to the angular or radial displacement.

instead of a linear one. Second, the frame rate of our experimental setup is about 100 times slower that the ideal OBC, in order to use standard devices.

We now describe our experimental prototype (Fig. 16). The light of a 1-W high-power red LED is focused on an optical fiber. The spatially uniform beam produced at the output of the fiber is collimated by a lens and illuminates the array SLM used to display filtered and extended projections. The image produced goes through a 4f afocal system in order to make it sharp on the sensor. A Dove prism mounted at the center of a motorized gantry is introduced between the last lens and the sensor, as in the ideal OBC optical model. Its configuration is the same as the one studied in simulation. The flash, the image displayed, the position of the motor, and the acquisition system are synchronized with a Labview interface, which realizes the sequence shown in Fig. 17.

The prism and the motor were been aligned with care, using a laser beam as reference axis and with the help of the preceding model to understand the observed phenomenon. The filtered and extended projections contain positive and negative values, which can be displayed. There are two alternatives to overcome this difficulty. For the first one, an offset is added to projections to ensure positive values, before subtracting it L times from the reconstructed image. The second alternative consists in separating each projection into two subframes, one containing only positive values (negative ones are replaced by 0) while the other contains only negative ones. The final reconstructed image is the difference between the first and the second subframe. In our example, the second method will be favored: It provides a better image dynamic and increases the computation speed by a factor 2.

6.2 *Results*

Experimental results are reported in Table 5 and Figs. 18–20, and are discussed below.

Figure 18 concerns the reconstruction of a trivial geometrical shape (a square). Two kinds of distortions are observed: Geometrical ones and radiometric ones (affecting gray levels). Geometrical distortions, related to square borders (maximum error), clearly predominate. Indeed, as observed in the uniform area of the image and in the intensity profile, radiometric distortions appear negligible (evaluated mean squared error of 1.4%).

Figure 19 concerns the reconstruction of a medical

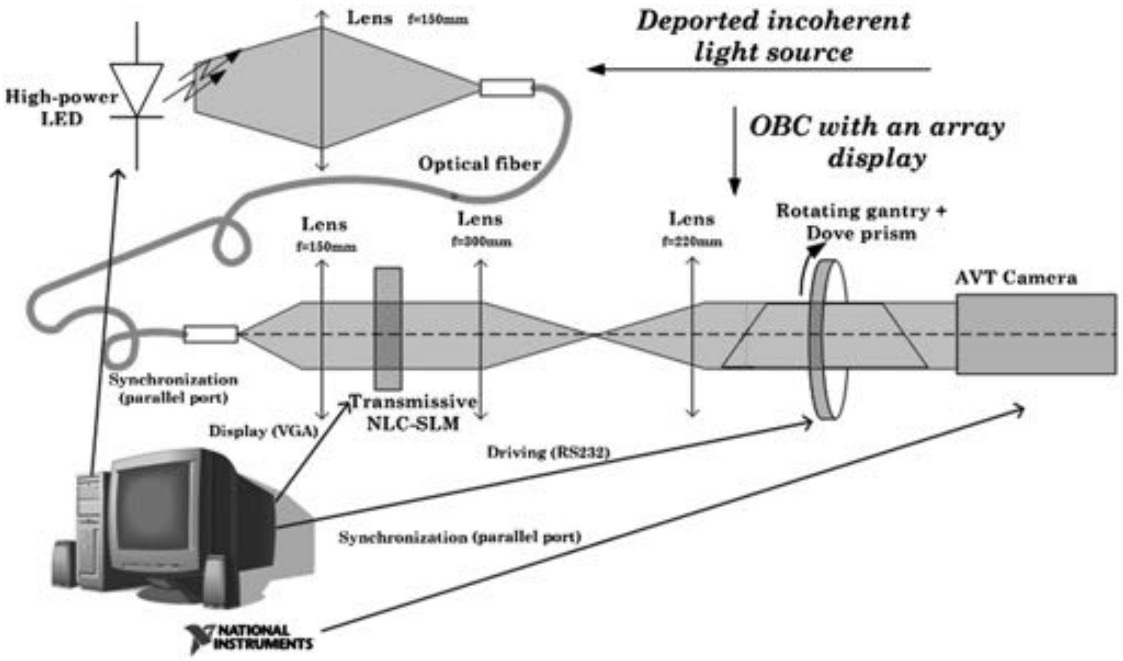

Fig. 16 Experimental setup.

phantom. When this image is synthesized, the mathematical expression for every component (equation of ellipse) of the image is known *a priori*. We performed a quantitative analysis of the geometric distortions by comparing equations of ellipses issuing from the reconstructed image with the original ones. Table 5 gives a comparison between the mathematical and the measured parameters of each ellipse (major axis, minor axis, coordinates, angle, and gray level) on the optically reconstructed image. The differences seem to be small (less than 5%). Note that this evaluation results from a global statistical integration involving the compensation of local distortions of the ellipses. Radiometric noise appears more important (up to 10% in some regions) and can be explained by a saturation phenomenon in the sensor. Nevertheless, in this example, we can observe that segmentation (classification of pixels in areas according to their gray level) remains possible with good efficiency (94% of pixels are classified in the same area for both the optically reconstructed image and the digitally reconstructed one).

Figure 20 concerns the reconstruction of a natural medical image, for illustration purposes only. The image seems correctly reconstructed except for a slight blurring effect. For example, some details (veins or fat tissues) magnified in the region of interest appear distinctly in both pictures. It should be noticed that the slice presented on Fig. 16 is only one example of a natural image picked from the medical field. The sinogram used to compute the inverse Radon transform was synthesized from the original 2-D image and is not the result of real 3-D acquisition.

7 Discussion

The alternative way to speed up the FBP algorithm implementation proposed in this paper has been studied in simulation and with an experimental prototype.

As foreseen, the quality of processing is worse than with a digital solution. Nevertheless, it can be enhanced by improving the optical setup. This point is essential in order to envisage a commercial exploitation.

First, one should take care of the linear display. For the moment, there are no existing commercial devices meeting our needs. There are two alternative technologies: A passive SLM with an external light source, or a self-emitting system. At first sight, LEDs or organic LEDs seem to be most mature for this application. A device with large but thin pixels would be ideal for our application in order to optimize, in one direction, the integration capacity and, in the other, the amount of emitted light.

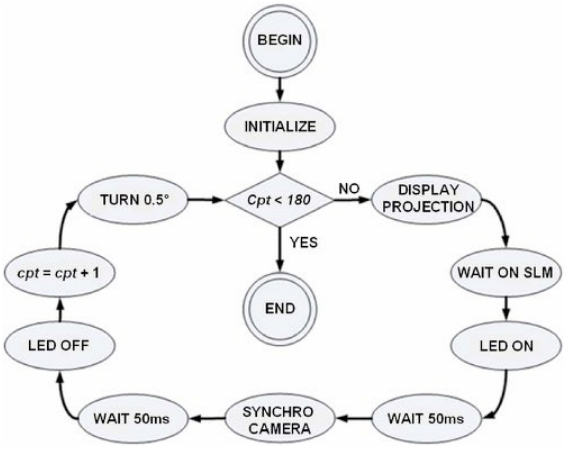

Fig. 17 The reconstruction sequence.

Table 5 Comparison between the mathematical and the measured parameters of the ellipses composing the Shepp-Logan phantom.

Synthesized phantom					Optically reconstructed phantom			
Center	Min. axis	Maj. axis	Gray level	Angle (deg)	Center	Min. axis	Maj. axis	Gray level
(0, 0)	0.69	0.92	1	90	(0, 0)	0.71	0.88	1
(0, −0.018)	0.66	0.87	−0.8	90	(0, −0.010)	0.64	0.84	−0.752
(0, 0.35)	0.21	0.25	0.1	90	(−0.002, 0.332)	0.20	0.24	0.200
(0, 0.1)	0.046	0.046	0.1	0	(−0.008, 0.082)	0.043	0.043	0.98
(0, −0.1)	0.046	0.046	0.1	0	(−0.008, −0.104)	0.043	0.043	0.167
(−0.08, −0.605)	0.023	0.046	0.1	0	(−0.086, −0.604)	0.021	0.055	0.154
(0, −0.609)	0.023	0.023	0.1	0	(−0.004, −0.609)	0.021	0.022	0.159
(0.06, −0.598)	0.023	0.046	0.1	90	(0.047, −0.598)	0.039	0.032	0.163
(0, −0.22)	0.16	0.41	−0.2	72	(0, −0.228)	0.156	0.410	−0.258
(0, 0.22)	0.11	0.31	−0.2	108	(0, 0.218)	0.110	0.302	−0.253

Distances are given in relative distance (u.a.d) with respect to the image size. The image size is 2 u.a.d × 2 u.a.d and the center is in (0,0). A 1 u.a.d diameter circle means a circle whose diameter is half the size of the whole image.

Second, the optical part, involving the Dove prism, the motor, and the lenses, should be optimized and integrated. It has been shown that the precision required in the alignment of those components is crucial in order to have good image quality. The realized prototype highlights distortions encountered if this point is neglected. Our experience lets us think that an on-table setup should be avoided because of the difficulty of alignment, the lack of precision of the alignment apparatus (magnetic support, micrometric screw, etc.), and the small uncontrolled displacement of parts (the alignment of the setup must be checked and corrected often). From this point of view, an integrated optomechanical system should be preferred. Solid-state optics can be a good alternative too, but the motion of the Dove prism and the motor can be very challenging.

In our situation, the acquisition system (the camera) is no longer a problem. Indeed, high-speed systems with very good image quality and integrated processing capacity are already being commercialized.[21]

FBP was used to reconstruct an image from a 2-D sinogram, which is rarely done in the medical field, for instance, where 3-D scanners are used. Unfortunately, our system could not be directly extended to a 3-D geometry. Nevertheless, the OBC would be of interest for some 3-D algorithms, (so-called *rebinning algorithms*), for they reduce the 3-D problem to a set of 2-D ones. The advanced single-slice rebinning (ASSR), useful and well known for helical CT scans, is an example.[22] An adaptation of this system in a 3-D ASSR hybrid processor is still under study.

In this situation, the speedup offered by the proposed architecture could benefit future applications in assisted therapy, for example. Indeed, it is well known that respiratory motion and heartbeat limit the accuracy of many surgical interventions (laparoscopic procedures, cardiac or liver interventions, urological surgery, radiotherapy, etc.).[23] Although those motions can be limited or stopped in some cases, they remain challenging, especially for robots, which have to be synchronized with the natural periodic motions of the patient. The frequency band ranges between 0 and 30 Hz, according to the targeted organ.[23] The efficiency of the feedback system depends on the sampling frequency, which should be at least 60 Hz, according to Shannon's theorem, and up to 300 Hz in some cases. Some techniques using traditional 2-D information provided by a high speed camera to measure organ motions and synchronized robotic instruments have been demonstrated.[24] In the near future, a fast tomographic acquisition system combined with a high-speed reconstruction processor, including our OBC, for instance, should be able to provide 3-D volumic information with a sufficient refresh rate to meet the needs of such synchronized robots.

8 Conclusion

An optical processor dedicated to the FBP algorithm has been presented in this paper. It has been studied in simulation and with an experimental prototype. The architecture has a huge potential in computation power: The OBC is about two orders of magnitude faster than a standard up-to-date digital processor.

Some limitations have been highlighted. The leading one concerns the difficulty in aligning the optical setup. Distortions due to misalignment appear in the image reconstructed with our prototype and have been explained and studied in simulation with an optical model of the OBC. According to this model, the axial and angular alignment should be improved by about one order of magnitude in order to reach very good image quality.

Nevertheless, the results obtained with our rough prototype remain exploitable. In the discussion, some proposals

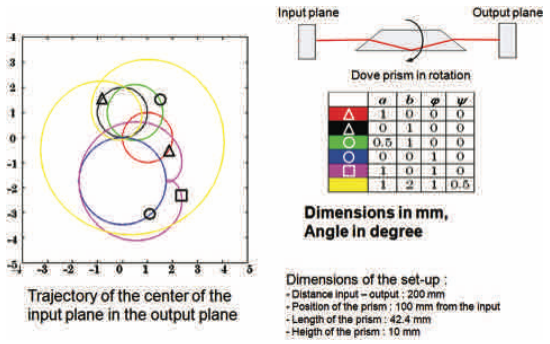

Fig. 14 Illustration of the effect of the misalignment between the Dove prism axis and the rotation axis. Each curve corresponds to the trajectory of the center of the input plane for a specific configuration.

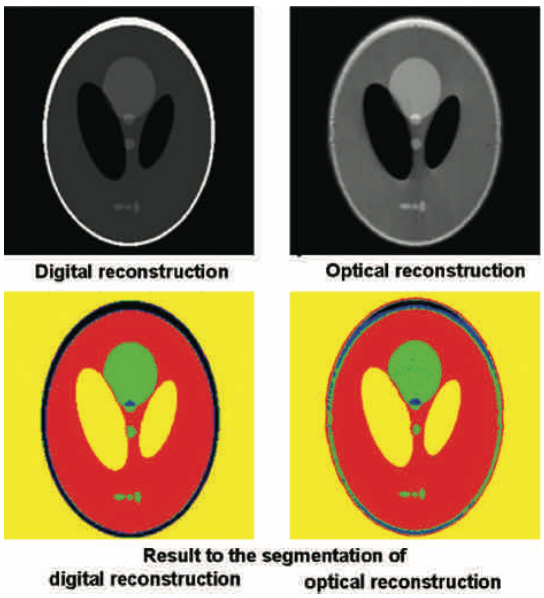

Fig. 19 Reconstruction of a phantom with the hybrid processor including the OBC.

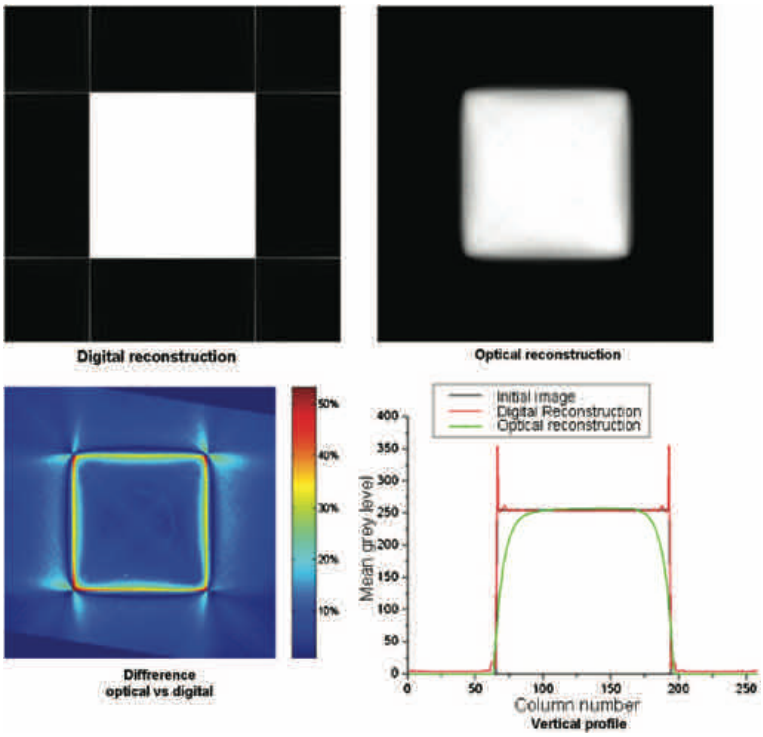

Fig. 18 Reconstruction of a square with the experimental hybrid processor including the OBC.

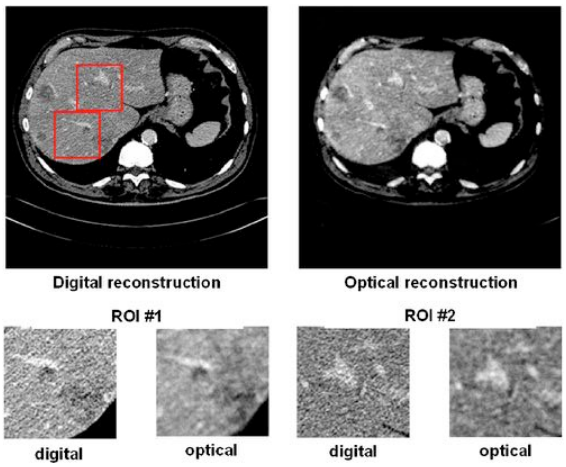

Fig. 20 Reconstruction of a medical image with the hybrid processor including the OBC (ROI stands for region of interest).

have been put forward in order to get round the difficulties. The OBC can be, in the near future, a very helpful system to speed up tomographic reconstructions in order to meet challenging needs such as those arising in computer-assisted therapy.

Acknowledgments

This research was supported by Micro Module under a CIFRE contract (No. 414/2003) between that company and the InESS Laboratory. The authors want to thank Mr. Jérémy Bartringer (R&D engineer at the InESS) for his involvement in this project and especially his invaluable help in the realization of the prototype; Mr. Aurélien Mazel (engineer student at the Ecole Nationale Supérieure de Physique de Strasbourg), who also participated in the realization of the prototype; and Mr. Pamela Lhote (R&D assistant at the IRCAD) for a careful review of the English.

References

1. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*, Society of Industrial and Applied Mathematics (2001).
2. J. S. Duncan and L. H. Staib, "Image processing and analysis at IPAG," *IEEE Trans. Med. Imaging* **20**(12), 1505–1518 (2003).
3. R. H. Taylor and D. Stoianovici, "Medical robotics in computer-integrated Surgery," *IEEE Trans. Rob. Autom.* **19**(5), 765–781 (2003).
4. P. Edholm and G. Herman, "Linograms in image reconstruction from projections," *IEEE Trans. Med. Imaging* **6**, 301–307 (1987).
5. S. Basu and Y. Bresler, "$O(N^2 \log_2 N)$ filtered backprojection reconstruction algorithm for tomography," *IEEE Trans. Image Process.* **9**, 1760–1773 (2000).
6. T. Rodet, P. Grangeat, and L. Desbat, "Multichannel algorithm for fast reconstruction," *Phys. Med. Biol.* **47**(15), 2659–2671 (2002).
7. P. E. Danielsson and M. Ingerhed, "Backprojection in $O(N^2 \log_2 N)$ time," in *Proc. IEEE Imaging Conf.* (1997).
8. T. Rodet, P. Grangeat, and L. Desbat, "Algorithme rapide de reconstruction tomographique basé sur la compression des calculs par ondelettes," in *Proc. 19e Colloque GRETSI*, Paris (2003).
9. A. Vander Lugt, *Optical Signal Processing*, Wiley-Interscience, New York (1992).
10. M. Nishimura, D. Casasent, and F. Caimi, "Optical inverse Radon transform," *Opt. Commun.* **24**(3), 276–280 (1978).
11. M. Nishimura, D. Psaltis, F. Caimi, and D. Casasent, "Implementation of the inverse Radon transform by optical convolution," *Opt. Commun.* **25**(3), 301–304 (1978).
12. T. Lu, S. S. Upda, and L. Upda, "Optoelectronic implementation of filtered backprojection tomography algorithm," *Proc. SPIE* **1564**, 704–713 (1991).
13. T. Lu, "Optoelectronic system for implementation of iterative computer tomography algorithms," U.S. Patent No. 5,414,623 (1995).
14. Boulder Nonlinear Systems, www.bnonlinear.com.
15. Lenslet, www.lenslet.com.
16. Newport, www.newport.com.
17. Micron, www.micron.com.
18. Analog Devices, www.analog.com.
19. V. V. VLCEK, "Computation of filtered backprojection on graphic cards," in WSEAS'05 Trans. on Computer, vol. 9(4), pp. 1216–1218 (2005).
20. Siemens Medical, www.medical.siemens.com.
21. Optronis GmbH, www.optronis.com.
22. M. Kachelreiss, T. Fuchs, S. Schaller, and W. A. Kalender, "Advances in single-slice rebinning for tilted spiral cone-beam CT," *Med. Phys.* **28**(6), 1033–1041 (2001).
23. N. Riviere, J. Gangloff, and M. de Mathelin, "Robotic compensation of biological motion to enhance surgical accuracy," *Proc. IEEE* **94**, 1705–1716 (2006).
24. A. Krupa, J. Gangloff, C. Doignon, M. de Mathelin, J. Leroy, L. Soler, and J. Marescaux, "Autonomous 3D positioning of surgical instruments in robotized laparoscopic surgery using visual servoing," *IEEE Trans. Rob. Autom.* **19**(5), 842–853 (2003).

Morgan Madec received his MS degree in instrumentation and microelectronics in July 2003 from the University Louis Pasteur of Strasbourg (France), his Eng. Dipl. in September 2003 from the Ecole Nationale Supérieure de Physique de Strasbourg (France) and his PhD degree in electronics from the University Louis Pasteur of Strasbourg (France) in 2006. From 2003 to 2006, he worked on the design and the characterization of optical processor in order to speed up image reconstruction in the medical field. Since 2006 he has pursued this research activity in the laboratory InESS is Strasbourg in collaboration with Micromodule (Brest, France).

Jean-Baptiste Fasquel received his diploma of the general engineering school ENSPS in France and his diploma of Advanced Studies in applied optics and computer vision from the University of Strasbourg in 1998. He obtained his PhD in digital and optical image processing from the University of Strasbourg, France in 2002. Since 2002, he has been a permanent researcher in computer sciences at the Research Institute against Digestive Cancer (IRCAD, Strasbourg). His main areas of interest are medical image processing and software engineering for computer guided surgery.

Wilfried Uhring received his PhD degree in electronics from the University Louis Pasteur of Strasbourg, France, in 2002. From 1999 to 2002, he worked on the conception and improvement of ultrafast optical detection devices such as Streak camera. Since 2003 he has pursued this research activity in order to develop a solid streak camera within a simple CMOS chip. Since 2004, he has extended this research to use the fast video CMOS sensors in optoelectronic processing. In January 2005, he joined the new laboratory InESS (Institut d'Électronique du Solide et des Systèmes).

Pascal Joffre received his PhD degree in optoelectronics from the Institut National Polytechnique de Grenoble, France, in 1991. From 1983 to 1998, he developed liquid crystal devices for infrared and visualization in the Thomson-CSF central research laboratory. He was in charge of commercial aspects, intellectual properties and development tasks. In 1999, he joined Micro Module where he is director and sales manager.

Yannick Hervé received his Agrégation degree from Ecole Normale Supérieure in electrical sciences in 1985, his PhD degree in computer science and electronics in 1988, and HDR degree in 2003. He is an associate professor at the University Louis Pasteur in Strasbourg, France. His research work in the field of VHDL-AMS virtual prototyping is within the Department of Complex Systems Modeling and Simulation at the InESS-UMR 7163 Laboratory in Strasbourg, France. He was the co-founder of ″Systems'Virtual Prototyping″ Company.

Optical Engineering 47(5), 058201 (May 2008)

Optoelectronic implementation of helical cone-beam computed tomography algorithms

Morgan Madec
Institut d'Electronique du Solide
et des Systèmes
23, rue du Loess
67037 Strasbourg Cedex 02, France
and
Société Micro Module
38, rue du Jim Sévéllec
29200 Brest, France
E-mail: Morgan.Madec@iness.c-strasbourg.fr

Jean-Baptiste Fasquel
Hôpital Civil
Institut de Recherches contre le Cancer
de l'Appareil Digestif
67091 Strasbourg, France

Wilfried Uhring
Institut d'Electronique du Solide
et des Systèmes
23, rue du Loess
67037 Strasbourg Cedex 02, France

Pascal Joffre
Société Micro Module
38, rue du Jim Sévéllec
29200 Brest, France

Yannick Hervé
Institut d'Electronique du Solide
et des Systèmes
23, rue du Loess
67037 Strasbourg Cedex 02, France

Abstract. This paper presents a hybrid processor dedicated to the reconstruction algorithm in multislice spiral computed tomography. The described architecture focuses on the advanced single-slice rebinning algorithm, which is a basic 2-D-to-3-D rebinning method developed in 2000. The hybrid processor is composed of four cores (rebinning, filtering, backprojection, and interpolation), including an optical processor for the backprojection. The system is modeled with a multi-abstraction-level approach. The model permits one to evaluate the dependence of both the reconstruction quality and the computation time with different parameters (reconstruction parameters, device features, etc.). It is used in a substantial simulation process allowing the identification of predominant degradation sources and the evaluation of their impact, and leading to the specification of each subsystem. A prototype of each core has been realized. The optical core has been identified as the most critical element, although results are very encouraging. This study has underlined that a computational speedup of more than two orders of magnitude could be reached. This is expected to be very useful for future challenging applications in the field of image-guided computer-assisted surgery, where the reconstruction rate would become critical to ensuring acceptable responsiveness.
[DOI: 10.1117/1.2931691]

Subject terms: computed tomography; optical signal processing; ASSR; system modeling.

Paper 070901R received Nov. 8, 2007; revised manuscript received Feb. 25, 2008; accepted for publication Mar. 7, 2008; published online May 29, 2008.

1 Introduction

Today, imaging is often used in computer- and robot-assisted surgery.[1] At present, only 2-D images can be generated at a sufficient frame rate to be exploited in a lookup system.[2] Three-dimensional images could provide more complete information, improving the accuracy of guidance. Nevertheless, due to the reconstruction time, real-time or near-real-time servoing cannot be envisaged, although required to take physiological motion (respiratory and heartbeat) into account to facilitate some surgical interventions (laparoscopic procedures, cardiac or liver interventions, urological surgery or radiotherapy, etc.).[3] Although these motions can be limited or stopped in some cases, they remain challenging, especially for robot servoing. The frequency band extends from 0 to 30 Hz, depending on the targeted organ.[3] The efficiency of the lookup system depends on the sampling frequency, which should be at least 60 Hz, according to Shannon's theorem. In addition, several slices are required to gather enough information. As a consequence, a reconstruction rate of 300 frames/s (for a 5-slice-thick volume at 60 Hz) seems to be a minimum. On the other hand, for surgical operations based on computed tomography (CT) (e.g., biopsies,[4] spine surgery[5]), imaging is used in a human visualization process. As a consequence, the frame rate is less critical but the reconstructed volume is bigger, which leads to a similar reconstruction-rate requirement. Currently available reconstruction rates are one to three orders of magnitude lower (depending on the application), even with up-to-date electronic solutions embedded in medical scanners.[6,7] Moreover, the development of the hardware will probably slow down in coming years (due to the expiration of Moore's law).

The most time-consuming operation in a reconstruction process is always the inverse Radon transformation, whatever algorithm is used. Three-dimensional reconstruction can be performed through three kinds of algorithms: (i) the

3-D analytic algorithms,[8,9] where the whole volume is reconstructed at the same time [as in the Feldkamp-Davis-Kress (FDK) method[10] for 3-D cone-beam CT], (ii) the algebraic, or iterative, algorithms[11,12] improving the quality of the image at the cost of the computation time, and (iii) the rebinning algorithms, decomposing the 3-D reconstruction into a set of 2-D ones.[13–16] The advanced single-slice rebinning (ASSR) algorithm, which is discussed in this paper, belongs to the last category.[13]

Rebinning simplifies the reconstruction, but 2-D inverse Radon transformation is still the slowest step of the process. This operation can be carried out through many methods, but the filtered backprojection (FBP) algorithm seems to be the most commonly used.[11] The FBP algorithm can be decomposed into two steps: the filtering of the acquired data and the backprojection. This operation is very time-consuming, with a complexity of $O(N^{3/2})$, where N is the number of points in the reconstructed image. Some clever methods have been introduced in order to reduce the complexity of backprojection. One of the best-known is the linogram method, developed in 1987.[17] Recently, some other methods, based on the divide-and-conquer principle, have been published.[18–20] In 2003, a new inversion method based on the wavelet transform was introduced.[21]

An alternative way to speed up the execution of the FBP algorithm is to improve the hardware. There are some dedicated commercial solutions, embedding up-to-date electronic components, such as application-specific ICs (ASICs), or high-performance field-programmable gate arrays (FPGAs).[22–24] These solutions are expensive, and the materials used are not standard. A low-cost alternative consists in using the graphical process units of standard graphic cards, which permit speeding up the computation of the 2-D and 3-D inverse Radon transformations by an order of magnitude.[25,26] Though these methods show impressive speedup, they are unfortunately not sufficient to meet the expected needs of future computer-assisted surgery. The possibility an optical architecture dedicated to the FBP algorithm has also been investigated in the past, but technologies were not mature enough.[27–29] In the early twenty-first century, the emergence of new techniques and devices in optoelectronic has led to renewed interest in optical processing.

This research follows up a preliminary work on the purely optical aspects of the system, which has already been published in Ref. 30. In this paper, the focus is on the extension of this work to a 3-D algorithm and on the modeling of the whole (hybrid) processing system. As is underlined in the following sections, the computation of the FBP becomes faster than the other steps of the ASSR algorithm. Therefore, these other steps have been adapted and optimized, leading to the complete new hybrid architecture.

After a brief review of the ASSR algorithm, our hybrid architecture is detailed. Then, a complete model of the system, including timing and quality consideration, is described and leads to a consistent study in simulation. Before discussing and concluding on this work, simulation and experimental results are presented.

2 ASSR Theory

The ASSR is a reconstruction algorithm for cone-beam spiral CT.[13] It is not exact, but provides a good approximation of the explored volume as long as the cone angle remains small. It is composed of four consecutive steps: (i) the choice of the reconstruction plane, (ii) the rebinning of the acquired data in order to build a 2-D sinogram (a 1-D sinogram being equivalent to the Radon transform of the slice), which corresponds to a virtual parallel beam acquisition in the reconstruction plane, (iii) the 2-D reconstruction (i.e. computation of the inverse Radon transforms of virtual parallel beam sinograms), (iv) the generation of the reconstructed volume by interpolating the resulting 2-D planes. This algorithm is detailed in Ref. 13 and summarized hereafter. The notation is that used in Ref. 13, except for the point coordinates in the virtual parallel beam sinogram [(ρ,θ) instead of (ξ,ϑ) in Ref. 13].

2.1 *Choice of the Reconstruction Plane*

A reconstruction plane, indexed by k, is chosen to optimally fit a 180-deg spiral segment. As a consequence, it is tilted with respect to the horizontal Oxy plane. The equation of the k'th virtual reconstruction plane (R-plane) is given by

$$R(k): x\cos\alpha_R(k)\tan\gamma + y\sin\alpha_R(k)\tan\gamma - z + k\,\Delta z_R = 0, \quad (1)$$

where Δz_R is the vertical reconstruction pitch, γ is the tilt angle (defined by the scanning geometry[13]) and $a_R(k)$ is the reference angle of the k'th plane, defined by the following equation (d is the pitch of the helix): $a_R(k)=2\pi k\,\Delta z_R/d$.

2.2 *Rebinning*

Data are acquired by an x-ray sensor with N_u columns (parallel to the helix trajectory) and N_v lines. The coordinates (u,v) relate to the position on the moving sensor. A full scan is composed of N_α acquisitions, and one acquisition is indexed by α. Moreover, the position on a 2-D sinogram is given by the standard (ρ,θ) coordinate system. The rebinning can be mathematically defined by an application from to $\mathbb{R}^3$ to $\mathbb{R}^3$ that gives, for each triplet (ρ,θ,k), a triplet (u,v,α) so that the 3-D ray whose direction is given by (u,v,α) best fits the virtual 2-D ray defined by (ρ,θ) in the plane $R(k)$.

The equations are

$$u = \frac{R_D + R_F}{R_F}\frac{\rho}{[1-(\rho/R_F)^2]^{1/2}}, \quad (2)$$

$$v = \frac{R_D + R_F}{R_F}\frac{\rho\tan\gamma\cos[\theta+\sin^{-1}(\rho/R_F)]}{[1-(\rho/R_F)^2]^{1/2}}, \quad (3)$$

$$z = k\Delta z_R + \frac{d}{2\pi}\left(\theta + \sin^{-1}\frac{\rho}{R_F}\right), \quad (4)$$

where R_F is the distance between the focus (x-ray source) and the rotation axis, and R_D is the distance between the center of the detector and the rotation axis. To equalize the lengths of the two rays and to compensate the shift between computed pixel and voxel locations in the world Cartesian system, a correction factor is inserted. This factor depends only on ρ and θ.[13]

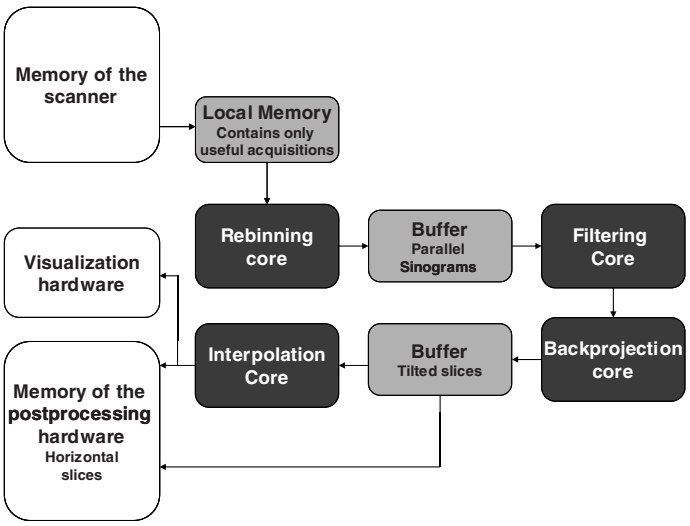

Fig. 1 Architecture overview and data flow.

The rebinning step provides a set of virtual parallel beam sinograms $R_k(\rho,\theta)$.

2.3 *2-D Reconstruction*

Reconstruction in the R-planes is performed using a standard 2-D algorithm, such as the FBP,[11] which can be decomposed into two steps: (i) the filtering of each column (θ constant) of the sinogram with a ramp filter $h(\rho)$, and (ii) the backprojection of the filtered data. These two operations can be defined by the following operators (**V** for the filtering and **B** for the backprojection):

$$\mathbf{V}:\mathbb{R}^2 \rightarrow \mathbb{R}^2 : f(\rho,\theta) \rightarrow f(\rho,\theta) * h(\rho,\theta) \tag{5}$$

and

$$\mathbf{B}:\mathbb{R}^2 \rightarrow \mathbb{R}^2 : f(\rho,\theta) \rightarrow g(x,y) = \int_0^{2\pi} f(x\cos\theta + y\sin\theta,\theta)\mathrm{d}\theta. \tag{6}$$

The reconstruction step provides a set of tilted slices $f_k(x,y)$ indexed by k.

2.4 *Interpolation*

Interpolation is used to calculate the voxel value in the world's Cartesian system from previously computed values in the R-planes. Thanks to the compensation coefficient added during rebinning (Sec. 2.2), pixels and interpolated voxels have the same position along the x and y axes. Therefore, a standard linear z interpolation is sufficient. The value of the voxel $f(x,y,z)$ can be calculated from the $f_k(x,y)$ distribution according to following weighted sum:

$$f(x,y,z) = \frac{\sum_{k=1}^{N_k} f(x,y) D_k(x,y)}{\sum_{k=1}^{N_k} D_k(x,y)}, \tag{7}$$

where $D_k(x,y)$ is the weighting function, which depends on the distance, along the z axis, between the target voxel and the reconstructed pixel in the k'th R-plane.

It should be noticed that the interpolation is application-driven and depends on the targeted information. Interpolated images are pretty for visual examination, but some postprocessing (the detection of an object in a volume, for instance) could be applied directly to the tilted slices. That will be the case if the interpolation can be integrated during the processing without excessive calculation cost.

3 Description of the Hybrid Processor

The execution of the ASSR algorithm is very time-consuming on standard hardware. For this reason, a hybrid (optoelectronic) processor has been designed to execute this algorithm. The system is composed of four independent cores: one for the rebinning, one for the filtering in the FBP, one for the backprojection, and a last one for the z interpolation. Backprojection, which is the slowest step, is performed through the use of an optical architecture. As underlined in Sec. 5, the speedup provided by this solution renders it pertinent to design a specific architecture for the other steps.

3.1 *Global Overview*

The four cores are pipelined and work in parallel (Fig. 1):

- building the n'th 2-D parallel-beam virtual sinogram from the acquired 3-D data set
- filtering the $n-1$st 2-D parallel-beam virtual sinogram

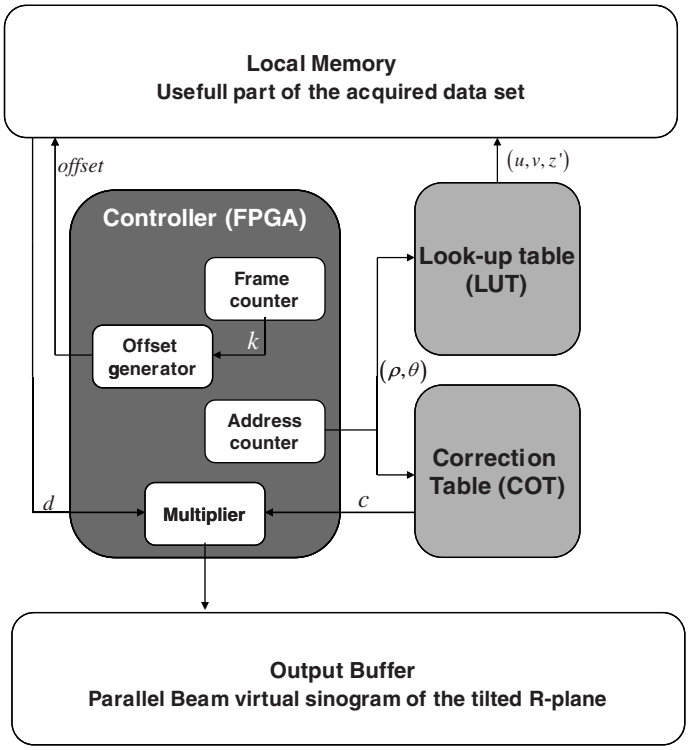

Fig. 2 Rebinning core.

- backprojecting the $n-2$nd 2-D parallel-beam virtual sinogram in order to reconstruct the tilted slice corresponding to the $n-2$nd R-plane
- Use of the $n-3$rd tilted slice in order to compute in parallel its contribution, pixel by pixel, to whole reconstructed volume.

As a consequence, the global computation time is determined by the slowest step. This point is further discussed in Sec. 4.

3.2 *Rebinning Core*

The rebinning core is composed of standard electronic devices: memories and FPGAs. The design of this part is simplified because u, v, and the correction coefficient do not depend on k. Moreover, z is linear in k. Thus, no complex computation is needed, and the rebinning can be performed with lookup tables (LUTs, associative memories) and correction tables (COTs). The content of the LUTs and the COTs is computed once at the initialization of the system and is a function of the reconstruction parameters. The architecture of this core is given by Fig. 2. A controller, which contains a frame counter, an offset generator, an address counter, and a Q-level pipelined 32-bit floating-point (FP) multiplier, drives the system. Two nonvolatile memories are used for the LUT and the COT. This core is highly pipelined, and all of the following elementary tasks are performed in parallel.

- Step 1: incrementation of the address counter
- Step 2: readout from the LUT of the address (u,v,z') corresponding to coordinates (ρ,θ)
- Step 3: readout from the local memory of the data d at the address $(u,v,z'+$ offset) and readout from the COT of the correction coefficient c at address (ρ,θ)
- Steps 4 to $4+Q-1$:FP multiplication between c and d
- Step $4+Q$:saving the result in the output buffer at address (ρ,θ).

In parallel, the local memory is updated with newly acquired data that should be used for the R-planes.

In addition, the system includes some buffers and FIFOs, required to compensate pipeline latency, which are not included in Fig. 2.

In practice, the analog values u,v,z,ρ,θ in Eq. (2)–(4) should be replaced by discrete ones. For this purpose, in this configuration of the rebinning core, the nearest neighbor approximation (NNA) is used. Although the limitation of that method has already been proven, we observed in simulation that the error introduced by it is not significant in comparison with the distortion introduced by the optical part. Nevertheless, if the NNA distortion were to become dominant, this architecture could be improved in order to allow finer approximation.

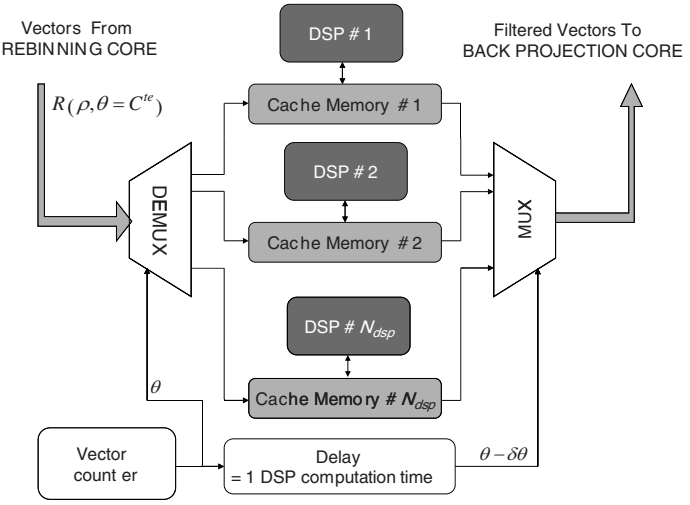

Fig. 3 Filtering core.

3.3 *Filtering Core*

Filtering can be performed on various hardware, but digital signal processors (DSPs)[31] seem to be the most suitable devices. The processing is sequential, and each column of the sinogram can be filtered independently. As a consequence, the filtering core is easily parallelizable by associating with it N_{dsp} DSPs, each of which will be in charge of one part of the sinogram (Fig. 3). The effective computation time can therefore be divided by N_{dsp}.

Furthermore, the possibility of using an optical filtering architecture (a Vander Lugt optical correlator, for instance[32]) has been investigated. This way, the filtering core and the backprojection core can be merged. After a meticulous analysis, including experimental tests, this solution has been given up because the poor speedup gain seems to be not worth the quality loss due to coherent light noises (speckle and parasitic interference fringes).

3.4 *Optical Backprojection Core*

The optical backprojection core (OBC) is the optical part of the hybrid processor. It is based on a geometrical interpretation of Eq. (6). Let $g(x,y)$ be the backprojection of the sinogram $f(\rho,\theta)$. Considering a discrete form of Eq. (6), one can write

$$g(x,y)=\sum_{k=0}^{N_\theta-1} f_k(\rho,\theta_k), \tag{8}$$

with $\rho=x\cos\theta_k+y\sin\theta_k$. From a geometrical point of view, a new coordinate system (O,x',y') is obtained by rotating the world coordinate system by an angle $-\theta$ and letting $\rho=x'$; then f_k depends only on x' and θ_k. One can consider this function as an image with uniform columns: indeed, each point of a column defined by the equation $x'=a$ has the grey level $f(a,\theta_k)$ of the sinogram. Therefore, backprojection can be interpreted as the sum of the previously defined images rotated by an angle θ_k. This operation, when implemented in electronic hardware, is very time-consuming, but it can be performed very fast using an optical calculator. The image rotation can be made with a Dove prism, and the accumulation of each image is done directly on the sensor (superposition by multiple exposures). The problem of displaying negative gray levels, which may occur in FBP due to the ramp (high-pass) filtering, is discussed in detail in Ref. 30. Several methods have been proposed to get round this problem. The optical processor described in Fig. 4 is an example of an optical architecture for the purpose. It is composed of a noncoherent uniform light source, a linear spatial light modulator used to display the projection, a beam expander, an image rotator, and an acquisition system. The beamexpander is composed of four cylindrical lenses, and the image rotator of a Dove prism mounted on the gantry of a motor. The passive display system coupled with the noncoherent light source could be replaced by an active system (an LED array, for example). A complete analysis of this architecture can be found in Ref. 30.

3.5 *Interpolation Core*

The interpolation core is very difficult to design, because it depends on the application. In this subsection, we propose a general-purpose approach for high-speed linear interpolation. The processor is composed of an association of *interpolation clusters* (ICs), described by Fig. 5. An IC computes the weighted sum, which corresponds to the following pseudocode:

$$D:=\max(e-|p\cdot e-z(i,j,k)|,0),$$

$$X_{\mathrm{interp}}(i,j,p):=X_{\mathrm{interp}}(i,j,p)+X_{\mathrm{tilted}}(i,j,k)\times D$$

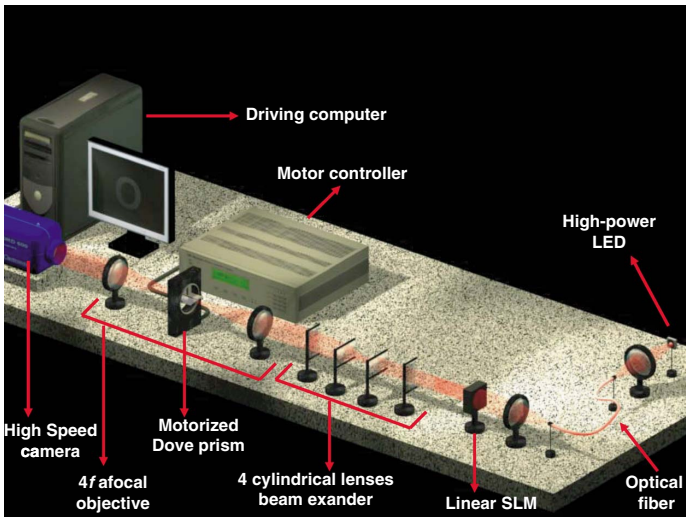

Fig. 4 The optical backprojection core (OBC).

$$C(i,j,p) := C(i,j,p) + D,$$

where D is the weighting coefficient calculated according to the distance between the interpolated voxel and the position of the pixel in the R-plane, e is the thickness of a horizontal slice, p is the index of the interpolated plane, $z(i,j,k)$ is the altitude of the pixel (i,j) in the k'th R-plane and is computed according the equation of the plane, and $X_{\text{interp}}(i,j,p)$ is the interpolated volume, $X_{\text{tilted}}(i,j,k)$ is the k'th tilted reconstructed plane, and $C(i,j,p)$ is the sum of the weighting coefficients for the point $X_{\text{interp}}(i,j,p)$. The operation is pipelined on 16 levels. The delay and FIFOs used to compensate the pipeline latency are not represented in Fig. 5. With one cluster per horizontal slice, it is possible to perform a highly parallelized z interpolation. The number of clusters can be optimized by considering that a tilted slice contributes only to a limited number of horizontal slices. The interpolation core incorporates a controller, a unit computing $z(i,j,k)$, and a weighting unit performing $X := X/C$ as soon as the slice interpolation is completed.

Again, the limitations of linear interpolation have already been highlighted. Nevertheless, we found that, for our application, it is not critical, because the distortion introduced at this step is weak in comparison of the distortion of the optical backprojector. This point asks for further investigations.

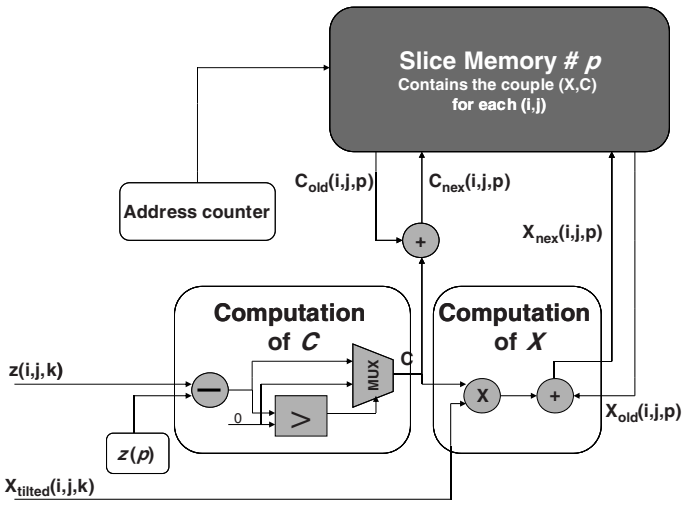

Fig. 5 A cluster in the interpolation core.

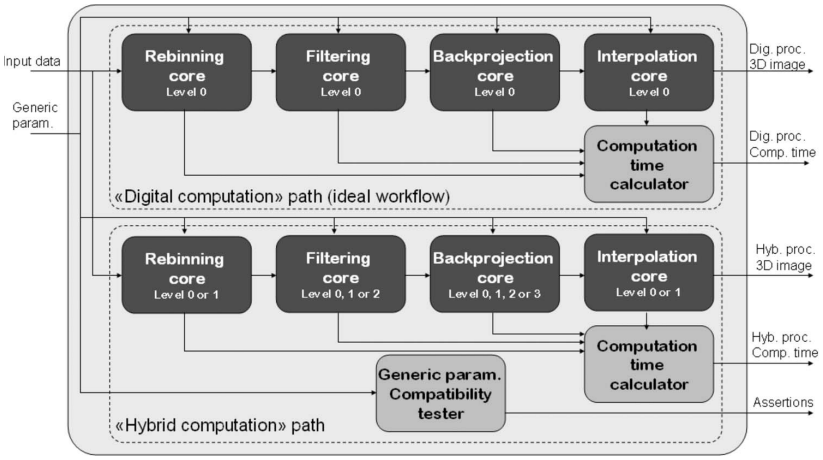

Fig. 6 Overview of the model.

4 Models

A model of the hybrid processor has been developed. It is a helpful tool for the design of the architecture and the evaluation of both computation time and reconstruction quality. We chose to develop a modular and multi-abstraction-level (Fig. 6) model implemented using MATLAB.

4.1 *Overview of the Model*

The model takes as an input the full acquisition set and returns:

- the computation time
- the 3-D image reconstructed with the proposed hybrid architecture simulator
- the equivalent digitally reconstructed 3-D image
- some assertions about the compatibility between reconstruction parameters and components used (the size of the linear display must be greater than the size of a projection, the size of the LUT memory must be greater than the number of points in a tilted slice, etc.).

The model includes about 50 generic parameters (GPs), which correspond to reconstruction parameters and embedded component characteristics. A nonexhaustive list of GPs is given in Fig. 7.

Each subpart of the model (Fig. 6) is described under different abstraction levels (Fig. 8). The equivalent digitally reconstructed image, which corresponds to the ideal workflow, is computed using the level 0 model for each entity.

4.2 *Computation-Time Model*

For level 0, we assume that the computation time of each step can be neglected in comparison with the computation time of the backprojection. This assumption become false when the optical core is used: level 1 or 2 is therefore used.

Electronic parameters	Optical parameters	Optoelectronic parameters	Computation parameters
FPGA	Lenses focal distances	**Linear SLM**	**Acquisition**
24-bit counter fmax	Optical fiber aperture	Frame rate	Number of lines on the sensor
FP Multiplier fmax	Distance between the lenses, the source, the linear SLM, the Dove prism and the camera	Resolution	Number of columns
FP adder fmax		Coding Depth	Number of acquisition per revolution
SRAM		Noise	
Access time		Pixel size	Number of revolutions
Number of bits	Dove prism dimensions	**Camera**	Helix pitch
Size of local memory	Source power	Frame rate	Angular increment
Size of output buffer	Sensor quantum efficiency	Resolution	Distance focus-centre
Size of DSP cache memories	**DSP**	Coding Depth	Distance focus-sensor
Size of buffers	Type	Noise	Radius of explored area
Size interpolation clusters	Computation time	Pixel size	**Reconstruction**
EEPROM	Number of DSPs	Number of acquisition per reconstruction (see the next section)	Angular increment
Access time	Number of bits		Number of projections on the virtual PB sinogram
Number of bits	**Data rates**	**Motor**	Number of points per proj
Size of LUT	Input data rate	Speed	Number of tilted slice
Size of COT	Electronics to optics interface	Positioning noise	Number of points in the interpolated volume.
DDRAM (scanner memory)	Optics to electronics interface	Misalignment	
Access time	Output data rate		

Fig. 7 List of the generic parameters of the model.

Rebinning core	Filtering Core	Backprojection core	Interpolation core
Level 0 FP rebinning computation (equivalent of standard computation) Insignificant comp. time **Level 1** Quantization effect Computation time Memory check	**Level 0** FP Digital filtering (equivalent of standard computation) Insignificant comp. time **Level 1** Computation time Memory check **Level 2** Quantization effect	**Level 0** FP Digital backprojection (equivalent of standard computation) Computation Time on digital processor **Level 1** Use of the geometrical interpretation (digital) Computation time on the optical architecture **Level 2** Quantization effects **Level 3** Alignment inaccuracy Checking the dimensions of the setup ("optical" model) Motor positioning inaccuracy ("opto-electronic" model)	**Level 0** FP Digital interpolation (equivalent of standard computation) Insignificant comp. time **Level 1** Computation time Memory check Quantization effect

Fig. 8 Description of the different abstraction levels for each model.

For this study, four reference run times are defined, one for each core:

- T_{reb} is the time the rebinning core needs to compute one new point in the virtual PB sinogram.
- T_{fil} is the time the filtering core needs to filter one vector of the virtual PB sinogram.
- T_{bp} is the complete time the backprojection core needs to reconstruct a 2-D slice from the virtual PB sinogram.
- T_{int} is the time the interpolation core needs to process one pixel of the tilted reconstructed slice (the time it needs to add a reconstructed pixel to all the horizontal slices it contributes to). It can also be defined as the computation time of one cluster.

The four cores operate in parallel and are used in a pipeline. Thus, the complete computation time can be described by following equation:

$$T_{fbp} = N_{\alpha_R} \max\left(N_\rho N_\theta T_{reb}, \frac{N\theta}{N_{dsp}} T_{fil}, T_{bp}, AN_x N_y T_{int}\right). \quad (9)$$

Here N_ρ, N_θ, N_{α_R}, N_x, and N_y are the reconstruction parameters (respectively, the number of points per projection, the number of angular projections per virtual PB sinogram, the number of R-planes for the whole volume, and the numbers of voxels along the x and y axes in the reconstructed volume), N_{dsp} is the number of DSPs used in the filtering core, and A is a coefficient defined as the ratio between the number of tilted slices needed to interpolate one horizontal slice and the number of available clusters in the interpolation core.

4.2.1 *Rebinning core*

The time T_{reb} is defined as that of the slowest process that is executed in parallel in the $(Q+4)$-level pipeline.

4.2.2 *Filtering core*

The time T_{fil} is estimated from the manufacturer's data sheet. For instance, Texas Instruments gives an equation estimating the Fourier transform run time according to the vector size.[31]

4.2.3 *Backprojection core*

For the backprojection core, we assume that the incremental rotation of the prism (time T_{mot}), the display of one projection (time T_{disp}), and the readout on the sensor (time T_{rea}) are parallelized and pipelined: Once the prism is placed, the vector is displayed, the camera is ready, and the system is lighted during a time T_{exp}. In this condition, the computation time T_{bp} is

$$T_{bp} = \max(N_\theta T_{disp}, T_{read}, N_\theta T_{mot}) + T_{exp}. \quad (10)$$

The exposure time is inversely proportional to the power of the light source. With the performance of up-to-date devices,[33] this time can be neglected. In practice, T_{disp}, T_{mot}, and T_{read} have the same order of magnitude. As a consequence, the contribution of the total read time is low compared to contributions of T_{disp} and T_{mot}. To optimize our system and to enhance the dynamic range of the reconstructed image, the optical and digital accumulations are combined: The reconstruction is obtained by the digital addition of N_{acq} subframes, each of them resulting from the optical accumulation of N_θ / N_{acq} images. The computation time becomes

$$T_{bp} = \max(N_\theta T_{disp}, N_{acq} T_{read}, N_\theta T_{mot}) + T_{exp}. \quad (11)$$

If N_{acq} is kept low, the global computation time is not penalized. However, high values for N_{acq} increase the image dynamic range. N_{acq} is briefly rediscussed in Sec. 4.3, but widely in Sec. 6.2.3.

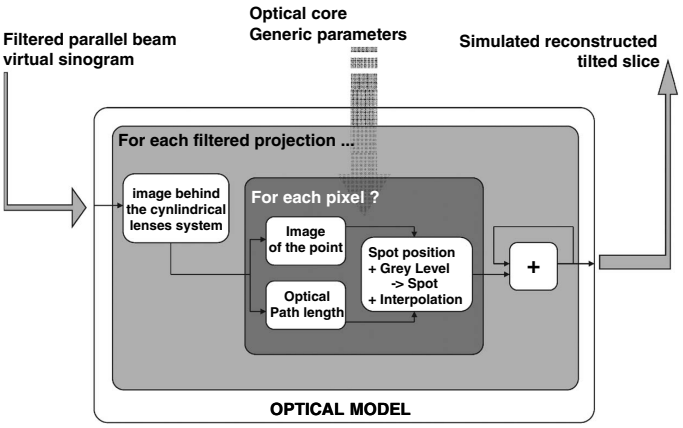

Fig. 9 Level 3, or optical, model of the backprojection core.

4.2.4 *Interpolation core*

The computation time of one cluster depends on its implementation on the FPGA. Here T_{int} is measured with a functional simulation on the FPGA programming tool (Quartus II).

4.3 *High-Level Quality Model (Behavioral Model)*

The *high-level* quality model includes the level 1 and level 2 descriptions of each model (level 3, the low-level quality model for the optical part, is discussed in Sec. 4.4). Let us analyze the first-order distortion source of each coprocessor.

4.3.1 *Rebinning and interpolation*

Rebinning and interpolation are digital operations that do not introduce noise, except the intrinsic noise of the method (the helical trajectory of the spot being approximated by a plane). For both cores, the level 1 model takes into account some quantization problems due to the data format conversion (32-bit to 16-bit, floating point to fixed point, etc.).

4.3.2 *Filtering*

The quality of filtering is very dependent on the data format that the DSPs can support. Furthermore, the computation time of a DSP increases with its accuracy.[31] Those facts justify the design of a level 2 model for this architecture.

4.3.3 *Backprojection*

The optical core is the critical part of the hybrid processor. Most of the distortions are introduced there. Noise comes from both quantization effects and optical setup misalignment. The level 2 model only takes the following degradation sources into account:

- *System input:* The display device works with integers, whereas the filtering core computes floating-point or fixed-point filtered projections.
- *Electro-optic interface:* The gray level displayed on the SLM is converted to a photon flow.
- *Optoelectronic interface:* The number of incident photons (and in their spatial distribution) is first recorded by the sensor. The device achieves the conversion to electrons and then to digital values (integers). The parameter N_{acq} permits one to enhance the coding depth at the output of the system.

4.4 *Low-Level Quality Model (Optical Model)*

Optical effects are taken into account in the level 3 model, which is presented in this subsection. In optical processing, major distortions come from the difficulty in aligning optical setups. To study this point, an optical model of the backprojection optical core has been designed. It takes the following phenomena into account:

- misalignment of the optical axis (passing orthogonally through the center of the input SLM plane) with the Dove prism axis
- misalignment of the Dove prism axis with the rotation axis of the motor
- misalignment of lenses (longitudinal displacement)
- defocus on the sensor plane due to the difference of optical paths introduced by previous effects.

The equations of the model are given in detail in Ref. 30 and are not presented in this paper, where the focus is on the modeling of the overall architecture. The level 3 optical model can be summarized by the workflow given in Fig. 9.

5 Simulation: Computation Time

The consistent and accurate model of the hybrid processor is used for an exhaustive study of the system in simulations. Timing and the quality aspects are analyzed.

First, let us analyze the computation time, which varies with many parameters. We consider a standard clinical case: a scan of the abdominal area with a Siemens SOMATOM Sensation 16 scanner.[6] This system uses a 672×16 x-ray sensor (today, 64-row sensors are standard), with a helical pitch of 18 mm, a 0.5-s rotation time, and an angular resolution of 1160 acquisitions per round. Data used for tests and provided by Siemens correspond to a 250-mm-radius and 170-mm-height cylindrical scanning area. This volume is reconstructed in a $512 \times 512 \times 180$ grid. In each

Table 1 Typical performance of up-to-date optoelectronic and electronic materials. Exposure time is neglected; this can be done as long as a high-power light source can be used.

Stage	Time	Type	Value	T_{bp}
Standard	T_{disp}	Nematic liquid crystal technology (≈100 frames/s)[35]	10 ms	1.8 ms
	T_{mot}	Standard rotation stage (≈720 deg/s)[36]	1.3 ms	
	T_{read}	Standard video camera (≈60 frames/s, N_{acq}=90)	16.6 ms	
State of the art	T_{disp}	Multiple quantum-well technology (~50,000 frames/s)[35,37]	20 μs	3.6 ms
	T_{mot}	Rotation stage a hard disk (~10,000 rpm)	16 μs	
	T_{read}	High-speed video camera (≈5000 frames/s, N_{acq}=18)[38]	200 μs	
Near future	T_{disp}	Vertical-cavity surface-emitting laser (~100,000,000 frames/s)[37]	0.01 μs	400 μs
	T_{mot}	Future high-speed high-precision rotation stage (≈80,000 rpm)	2 μs	
	T_{read}	Future high-speed video camera (≈20,000 frames/s, N_{acq}=8)	50 μs	

tilted plane, virtual parallel-beam sinograms with 180 projections of 672 points are used per reconstruction. The number of projections used in this example is small in comparison with standard values (~1000) used in digital reconstruction. In fact, however, beyond 180 projections there is no significant quality improvement with the optical processor, which is not the case on a digital processor. This point is discussed again in Sec. 6.2.

In 2004, for this process, the computation time on the dedicated reconstruction system (Syngo Explorer©) was about 2 slices/s.[34] Up-to-date commercial x-ray scanners claim to have reconstruction rates of up to 40 slices/s,[7] but are not usable for comparison, because no information is given about the resolution used to reach this figure.

Our analysis is based on the performance of up-to-date electronic and optoelectronic devices, which are described

Table 2 Estimation of computation time for the full ASSR algorithm.

Process	Time	Type	Value	Total time
Rebinning	T_{reb}	Simulated on a Cyclone II FPGA with a 32-bit floating-point multiplier pipelined on 5 levels. Critical time is the readout of the nonvolatile memory (Cypress nonvolatile RAM in this case).	25 ns	507 ms
Filtering	T_{filt}	Extracted from a benchmark of the ADSP-TS201[39]	64 μs	525 ms
		Use of 4 DSPs in parallel		
Backprojection	T_{bp}	See Fig. 7 (state-of-the-art value)	3.6 ms	648 ms
Interpolation	T_{inter}	Simulated on a Stratix II FPGA. The cluster is pipelined on 16 levels. RAM access time is the limiting factor. A=1.	12 ns	566 ms
Full reconstruction	T_{rec_3D}	Computation time of the hybrid processor		≈0.65 s
—		Comparison—computation time with Syngo Explorer[34]		≈90 s

in Table 1. Results are given in Table 2. The hybrid processor allows very high-speed reconstruction with a computational acceleration of about 140 in comparison with a standard technology. We can observe, in Fig. 7, that backprojection is still the slowest step in the global computation time, but is not predominant in comparison with the other cores. This means that, from a computation-time point of view, developments in the electronics part of the system are as important as the development in the optics one.

We can foresee that an extension, in the near future, to 1024×1024-pixel slices will be easy, because optoelectronic devices already exist with such a resolution. In addition, the speedup may be more impressive.

If we consider a reduced volume, say $512 \times 512 \times 16$, the computation time for the full reconstruction is about 60 ms (16 Hz), which is of the same order of magnitude as the targeted 3-D reconstruction rate. For the moment, a real-time system with the hybrid processor would be limited by the acquisition (0.5 s per revolution—about 2 Hz—to scan the $512 \times 512 \times 16$ volume). The data rates at the input and at the output of the system are not limiting. They are given by the following trivial equations:

$$D_{\text{in}} = \frac{N_\rho N_\theta N_{\alpha_R} d_{\text{in}}}{T_{\text{rec\_3D}}} \tag{12}$$

and

$$D_{\text{out}} = \frac{N_x N_y N_z d_{\text{out}}}{T_{\text{rec\_3D}}}, \tag{13}$$

where d_{in} and d_{out} are the coding depths of the data at the input and output of the system, and N_z is the number of voxels in the reconstructed volume along the z axis. In our case, inputs are single-precision floating-point data (32 bits), and outputs are integers with a 16-bit coding depth. The input and the output data flow are about 1 Gbit/s, which can be reached with standard media.

6 Simulation: Quality Analysis

6.1 *Preliminary Definitions and Quality Measurement*

Quality measurement is a hard but necessary task. First, we propose to clarify some notions used in the rest of the paper: coding depth, effective dynamic, dynamic range, and SNR. In medical imaging, images are quantified in Hounsfield units (HU), which belong to an arbitrary scale for describing radiodensity. Hounsfield units typically range from −1000 to +1000. In terms of digital imaging, the radiodensity of a voxel in a 3-D volume (or of a pixel in a 2-D slice) is likened to a gray level and is mapped, by a linear transformation, into the range 0 to $2^N - 1$, where N is the coding depth of the image (the number of bits used to write a quantified value). The dynamic range is the difference between the maximum value and the minimum one.

In the following, the distortion of the image is assumed to be white noise, which simplifies our approach. In practice, there are some spatial correlations in distortion, but simulations show that white noise remains a good approximation as long as $N_\theta \gg 1$. We define the effective dynamic (ED) as the number of accessible and meaningful levels. For example, let us consider an image encoded with 8 bits (gray levels between 0 and 255—the dynamic range is 255) and corrupted by a 5%-rms noise. If a pixel has an 8-bit measured value of 50, the true value is between 37 and 63 (if one assumes that the noise distribution is uniform). The coding depth is 8 bits, but the values of the last 4 bits are not significant. One says that the ED is only 4 bits. Due to the noise, there are only about 16 meaningful gray levels. The effective ED is more significant than the SNR, because it is independent of the image. These two parameters are linked by following equation:

$$\text{ED} = \frac{1}{6}\left[\text{SNR}_{\text{dB}} + 20 \log\left(\frac{D}{\sigma}\right) - 10.8\right]. \tag{14}$$

For instance, with a D/σ (ratio of the dynamic range of the image to its statistical power) of about 4, which is the case for the medical images used in this paper, a SNR of 30 dB corresponds to 36 meaningful levels, or 5.2 bits equivalent. If the image is encoded in 16 bits, computation gives the value of the radiodensity of one voxel (or pixel) of the image in the range −1000 to 1000 HU, in steps of 0.03 HU, but each value corresponds to a range ±28 HU. The last 10 bits of the 16-bit writing are meaningful. However, the dynamic range can be reduced to one from −30 to 150, which is about the useful window for organ exploration,[40] and radiodensities can be encoded with 8 bits, each value having an accuracy of ±5 HU.

Simulations performed with the proposed model provide both the reconstructed image and the distortion in comparison with the digitally computed image. With this information, one computes the SNR and deduces the effective dynamic of the process.

In many applications, those quantitative criteria are not the best adapted. The expected image quality is strongly linked with the application and the targeted postprocessing. This work is carried out in this paper, but is started on a very simple task: the segmentation. Specific quality criteria are introduced, as for instance the segmentation efficiency, which is the fraction of well-classified pixels on the image processed with the proposed method when the standard method is taken as reference.

6.2 *Quality—High-Level Analysis*

The model is also used to evaluate the quality of the reconstruction. At first sight, the optical backprojection core seems to be the greatest source of distortion. Unfortunately, the huge simulation time of the level 3 model of this architecture makes its use for the full ASSR algorithm difficult. For this purpose, the analysis is divided in two parts. In this subsection, the distortion is analyzed on the 3-D reconstruction with high-level models, and in the next one the focus is on the optical part with a simple 2-D case.

6.2.1 *Rebinning*

The only operation in this part is a multiplication, which can be implemented, on the FPGA, with a 32-bit floating-point multiplier without time loss in comparison with the data transfer time. Under this condition, the result has a quality comparable to the one obtained with a classic implementation.

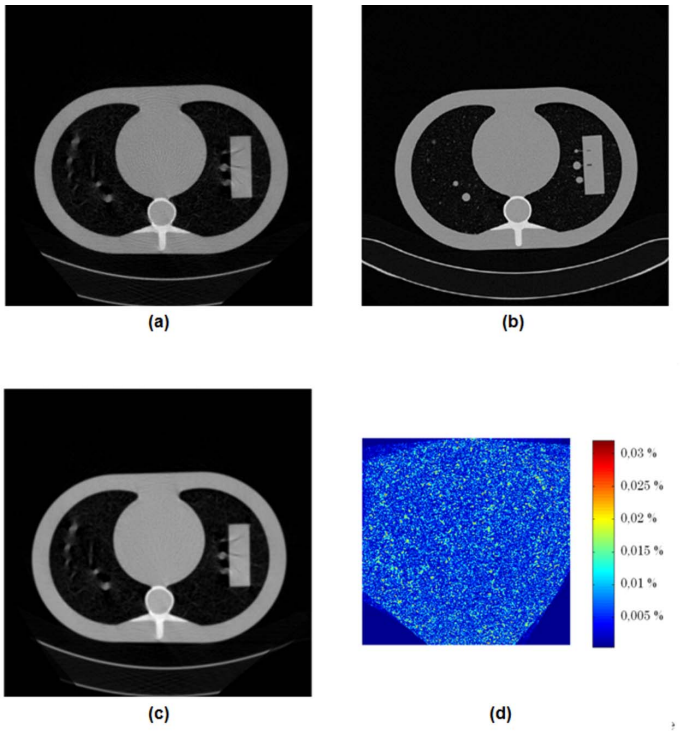

Fig. 10 Simulation: impact of the quantization. Examples of optically reconstructed images simulated with level 2 model (c) and the difference between those images and the digitally reconstructed ones (a) are represented for an 8-bit coding depth (SLM). (a) Horizontal slice reconstructed with the "level 0" model (equivalent to a full digital processing). (b) Reference image given by Siemens. (c) Horizontal slice reconstructed with the "high-level" models (quantization effects only). (d) Difference between (a) and (c).

6.2.2 *Filtering*

Filtering is a critical point. First, the ramp filter cannot be replaced by a convolution mask (even a large one) without huge distortion in the image. The use of the Fourier transform cannot be avoided. As a consequence, the use of a fixed-point DSP is not acceptable and introduces important distortions. A 32-bit floating-point DSP seems to be the better solution. Furthermore, with this component, computation time becomes critical, and many DSPs should be used in parallel. With a 16-bit floating-point DSP, the quality is lower but sufficient. If one takes only this noise into account, the SNR can be up to 100 dB.

6.2.3 *Backprojection*

We first consider the optic as ideal and only take into consideration the effect of the quantization (level 2 model). The SNR is analyzed as a function of the dynamic range of the input SLM, the dynamic range of the camera, and the parameter N_{acq}. Results are shown in Fig. 10. The images we obtain with level 0 and high-level models are similar, but we observe important differences from the reference image given by Siemens. In our view, the distortion is due to the beam-hardening effect,[11] which should not have been corrected on the raw data given by Siemens. There are many methods that permit reducing this effect. At first sight, we can assume that the preprocessing needed to compensate this phenomenon will not affect the global computation time. Nevertheless, this should be confirmed by an additional study.

We point out that the spatial distribution of the error is uniform for 8-bit and 4-bit display because systems (some edge effects appear with a 4-bit system). The noise is naturally filtered, because the algorithm includes a summation process. Then binary reconstruction (with a 1-bit system) works, but the image is not usable. The difference between the quantized filtered projections and the reconstructed image is about 30 dB. This improvement is to be linked with the number of projections used for the reconstruction—here, twice 180 projections (180 for positive values and 180 for negative ones) are used, and the theoretical improvement is $10\log(360) \approx 25.2$ dB.

The number of acquisitions reconstruction, N_{acq}, does not allow significantly improving the SNR, especially because of the input quantization. Actually the SNR on a single acquisition (60 dB for a 10-bit camera) is superior to the maximum SNR that can be obtained with an 8-bit quan-

tified display system (55 dB). However, increasing N_{acq} permits us to reduce the distortion due to the saturation on the camera.

6.2.4 *Interpolation*

Interpolation, like rebinning, is not a noisy step. Operations are carried out on 16-bit or 32-bit fixed- or floating-point data, which is sufficient in comparison with the resolution at the output of the optical core. Furthermore, the linear interpolation can be considered as a weighted mean, and as a consequence the SNR for interpolated images should be slightly higher than the SNR for the tilted slice. This has been observed in simulation, but the increase is not significant (between 0 and 2 dB).

6.2.5 *Overall system*

This analysis confirms that the optical backprojection core is the main source of distortion. Without taking optical distortions into account, quantization both at input (8-bit SLM) and at output (10-bit sensor) limits the SNR to about 55 dB, representing about 9 bits equivalent.

6.3 *Quality—Low-Level Analysis*

The optical backprojection core is the critical point of our architecture and the major distortion source. A narrow study of this part has been carried out with the level 3 model. Thus point is discussed in detail in Ref. 30, where the reader can find many simulation results. Here, only key results are summarized.

The misalignment of the optical setup introduce a blurring effect on the reconstructed image: Each pixel of the reconstructed image seems to be distributed over a ellipse, whose parameters depend on the amplitude of the misalignment. At first sight, each displacement seems to have a first-order effect on the SNR (decrease of 20 dB per decade). The simulation process allows us to specify the accuracy of the optical part. For example, we can estimate that an angular misalignment below 0.01 deg (on each degree of freedom) and a radial one below 1 pixel (on each degree of freedom) should be necessary in order to reach a SNR above 30 dB ($\approx$6 bits). Besides, this study seems to confirm that quantization noise can be neglected in the face of the misalignment noise.

The two main conclusions of the global simulation process are:

1. From the *computation-time* point of view, the pipelined structure is consistent and the processing times of the four cores are the same. The global processing time is more than two orders of magnitude better than on traditional hardware.
2. From the *quality* point of view, effort should be directed toward the optical core.

The model is also very useful for the realization of a prototype, especially the level 3 model of the OBC, which is used here during the alignment task, first to understand observed phenomena and then to facilitate their compensation.

7 Prototype and Preliminary Results

The study performed in simulation with the model of the hybrid processor gives encouraging results. Although it is very complete, the model cannot take disruptive phenomena into account, especially for the optical part.

7.1 *The Prototype*

We did not regard the development of a prototype including all the elements of the system as necessary, but we tried to validate each core separately. Rebinning and interpolation parts have been validated with functional simulation on a FPGA using dedicated software (Quartus), and filtering has been validated through the level 2 model implemented under MATLAB. For the last core, the OBC, which has been identified as the most sensitive part of the system, a prototype has been built.[30] It has been first tested only in a 2-D situation.

7.2 *Previous Results on the Optical Part*

The results are very promising for such a rough experimental prototype that is not optimized for practical reasons. The following results have been pointed out.[30]

Two kinds of distortions are observed: geometrical ones and radiometric ones (affecting gray levels). Geometrical distortions, related to borders (blurring), clearly predominate. On a synthesized shape (the Sheep-Logan phantom in this case), these distortions introduce a variation of about 5% of the geometrical parameters of the shape. Radiometric noise appears in uniform areas. It seems to be important (up to 10% rms) and can be explained by additional phenomena that are too complex to be taken into account in the model: saturation on the sensor, nonuniformity of the light beam, aberration, and complex optical effects, as well as some speckles and interference due to the partial coherence of the light source. (The spectral bandwidth of the LED is relatively large, but the coherence length of the emitted beam is sufficient to cause such texture.)

7.3 *Results on the Whole Hybrid 3-D Reconstruction Process*

We now consider the reconstruction of the 3-D volume, with the data provided by Siemens, using the prototype of the OBC. It appears that the image is corrective reconstructed except for a border effect. The effective dynamic is about 4 bits equivalent, which can be sufficient, in this case, to extract pertinent information from the result. As an example, here, the segmentation of the optical processed image is 93% accurate (Fig. 11). As observed in simulations, qualitative and quantitative results are comparable before and after interpolation.

8 Discussion and Conclusions

An alternative to speedup algorithms for helical cone-beam CT, based on an optoelectronic architecture, is proposed in this paper and is studied in simulation and with an experimental prototype. The architecture has a huge potential in computation power: the hybrid processor can be up to two orders of magnitude faster than a standard up-to-date digital processor. Such performance would be useful, and indeed necessary, for computer-assisted surgery, as discussed in the introduction. The architecture has a large capability of

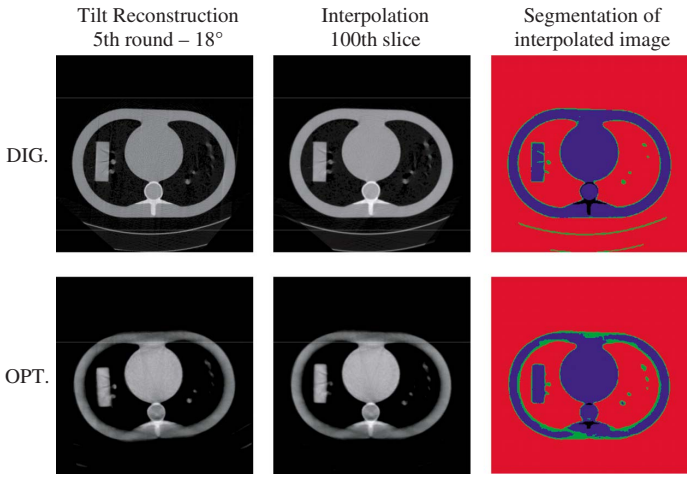

Fig. 11 Examples of reconstruction with the hybrid ASSR processor. Upper Row corresponds to digitally reconstructed image whereas Lower one corresponds to reconstruction with the prototype of the hybrid processor. Images on the left are tilted reconstruction corresponding to an acquisition at 18 deg in fifth round in the Siemens data set. Image in the center column corresponds to the 100th interpolated slice in the Siemens data set and images on the right correspond to segmentation result on the center image.

evolution (Table 1), and could benefit from the current progress of electronics and optoelectronics. In addition, it can be extended to other reconstruction algorithms using 3D-to-2D rebinning methods (AMPR,[14] SMPR,[15] CB-SSRB,[16] etc.) but not to direct 3-D algorithms,[8,9] for instance.

The backprojection, which has been identified as the most time-consuming operation in a fully digital system, is implemented on an optical processor. It is unsurprising that this core is the most sensitive part of the architecture and its processing quality is one of the biggest problems for future work. Our optical setup is only a rough prototype used to validate the method and has to be improved in order to envisage commercial exploitation.

The first present technological lock-in is the linear display. Only a few commercial devices meet our needs at present. Two technologies are considered: passive SLMs and active arrays (LED bars, for example). The first kind of display has been widely studied and improved for video applications (overhead projectors, wide-screen LCD television), but there are only a few breakthroughs for high-speed applications. Ferroelectric LCDS[41] have been the main innovation in this field in the past 10 years, but their speed limitation ($\sim$10-μs switching time) may be prohibitive in our system. Nowadays, a new technology is being developed, based on multiple quantum-well light modulators.[35,37] On the other hand, the active array may well be the most promising display technique. LED and organic LED (OLED) displays are being developed for many customer applications, and innovations in this field are promising. The refresh rate of such devices is only limited by the addressing process, which is greatly simplified for a 1-D array. The ideal display for our application would be a wide LED matrix with uniform lines. Such a device would be able to replace the linear SLM and the four-cylindrical-lens beam expander in the optical core (Fig. 4), and consequently would reduce the geometrical distortion in the due reconstructed image due to the limitation of computation power.

Second, the optical part, involving Dove prism, motor, and lenses, should be developed with special care. In order to achieve high reconstruction quality, this design should be optimized and integrated. Simulations shown in Sec. 6.2 and 6.3 highlight that the precision required for the alignment of those components in order to have good image quality is crucial and difficult to reach with standard means. For this purpose, an integrated optomechanical system should be preferred. Solid-state optics can be a good alternative too, but the motion of the Dove prism and the motor can be very challenging.

In spite of the difficulties encountered during the realization of the prototype, the results obtained with our rough prototype encourage us to carry on with this study. The main interest of this work lies in the approach to the challenging but important problem of the reconstruction time in medical imaging. For the first time, an optoelectronic processor is proposed to partially solve this problem. In spite of some obstacles for which further studies are required, the global processing architecture has been proposed, modeled, and validated experimentally. This work may be the preliminary step of the development of an industrial hybrid processor, which could be, in a near future, a very promising alternative to standard tools for high-speed tomographic reconstruction in order to meet challenging needs such as those arising in computer-assisted therapy.

Acknowledgments

This research was supported by Micro Module under a CIFRE contract, No. 414/2003, between the Micro Module

company and the InESS laboratory. The authors want to thank Mr. Jérémy Bartringer (R&D engineer at the InESS) for his involvement in this project, and especially his invaluable help in the realization of the prototype; Mr. Aurélien Mazel (engineer student at the Ecole Nationale Supérieure de Physique de Strasbourg), who also participated in the realization of the prototype; and Mr. Pamela Lhote (R&D assistant at the IRCAD) for a careful review of the English.

References

1. J. S. Duncan and L. H. Staib, "Image processing and analysis at IPAG," *IEEE Trans. Med. Imaging* **20**, 1505–1518 (2003).
2. A. Krupa, J. Gangloff, C. Doignon, M. de Mathelin, J. Leroy, L. Soler, and J. Marescaux, "Autonomous 3D positioning of surgical instruments in robotized laparoscopic surgery using visual servoing," *IEEE Trans. Rob. Autom.* **19**, 842–853 (2003).
3. N. Riviere, J. Gangloff, and M. de Mathelin, "Robotic compensation of biological motion to enhance surgical accuracy," *Proc. IEEE* **54**, 1705–1716 (2006).
4. J. R. Haaga and R. J. Alfidi, "Precise biopsy localization by computer tomography," *Radiology* **118**, 603–608 (1976).
5. P. Merloz, J. Tonetti, A. Eid, C. Faure, S. Lavallee, J. Troccaz, P. Sautot, A. Hamadeh, and P. Cinquin, "Computer Assisted Spine Surgery," *Clin. Orthop. Relat. Res.* **337**, 86–96 (1997).
6. Siemens Medical, Siemens SOMATOM Sensation 16 specifications, www.medical.siemens.com.
7. Philips Healthcare, Brilliance CT 64-channel scanner specifications, www.medical.philips.com.
8. M. Defrise and R. Clackdoyle, "A cone-beam reconstruction algorithm using shift-variant filtering and cone-beam backprojection," *IEEE Trans. Med. Imaging* **13** 186–195 (1994).
9. H. Kudo and T. Saito, "Derivation and implementation of a cone-beam reconstruction algorithm for nonplanar orbits," *IEEE Trans. Med. Imaging* **13**, 196–211 (1994).
10. L. A. Feldkamp, L. C. Davis, and J. W. Kress, "Practical cone beam algorithm," *J. Opt. Soc. Am.* **1**, 612–619 (1984).
11. A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*, 2nd ed., Society of Industrial and Applied Mathematics (2001).
12. H. Kudo, T. Rodet, F. Noo, and M. Defrise, "Exact and approximate algorithms for helical cone-beam CT," *Phys. Med. Biol.* **49**, 2913–2931 (2004).
13. M. Kachelriess, S. Schaller, and W. Kalender, "Advanced single-slice rebinning in cone-beam spiral CT," *Med. Phys.* **27**, 754–772 (2000).
14. S. Schaller, K. Stierstorfer, H. Bruder, M. Kachelriess, and T. Flohr, "Novel approximate approach for high-quality image reconstruction in helical cone beam CT at arbitrary pitch," *Proc. SPIE* **4322**, 113–127 (2001).
15. K. Stierstorfer, T. Flohr, and H. Bruder, "Segmented multiple plane reconstruction: a novel approximate reconstruction scheme for multi-slice spiral CT," *Phys. Med. Biol.* **47**, 2571–2581 (2002).
16. F. Noo, M. Defrise, and R. Clackdoyle, "Single-slice rebinning method for helical cone-beam CT," *Phys. Med. Biol.* **44**, 561–570 (1999).
17. P. Ehohn and G. Herman, "Linograms in image reconstruction from projections," *IEEE Trans. Med. Imaging* **6**, 301–307 (1987).
18. S. Basu and Y. Bresler, "$O(N^2 \log_2 N)$ filtered backprojection reconstruction algorithm for tomography," *IEEE Trans. Inf. Theory* **9**, 1760–1773 (2000).
19. T. Rodet, P. Grangeat, and L. Desbat, "Multichannel algorithm for fast reconstruction," *Phys. Med. Biol.* **47**, 2659–2671 (2002).
20. P. E. Danielsson and M. Ingerhed, "Backprojection in $O(N^2 \log_2 N)$ time," in *Proc. IEEE Nuclear Science Symp. '97*, Vol. **2**, pp. 1279–1283 (1997).
21. T. Rodet, P. Grangeat, and L. Desbat, "A fast 3D reconstruction algorithm based on the computation compression by prediction of zero-order wavelet," in *Proc. VIIth Int. Conf. on Fully 3D Reconstruction in Radiology and Nuclear Medicine*, Vol. **7**, pp. 135–138 (2003).
22. TeraRecon, Inc., www.terarecon.com.
23. Mercury Computer System Inc., cell accelerator board, MCJ Race++ VME mother boards, http://www.mc.com.
24. Mango DSP, Harrier cPCIDSP series, www.mangodsp.com.
25. F. Xu and K. Mueller, "Accelerating popular tomographic reconstruction algorithms on commodity PC graphics hardware," *IEEE Trans. Nucl. Sci.* **52**, 654–663 (2005).
26. V. V. Vlcek, "Computation of filtered backprojection on graphic cards," *Int. J. Signal Process.* **1**, 1–12 (2004).
27. M. Nishimura, "Implementation of the inverse Radon transform by optical convolution," *Opt. Commun.* **25**, 301–304 (1978).
28. T. Lu, S. S. Upda, L. Upda, "Optoelectronic implementation of filtered backprojection tomography algorithm," *Proc. SPIE* **1564**, 704–713 (1991).
29. T. Lu, "Optoelectronic system for implementation of iterative computer tomography algorithms," U.S. Patent No. 5,414,623 (1995).
30. M. Madec, J.-B. Fasquel, W. Uhring, P. Joffre, and Y. Hervé, "Optical implementation of the filtered backprojection algorithm," *Opt. Eng.* **46**, 108202 (2007).
31. Texas instruments, DSP Village, http://www.dspvillage.ti.com.
32. A. Vander Lugt, *Optical Signal Processing*, Wiley-Interscience, New York (1992).
33. Philips, Lumileds, LUXEON catalog, http://www.lumileds.com/.
34. A. Sennst, M. Kachelriess, C. Liedecher, S. Schmidt, O. Watzke, and W. A. Kalender, "An extensible software-based platform for reconstruction and evaluation of CT images," *Radiographics* **24**, 601–611 (2004).
35. U. Efron, *Spatial light Modulators: Material, Devices, and Applications*, CRC Press (1994).
36. Newport catalog, www.newport.com.
37. Lenslet Ltd., EnLight 256 arithmetic optical processor specification, www.dspdesignline.com/showArticle.jhtml?articleID=192200372.
38. Micron, CMOS sensor catalog, www.micron.com.
39. Analog Devices, DSP benchmark, http://www.analog.com/processors/tigersharc/overview/benchmarks/index.html.
40. G. N. Hounsfield, "Computed medical imaging," *Med. Phys.* **7**, 183–290 (1980).
41. K. Bauchert and S. Serati, "High-speed multilevel 512×512 spatial light modulator," *Proc. SPIE* **4043**, 59–65 (2000).

Morgan Madec received his PhD degree in electronics from the University Louis Pasteur of Strasbourg, France in 2006. From 2003 to 2006, he worked on the design and the characterization of optical processors in order to speed up image reconstruction in the medical field. Since 2006 he has pursued this research activity in the laboratory InESS (Institut d'Électronique du Solide et des Systèmes) in Strasbourg, in collaboration with Micro Module, Brest, France.

Jean-Baptiste Fasquel received his diploma from the general engineering school ENSPS in France, and his diploma for advanced studies in applied optics and computer vision from the University of Strasbourg in 1998. He obtained his PhD in digital and optical image processing from the University of Strasbourg, France, in 2002. Since 2002, he has been a permanent researcher in computer sciences at the Research Institute Against Digestive Cancer (IRCAD, Strasbourg). His main areas of interest are medical image processing and software engineering for computer-guided surgery.

Wilfried Uhring received his PhD degree in electronics from the University Louis Pasteur of Strasbourg, France, in 2002. From 1999 to 2002, he worked on the conception and improvement of ultrafast optical detection devices such as streak cameras. Since 2003 he has pursued this research activity in order to develop a solid streak camera within a simple CMOS chip. Since 2004, he has extended this research to the use of fast video CMOS sensors in optoelectronic processing. In January 2005, he joined the new laboratory InESS (Institut d'Électronique du Solide et des Systèmes).

Pascal Joffre received his PhD degree in optoelectronics from the Institut National Polytechnique Grenoble France in 1991. From 1983 to 1998, he developed liquid-crystal devices for infrared and visualization in the Thomson-CSF central research laboratory. He was in charge of commercial aspects, intellectual property, and development tasks. In 1999, he joined Micro Module, where he is director and sales manager.

Yannick Hervé received his Agrégation degree from Ecole Normale Supérieure in electrical sciences in 1985, his PhD degree in computer science and electronics in 1988, and his HDR degree in 2003. He is an associate professor at the University Louis Pasteur in Strasbourg, France. His research work in the field of VHDL-AMS virtual prototyping is within the Department of Complex Systems Modeling and Simulation at the InESS-UMR 7163 Laboratory in Strasbourg, France. He was a co-founder of Systems' Virtual Prototyping Company.