Graph- based segmentation of skeletal striated muscles in NMR images

Pierre-Yves Baudin

▶ **To cite this version:**

Pierre-Yves Baudin. Graph- based segmentation of skeletal striated muscles in NMR images. Signal and Image processing. Ecole Centrale Paris, 2013. English. NNT : 2013ECAP0033 . tel-00858584v2

HAL Id: tel-00858584

https://theses.hal.science/tel-00858584v2

Submitted on 26 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

École Centrale des Arts et Manufactures de Paris

THÈSE

présentée par

Pierre-Yves Baudin

pour l'obtention du

GRADE de DOCTEUR

Spécialité : Traitement des Images

Laboratoire d'acceuil : Centre de la Vision par Ordinateur

Sujet :

De la segmentation au moyen de graphes d'images de muscles striés squelettiques acquises par RMN

Soutenue le :

devant un jury composé de :

GRAPH-BASED SEGMENTATION OF SKELETAL STRIATED MUSCLES IN NMR IMAGES

PIERRE-YVES BAUDIN

PhD Thesis in Computer Vision
April 2013

Center for Computer Vision
École Centrale Paris

Institute Of Myology

Siemens Healthcare

ABSTRACT

Segmentation of magnetic resonance images (MRI) of skeletal striated muscles is of crucial interest when studying myopathies. Diseases understanding, therapeutic follow-ups of patients, etc. rely on discriminating the muscles in MRI anatomical images. However, delineating the muscle contours manually is an extremely long and tedious task, and thus often a bottleneck in clinical research. Typical automatic segmentation methods rely on finding discriminative visual properties between objects of interest, accurate contour detection or clinically interesting anatomical points. Skeletal muscles show none of these features in MRI, making automatic segmentation a challenging problem. In spite of recent advances on segmentation methods, their application in clinical settings is diffcult, and most of the times, manual segmentation and correction is still the only option.

In this thesis, we propose several approaches for segmenting skeletal muscles automatically in MRI, all related to the popular graph-based Random Walker (RW) segmentation algorithm. The strength of the RW method relies on its robustness in the case of incomplete contours and its efficient optimization. Originally, the RW algorithm was developed for interactive segmentation: the user had to pre-segment small regions of the image – called *seeds* – before running the algorithm which would then complete the segmentation. Our first contribution is a method for automatically generating and labeling all the appropriate seeds, based on a Markov Random Fields formulation integrating prior knowledge of the relative positions, and prior detection of contours between pairs of seeds. A second contribution amounts to incorporating prior knowledge of the shape directly into the RW framework. Such formulation retains the probabilistic interpretation of the RW algorithm and thus allows to compute the segmentation simply by solving a large sparse linear system, like in the original method. In a third contribution, we propose to develop a learning framework to estimate the optimal set of parameters for balancing the contrast term of the RW algorithm and the different existing prior models. The main challenge we face is that the training samples are not fully supervised. Specically, they provide a hard segmentation of the medical images, instead of the optimal probabilistic segmentation, which corresponds to the desired output of the RW algorithm. We overcome this challenge by treating the optimal probabilistic segmentation as a latent variable. This allows us to employ the latent Support Vector Machine (latent SVM) formulation for parameter estimation.

RÉSUMÉ

La segmentation d'images anatomiques de muscles striés squelettiques acquises par résonance magnétique nucléaire (IRM) présente un grand intérêt pour l'étude des myopathies. Elle est souvent un préalable nécessaire pour l'étude les mécanismes d'une maladie, ou pour le suivi thérapeutique des patients. Cependant, le détourage manuel des muscles est un travail long et fastidieux, au point de freiner les recherches cliniques qui en dépendent. Il est donc nécessaire d'automatiser cette étape. Les méthodes de segmentation automatique se basent en général sur les différences d'aspect visuel des objets à séparer et sur une détection précise des contours ou de points de repère anatomiques pertinents. L'IRM du muscle ne permettant aucune de ces approches, la segmentation automatique représente un défi de taille pour les chercheurs.

Dans ce rapport de thèse, nous présentons plusieurs méthodes de segmentation d'images de muscles, toutes en rapport avec l'algorithme dit du marcheur aléatoire (MA). L'algorithme du MA, qui utilise une représentation en graphe de l'image, est connu pour être robuste dans les cas où les contours des objets sont manquants ou incomplets et pour son optimisation numérique efficace. Dans sa version initiale, l'utilisateur doit d'abord segmenter de petites portions de chaque région de l'image, appelées graines, avant de lancer l'algorithme pour compléter la segmentation. Notre première contribution au domaine est un algorithme permettant de générer et d'étiqueter automatiquement toutes les graines nécessaires à la segmentation. Cette approche utilise une formulation en champs aléatoires de Markov, intégrant une connaissance à priori de l'anatomie et une détection préalable des contours entre des paires de graines. Une deuxième contribution vise à incorporer directement la connaissance à priori de la forme des muscles à la méthode du MA. Cette approche conserve l'interprétation probabiliste de l'algorithme original, ce qui permet de générer une segmentation simplement en résolvant un numériquement un grand système linéaire creux. Nous proposons comme dernière contribution un cadre d'apprentissage pour l'estimation du jeu de paramètres optimaux régulant l'influence du terme de contraste de l'algorithme du MA ainsi que des différents modèles de connaissance à priori. La principale diffculté est que les données d'apprentissage ne sont pas entièrement supervisées. En effet, l'utilisateur ne peut fournir qu'une segmentation déterministe de l'image, et non une segmentation probabiliste comme en produit l'algorithme du MA. Cela nous amène à faire de la segmentation probabiliste optimale une variable latente, et ainsi à formuler le problème d'estimation sous forme d'une machine à vecteurs de support latents (latent SVM).

ACKNOWLEDGEMENTS

CONTENTS

1 INTRODUCTION

1.1 CONTEXT AND MOTIVATIONS

Magnetic resonance Imaging

Magnetic Resonance Imaging (MRI) is a medical imaging technique providing 2D or 3D views of the internal structures of the body. MRI makes use of the principle of Nuclear Magnetic Resonance (NMR) which allows to measure some properties of the nucleus of some specific atoms in presence of a powerful magnetic field. MRI is non-invasive and implies non-ionizing radiations (e.g. unlike X-rays) allowing long and repeated acquisitions with no hazard posed for the health of patients.

Neuromuscular Diseases

Neuromuscular diseases may involve disorders in the motor neurons, in cells of the spinal cord, in peripheral nerves or in the muscles themselves. There exist both hereditary – dominant or recessive – and acquired – inflammatory, toxic, etc – disorders. Among the features assessed by clinicians to characterize neuromuscular diseases one finds: time of onset, chronicity, presence or absence of pain, muscle weakness and distribution of the involvement – proximal, distal, facial, symmetric or asymmetric, hypertrophical or atrophical, etc.

Clinical MRI is a non-invasive and powerful tool to characterize skeletal muscle involvement in neuromuscular disorders. This imaging technique has progressively established itself as a reference technique for the initial assessment of pathologies and for the evaluation of therapeutic intervention. In the NMR Laboratory of the Institute of Myology in Paris, several research projects focus on developing biomarkers for skeletal muscle disorders based on MR images: relative image intensity, intensity histogram, signal heterogeneity and more sophisticated methods of texture analysis, geometrical analysis (distance, area, volume, shape), etc. Moreover, some NMR parameters are intrinsic to the different types of tissues and can be quantified: T_1, T_2, proton density, $T_2^\star$, $T_{1\rho}$ and $T_{2\rho}$, D (diffusion constant), etc; We refer the reader to specialized publications – such as Fleckenstein et al. (1996) – for further information on the topic of muscle MR imaging; a brief introduction is also provided in the appendices of this document (see appendix A). In Fig. 1.1.1, we display MRI axial cross-sections of healthy subject and patients, showing the varied visual patterns of some myopathies.

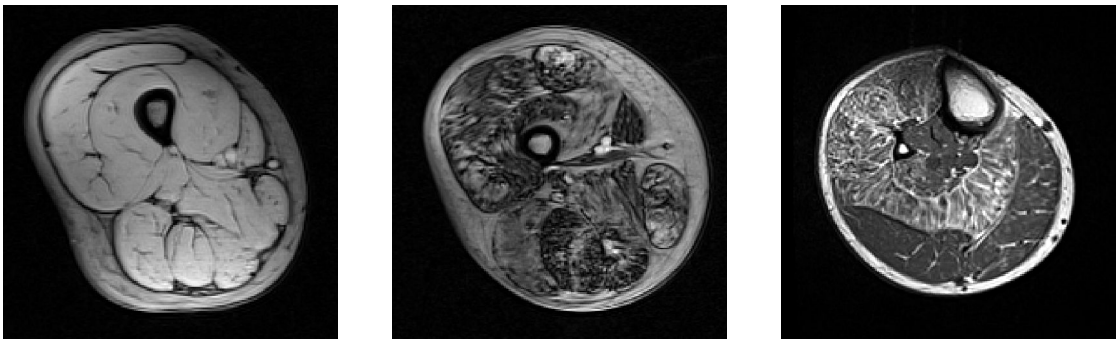

Figure 1.1.1 – Various axial cross-sections: (left) healthy subject (Dixon imaging); (middle) patient with centronuclear myopathy (Dixon imaging) – mutation of the MTM1 gene coding for the myotubularin; (right) patient with centronuclear myopathy (T_1-weighted imaging) – unknown mutation.

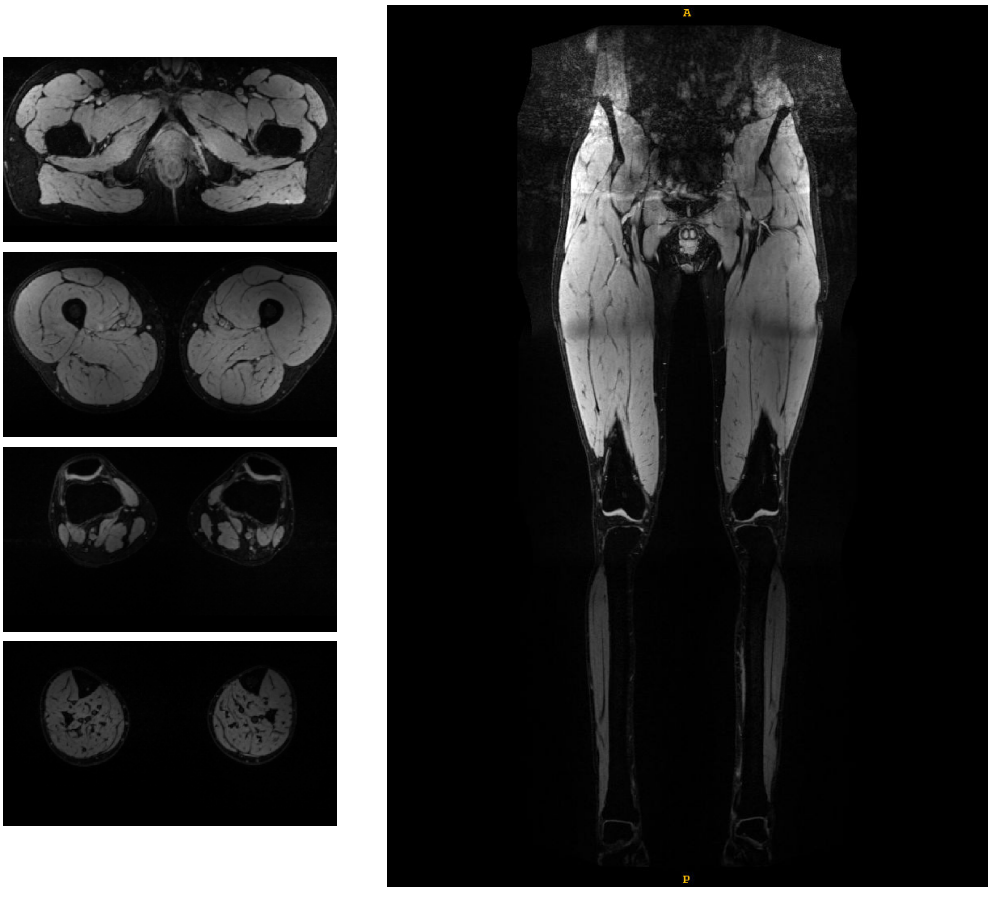

Figure 1.1.2 – (left) Several axial cross-sections of a 3D MRI volume of the lower limbs ($T_2^\star$ imaging); (right) coronal cross-section of the same volume.

Figure 1.1.3 – Image segmentation example. (left) Original image; (right) Segmented image. This segmentation is visualized by a boundary (red curve) separating the two regions and different colors assigned to the pixels, one per segment.

By displaying the anatomical structures – bones, vessels, fat and muscle tissues, etc. (see Figs. 1.1.2 and 1.1.1) – anatomical MR imaging allows to visually discriminate between the muscles and other organs, and thus to perform *segmentation*.

Image Segmentation

"In computer vision, *segmentation* is the process of partitioning a digital image into multiple segments (sets of pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics"[1].

Fig. 1.1.3 shows an example of image segmentation where a person in the picture is separated from the background.

Segmentation methods are needed to facilitate the identification of a muscle or of a muscle group. This step is particularly crucial when muscles have to be investigated selectively: for instance, when one wants to determine the NMR properties of each muscle group and study the differences between them. It is also necessary to measure the volume of the muscles and monitor their evolution.

Today, segmentation of muscle images is performed manually, by drawing regions of interest on the images. Such task requires enormous amounts of time, and is an extremely tedious and tiring process. Due to the advances in acquisition techniques and

1 Wikipedia, on 01.01.2013

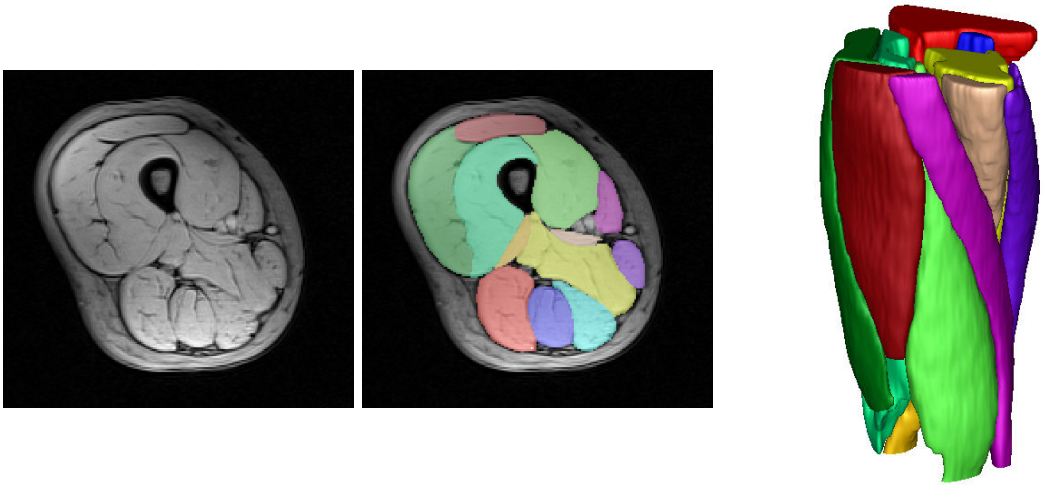

Figure 1.1.4 – (left) Axial cross-section of an MRI volume of a right thigh (Dixon imaging); (center) same cross-section with all muscles manually segmented; (right) 3D segmentation of the same volume.

computer storage facility, it is common to acquire MRI 3D volumes composed of more than 50 slices. As a result, manually segmenting a volume of a thigh – which comprises more than 15 muscles – represents more than four hours of work. Moreover, during clinical research protocols, patients are often scanned with several sequences and very large amount of images can be generated. As acquiring images becomes easier and faster for large volumes, segmentation of muscles images is becoming more and more a bottleneck in the chain of analysis. Besides, manually processing these data is not only time consuming, it is also highly observer-dependent: different experts produce significantly different segmentations, especially in places where the boundaries of the organs are not clearly visible. Fig. 1.1.4, shows an example of manual segmentation of the muscles in anatomical MRI.

For these reasons, it is highly desirable to possess automatized segmentation tools to process the data more quickly and consistently and thus facilitate their interpretation. This is a crucial objective to sustain the growth of applications of skeletal muscle MRI in the field of neuromuscular disorders.

Whereas many automatized methods have been developed for segmenting images of brains and hearts, very few attempts were made on skeletal muscles. Medical studies on the brain and the heart are more developed due to the existence of widespread diseases associated with theses organs, whereas severe muscle diseases are often genetic and fall in the category of orphan diseases. The specifics of muscle segmentation prevent using directly the methods developed for other organs and require specially adapted methods. Moreover, a large part of the existing publications on muscle seg-

mentation focus mainly on the fat-muscle tissue separation rather than on muscle group segmentation itself (Al-Attar et al., 2006; Barra and Boire, 2002; Mattei et al., 2006)

In this thesis report, we present our work on developing new automatic image segmentation methods adapted to MRI images of skeletal muscles.

1.2 PROBLEM DEFINITION

Any automatized segmentation method relies on (a) finding consistent features of targeted objects – e.g. shape or appearance descriptors – (b) allowing some variability in the target features and (c) defining means of overcoming anticipated difficulties – e.g noise in images:

- consistent features of the skeletal muscles in MRI are (cf. Fig. 1.1.4):
  - fixed relative positions – e.g. muscle *a* is always *top left* of muscle *b*;
  - interweaved positions, closely entangled – e.g. muscle *a* is always *adjacent* to muscle *b*;
  - shapes, to some degree – e.g. muscle *a* can be visually *identified* based on prior anatomy knowledge;
  - smooth boundaries, to some degree, especially in the direction parallel to the limbs – i.e. muscles can be visually separated from each other based on visible boundary structures.
  - muscle/background appearances – i.e. muscles can be visually distinguished from other tissues based on their appearance.
- variability is found in (cf. figure 1.2.1):
  - shapes, with large inter-subject variations of the muscles size and shapes.
- anticipated difficulties are (cf. figure 1.2.1):
  - non-discriminative inter-muscle appearance – e.g. one cannot distinguish muscle *a* from muscle *b* based on intensity/texture differences;
  - missing boundaries – e.g. muscle *a* is sometimes not separated from muscle *b* by any visible structure;
  - spurious contours – e.g. muscle *a* can present internal structures visually similar to boundaries.

Considering these characteristics, our aim is to design semi-automatic or automatic segmentation methods with the following features:

- multi-object;

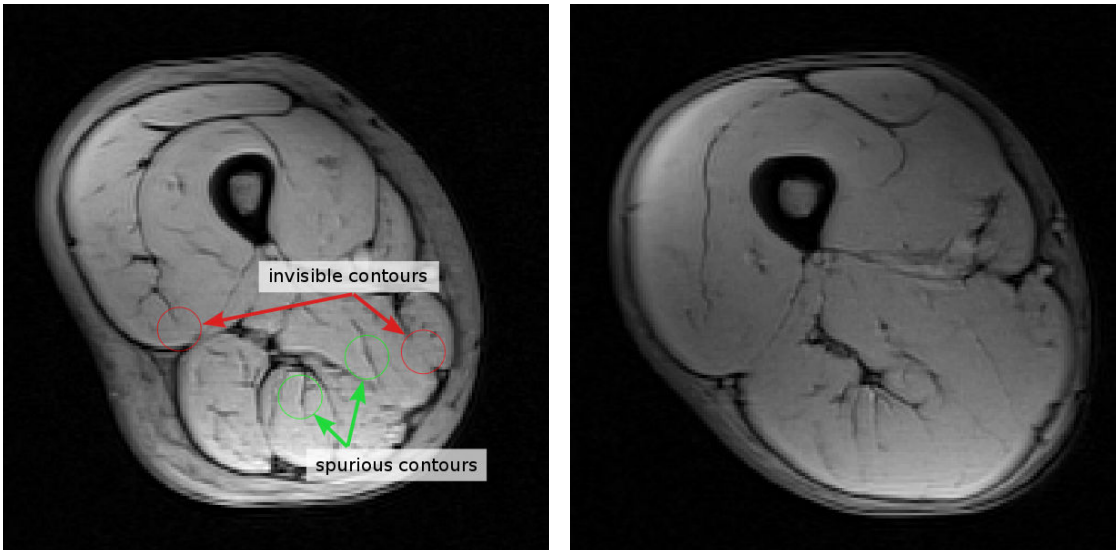

Figure 1.2.1 – Missing and spurious contours in muscle image. (left) Cross-section of a thigh from a 37 year-old female subject; (right) same height cross-section from a 29 year-old male subject. Notice the differences in shape and proportion of the muscles between the two subjects.

- relying solely on:
  - detectable boundaries;
  - prior knowledge of the muscle shapes and topology;
  - muscle/background intensity differences;
- robust to:
  - missing boundaries;
  - missing contours;
  - shape variations.

1.3 CONTRIBUTIONS

In this thesis report, we present several contributions related to the general Random Walks (RW) for image segmentation algorithm (Grady, 2006). This algorithm has the advantage of relying solely on boundaries while being robust to partial contours, and provides a simple multi-object segmentation framework with no topological issues (that is, no intersection issues).

Originally, the RW algorithm was developed for interactive segmentation: the user had to pre-segment small regions – called *seeds* – of the image to be segmented be-

fore running the algorithm which would then complete the segmentation. Our first contribution is a method for generating and labeling appropriate seeds automatically. This approach exploits the fact that relative positions of the muscles are consistent among individuals, even though the large inter-subject variations of skeletal muscles prevent using a deformable model based on a shape template. Thus, it is reasonable to develop a segmentation method whose prior information is not based on the position of the boundaries and on the shape of the objects, but rather on the approximate relative position of the objects and the existence of boundaries between them. Seeds are compatible with this idea: they are generally placed well inside the target object so that the actual boundaries are determined by the Random Walks algorithm. Thus, our method aims at placing the seeds automatically, based on prior knowledge of the relative positions and on the detection of contours between pairs of seeds. We have the following stages: 1) generate unlabeled seeds within homogeneous regions of the target image; 2) build a graph over the sampled seeds, viewing each seed as a node and adding edges between nearby ones; 3) assign a label to each seed automatically by minimizing an functional defined on the graph nodes and edges; 4) the image is segmented with the RW method using the newly generated seeds. During the labeling stage 3), the node-wise potentials depend on prior-knowledge over label assignments based on a comparison of the target image with a registered reference annotated image. The pair-wise potentials depend on the likelihood of the edge orientation and on the detection of boundaries between the seeds, where both quantities can be obtained by statistical learning over the reference image. This labeling problem is solved using an efficient discrete energy minimization method based on a belief propagation algorithm (Kolmogorov, 2006). Due to the non-specific (non sub-modular) formulation of the pair-wise potentials, the labeling cost functional can only be minimized approximately; however, we present experiments on real clinical data showing that the approximate solution is better than not using pair-wise potentials. This method was published in the proceedings of the 2012 ISBI conference (Baudin et al., 2012c).

This first method has the inconvenience of being two-stage: if the seed labeling fails, then the segmentation will irremediably fail in the same area. Fortunately, it was shown that it is possible to completely bypass the seed placement stage by incorporating prior-knowledge on the intensity to the RW formulation (Grady, 2005). However, intensity is not a reliable feature in the case of muscles separation. Our second contribution consists in proposing several models mixing Random Walks and prior knowledge of the shape. The Random Walks segmentation algorithm is a graph-based method working at the pixel level: each pixel is viewed as a node and edges are defined between each adjacent pixels; the weights of the edges are set according to the strength of the contrast between the connected pixels. The aim is to assign a label (e.g. the index of a muscle) to each pixel. Segmentation is achieved by minimizing a quadratic objective functional based on the combinatorial Laplacian matrix of the image, which amounts to solving a large sparse linear system. In its original

version, a unique solution is found only if there exists at least one seed per label. In our approach, a prior term depending on the squared distance of the segmentation to a statistical atlas is added to the RW functional. We point out that such formulation does not break the probabilistic explanation of the RW method, as the added prior term can be seen as adding implicit seed nodes. Noting that each pixel's prior can be weighted independently from the others, we propose several weighting schemes, either based on statistics over the training set or on information from the target image itself. This method was published in the proceedings of the 2012 MICCAI conference (Baudin et al., 2012a). In a second approach, published in the proceedings of the 2012 BMVC conference (Baudin et al., 2012b), we propose to further exploit our training set by building a statistical shape space – a low dimensional space of valid solutions – through Principal Component Analysis. This is motivated by the need to increase the flexibility of our model so that it can adapt to more diverse morphologies. The segmentation is no longer anchored to a unique average atlas like in the previous approach, but is constrained to remain "close" to the shape space, which models many possible valid segmentations. All approaches are validated on real clinical data.

While it is interesting to determine which model gives better results on the muscle data, it is also desirable to combine them together – e.g. using different weighting schemes plus the intensity prior. Moreover, we could also want to use several weighting functions for setting the graph edges in order to capture varied properties of the images simultaneously. However, achieving such generality requires to set balancing weights for a large number of parameters, which would certainly not be optimal if done manually. As a third contribution, we propose to develop a learning framework to determine the optimal set of parameters for balancing the contrast term of the RW algorithm and the different prior models. Our approach is guided by noticing the RW formulation bears important similarities with the more wide-spread pixel-based Markov Random Fields (MRF) (Boykov and Jolly, 2001) segmentation methods. In recent work, machine learning methods – such as Support Vector Machines (SVM) – were used to fit the parameters of the segmentation method to the data type of interest (Szummer et al., 2008). We propose to adapt the MRF/SVM learning scheme to the RW case. We explain why a direct transposition in the RW case of the previous formulation cannot solve the problem, due to having only a "hard" – i.e. binary – annotation vector for our training data whereas a "soft" – i.e. probabilistic – one is necessary. We overcome this issue by introducing an implicit soft annotation vector, which amounts to solving a latent SVM problem. Due to the large number of variables rendering certain stages of the problem too difficult to solve exactly, we propose approximation schemes in the "Annotation Consistent Inference" (ACI) stage and in the "Loss Augmented Inference" (LAI) stage. Using the same clinical data as previously described, we present preliminary results which demonstrate the potential of this learning approach for the prior-enhanced Random Walks segmentation method.

1.4 THESIS PLAN

This report is organized as follows: in chapter 2 we present a detailed state-of-the-art of segmentation methods in medical imaging; the next three chapters present our contributions to the field: automatic seed placement in chapter 3, introducing shape priors to the Random Walks framework in chapter 4 and developing a SVM-based learning scheme for improving the latter models in chapter 5; the last part concludes this report.

2

STATE OF THE ART IN SEGMENTATION

To address the problem of image segmentation, a wide variety of techniques was presented and one can find a very abundant literature. However, few research works have dealt with muscle image segmentation to the best of our knowledge. In this chapter, we review the existing segmentation methods applied to medical images and their optimization procedures, with a special attention paid on those which were designed to address muscle segmentation.

Almost all segmentation methods have in common is how they rely on a *cost functional,* or *energy functional,* $E(M, I)$ which measures how well a proposed segmentation model M explains the target image I. In other words, an energy functional is designed to measure the adequacy of the segmentation model with respect to a combination of criteria, such as alignment of the model surface with image edges, intensity homogeneity within model regions, plausibility of model shape, etc. The better the model M fits the image I, the lower is $E(M, I)$. Depending on the chosen approach, $E(\cdot, I)$ can be convex and have only one global minimum, or non-convex and have many local minima. Determining a model and its associated functional is the most crucial step in developing a new segmentation method. Indeed, the desired segmentation has to correspond to the global minimum, or to an "accessible" local-minimum of $E(\cdot, I)$, otherwise one cannot expect to retrieve such solution relying on E.

When presented with a new image I to segment, the model M is fit to the image via an *optimization* procedure, which consists on minimizing $E(\cdot, I)$ with respect to the model parameters. Naturally, this optimization step is also crucial in determining the quality of the obtained solution. Even when the global minimum of $E(\cdot, I)$ corresponds to the desired solution, it sometimes cannot be reached, either because the optimization procedure is mislead to an incorrect result – e.g. a local optimization can be "stuck" in non-desired local minimum – or because it would take too long to reach it – e.g. a NP-hard problem.

In this chapter, we will distinguish between two types of approaches. In a first part, we give an overview of vast family of *surface-based models* – i.e. models which rely on mathematical representations of the surface of the target objects. Among the most widely used models, *Deformable Models*, either explicit or implicit, are surface models which are progressively fit to the image in a *local* optimization process. An optimization process is said local when the solution is found by applying successive local changes from an initial state, a canonical example being the gradient descent. Among them, *landmark-based models* are surface models based on remarkable, anatomical or statistically consistent image points, which allows introducing statistical shape

priors for constraining the surface position. Finally, unlike Deformable Models, which mostly rely on non-optimal local-optimization methods, *graph-based surface models* can reach a global minimum of the cost functional thanks to efficient discrete optimization algorithms. This advantage is paid by a certain approximation of the solution due to the discretization of the solution space.

In a second part, we present *region-based models* – i.e. which parametrize every point of the image domain – devoid of intersection issues in the case of multi-object segmentation, but whose topology and shapes are more difficult to constrain. Among these, some use a typical approach based on continuous local optimization, while others rely on a graph formalism: *Graph-Cuts* and *Random Walks*. Graph-based models provide convenient and efficient frameworks for multi-object segmentation, for dealing with weak edges, and for accepting simple user interaction in the form of *seeds*. However there are still few examples of such models incorporating prior knowledge of shapes.

2.1 SURFACE-BASED MODELS

Although they may have very diverse mathematical formulation, or rely on very different types of information, surface-based models all have in common a parametrized model of the surface of the objects of interest. Given a particular state of the model, the image segmentation is determined by the location of the model surface: any point in the image domain is either on one side of the surface, or the other. If the surface is closed, then any point is either inside the volume delimited by this surface – thus assimilated to this object's class, or outside – thus assimilated to another class or to the background class.

The value of the energy functional of a surface model thus depends on the location of the surface with respect to the segmented image. What is measured by the functional can either be related to image boundaries – i.e "is the model's surface matching the edges of the image ?" – or to image regions – i.e. "do the properties of the regions delimited by the surface model match what I know of them *a priori* ?".

2.1.1 *Active Contours*

Active Contours is a general term which designates surface-based models whose functional minimization procedure is a suboptimal strategy consisting in evolving an initial surface (or contour in 2D) in the direction of the steepest descent of the functional. In general, the cost functional which is used is not convex and the adopted minimization approaches only reach local minima, highly dependent on the initialization state.

The first and most simple instances of active contours rely only on low-level image features, such as edges, region intensity, texture, as well as low-level shape features, such as contour length, object's surface (object's volume in 3D), contour smoothness, etc. However, a number of factors such as noise, occlusion, complexity of shapes,

etc. as well as sensibility to initial conditions often limit the performance of purely low-level methods. For these reasons, higher-level constraints on shapes were later introduced, called *shape priors*, based on previously seen similar objects.

Let us first consider a boundary object $\mathcal{C}$ embedded in the domain space Ω of the image I, such as a planar curve ($\mathcal{C} \subset \mathbb{R}^2$) or a surface ($\mathcal{C} \subset \mathbb{R}^3$) to be deformed from an initial position in order to minimize an energy functional $E(\mathcal{C})$. The energy $E(\cdot)$ is specific to the model, as it is determined by the kind of image information one wants to exploit; in the next sections, we present several formulations for $E(\cdot)$, depending on whether it uses edge information or region information.

The energy functional E can often be minimized using finite elements methods for the variational form, by making $\mathcal{C}$ depend on time:

$$\frac{\delta \mathcal{C}}{\delta t}(s,t) = -\nabla_{\mathcal{C}} E\left(\mathcal{C}(s,t), t\right). \tag{2.1.1}$$

This expression is the base evolution equation for most active contours models. Such formulation has the visual effect of making the the surface of the boundary object deform with time, hence the names "active contours" and "deformable models". The literature on active contours is extremely large and, for this reason, we only give a brief and incomplete survey of it. For a deeper review of active contour methods, we indicate to the reader several review articles and books: McInerney and Terzopoulos (1996); Montagnat et al. (2004); Cremers et al. (2006b); Paragios et al. (2005). Active contours can be driven either by the edges of the image or by the properties of the image regions – or by both simultaneously.

Active Contours and edge information

Active Contours for image segmentations based on edge information, also called *Snakes*, were introduced by Kass et al. (1988), along with the typical energy functional:

$$E(\mathcal{C}) = \int_0^1 \frac{\alpha}{2} |\mathcal{C}_s|^2 + \frac{\beta}{2} |\mathcal{C}_{ss}|^2 + E_{\text{ext}}(\mathcal{C})\, ds, \tag{2.1.2}$$

where $\mathcal{C}$ is a 2D parametric curve $\mathcal{C} : [0,1] \to \Omega$, $\mathcal{C}_s$ and $\mathcal{C}_{ss}$ denote the first and second derivative of $\mathcal{C}$, α and β are weighting parameters, respectively for tension ($\mathcal{C}_s$) and rigidity ($\mathcal{C}_s$). These first two terms can be viewed as *internal energy* terms, as they measure properties of the contour itself. E_{ext} is a function based on an *edge map* of the image I, such that it is minimal on image edges. This term is an *external energy* term, measuring the adequacy of the contour with the image. A canonical example for E_{ext} is the image squared gradient norm:

$$E_{\text{ext}}(x) = -|\nabla I(x)|^2. \tag{2.1.3}$$

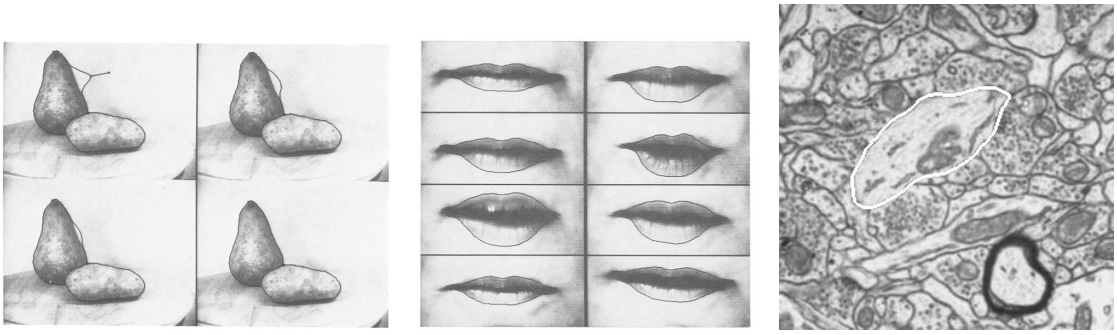

Figure 2.1.1 – Segmentation with snakes: (left) model evolution: the boundary model converges towards the edge of the pear (Kass et al., 1988); (middle) tracking a person's mouth with a snake in a video (ibid); (right) snake-segmented cell in an EM photo-micrograph (McInerney and Terzopoulos, 1996).

Deriving equation 2.1.2 by following the scheme of 2.1.1 gives:

$$\frac{\delta \mathcal{C}}{\delta t} = \alpha \mathcal{C}_{ss} - \beta \mathcal{C}_{ssss} - \nabla E_{\text{ext}}. \tag{2.1.4}$$

Such formulation can be extended to 3D straightforwardly (Terzopoulos et al., 1988). Many variations on Snakes were developed, e.g. to decrease the dependance on the initial conditions, the *balloon* model (Cohen, 1991) adds a force term to equation 2.1.4:

$$\frac{\delta \mathcal{C}}{\delta t} = \alpha \mathcal{C}_{ss} - \beta \mathcal{C}_{ssss} - \nabla E_{\text{ext}} + \nu \mathbf{n}, \tag{2.1.5}$$

where $\nu > 0$ sets the strength of the balloon force, and $\mathbf{n}$ is the outward normal to the contour. Such expression amounts to maximizing the area of $\mathcal{C}$ while keeping the boundaries smooth. Other examples include: adopting a multi-resolution framework using Fourier descriptors for parametrizing the contour (Leroy et al., 1996); replacing the finite difference numerical optimization approach by a more robust finite element method (Cohen and Cohen, 1993); using the gradient vector flow in the external energy term in order to extend the capture range of the initial state (Xu, 1998). The mathematical parametrization is also crucial to determine the extent of the solution space; for instance, initial deformable models were based on standard triangular or quadrangular mesh parametrization (Terzopoulos et al., 1988; Nastar and Ayache, 1993); however, this type of parametrization necessitates frequent reparametrization and makes topology changes – that is, changes in the object's Euler characteristic or in the number of objects – difficult. Types of parametrization allowing topology changes have been proposed, such as particle systems (Szeliski and Tonnesen, 1992; Lombardo et al., 1995) – which model the surface with elementary objects whose position is ruled by local forces, simulating the evolution of physical particles in a Newtonian

space. The popular Simplex meshes (Delingette, 1999) – which are topologically dual to triangular meshes – also allow some topology changes and benefit from convenient reparametrization schemes. More recently, Delaunay deformable models have been proposed (Pons and Boissonnat, 2007), which model the surface from a Delaunay "tetrahedralization" (that is, the extension of triangulation in 3D) of the entire space based on surface points. This model allows simple topology changes and is free from topology issues, such as undesired intersections of objects.

Active Contours and region information

The models presented in the previous section are considered edge-based, as they are driven by the proximity of edges according to the external energy E_{ext}. Such dependency make these models very sensitive to noise, and inadequate to segment objects which cannot be differentiated by detecting edges between them, e.g. in cases of fuzzy of blurred contours, or of highly textured objects.

However, other segmentation approaches are based on image regions rather than edges. The classical approach was proposed by Mumford and Shah (1989); Blake and Zisserman (1987), were the image I is approximated with a piecewise smooth function u, via minimizing the functional:

$$E(u, \mathcal{C}) = \frac{1}{2}\int_{\Omega} (I-u)^2\, dx + \lambda^2 \frac{1}{2}\int_{\Omega/\mathcal{C}} |\nabla u|^2\, dx + \nu\,|\mathcal{C}|\,, \tag{2.1.6}$$

with respect to both u and $\mathcal{C}$. Here, $\mathcal{C}$ is a finite set of C^1-arcs. Function u is designed to be a texture-less approximation of image I, with sharp variations only at the boundaries matched by $\mathcal{C}$ and smooth variations inside regions defined by $\mathcal{C}$. When minimizing this functional, the first term ensure the resemblance of u with I, the second term penalizes discontinuities of u except on the boundary $\mathcal{C}$ and the third term limitates the length of the boundary to segment only important objects. Minimizing this highly non convex functional is notoriously difficult and necessitates approximate methods and complex optimization procedures (Blake and Zisserman, 1987). However, a simpler version of the Mumford Shah functional 2.1.6 can be obtained in the limit $\lambda \to \infty$, yielding the *cartoon limit* (Mumford and Shah, 1989):

$$E(\{u_i\}, \mathcal{C}) = \frac{1}{2}\sum_i \int_{R_i} (I-u_i)^2\, dx + \nu\,|\mathcal{C}|\,, \tag{2.1.7}$$

where $\{u_i\}$ is a partition of I into constant regions $\{R_i\}$ determined by $\mathcal{C}$. This functional was modified by Zhu and Yuille (1996), introducing a Gaussian probabilistic model of the constant gray-level value within each regions:

$$E(\{\alpha_i\}, \mathcal{C}) = \frac{1}{2}\sum_i \int_{R_i} -\log P(I|\alpha_i)\, dx + \nu\,|\mathcal{C}|\,, \tag{2.1.8}$$

where $P(I|\alpha_i)$ is the posterior probability of obtaining the intensity values in I inside region R_i given distribution parameter α_i.

A hybrid model, called *diffusion snake,* was also proposed (Cremers et al., 2002):

$$E(u, C) = \frac{1}{2}\int_{\Omega}(I-u)^2\,dx + \lambda^2\frac{1}{2}\int_{\Omega/C}|\nabla u|^2\,dx + \nu\int|C_s|^2\,, \tag{2.1.9}$$

where the external energy is that of the Mumford Shah functional 2.1.6, while the internal elastic constraint on the boundary is that of the Snake 2.1.2. Such model can then be optimized via standard gradient descent methods.

2.1.2 *Surface models and shape priors*

The regularization factors in the approaches of the previous sections, rendered necessary due to noise and missing parts, all have the effect of minimizing the contour length (Bergtholdt et al., 2006). These purely geometric constraints can be viewed as shape priors, but they are based only on the intuition that natural shapes have smooth boundaries. Such hypothesis is hardly true in practical cases, and consistently yields excessive regularization, e.g. eroded contours and ignored small details. Shape priors based on previously-seen examples, by imposing the boundary to adopt realistic shapes, allow segmenting complex shapes despite noise and occlusions.

Single reference shape models

The most simple shape prior type in deformable models is enforced when the initial shape of the boundary corresponds to a template of the object to be segmented. The regularization terms are sometimes enough to ensure that the shape remains valid as the model deforms. This reference template boundary can either be manually positioned or automatically positioned onto the target image via rigid registration procedures. This reference shape can be a rough approximation (simple mathematical objects, quickly hand-drawn shapes), or a sample shape obtained from a previously segmented image.

An elastic shape model was introduced by Terzopoulos et al. (1987), proposing an internal energy term measuring the changes in the curvature and the length between nearby points – via the first and second fundamental forms – of the deformed body. Spring-mass analogies were also proposed (Vasilescu and Terzopoulos, 1992; Nastar and Ayache, 1993), in which the surface model – parametrized as a discrete mesh – has its shape constrained by a set of springs connecting the vertices together. The length of the springs at rest determines the equilibrium position(s) and thus the minimum cost shape configuration(s). The stiffness of the springs determines the elastic properties of the model and thus the range of shapes the model can take. Over the years several models simulating soft tissue deformation have been proposed for segmentation,

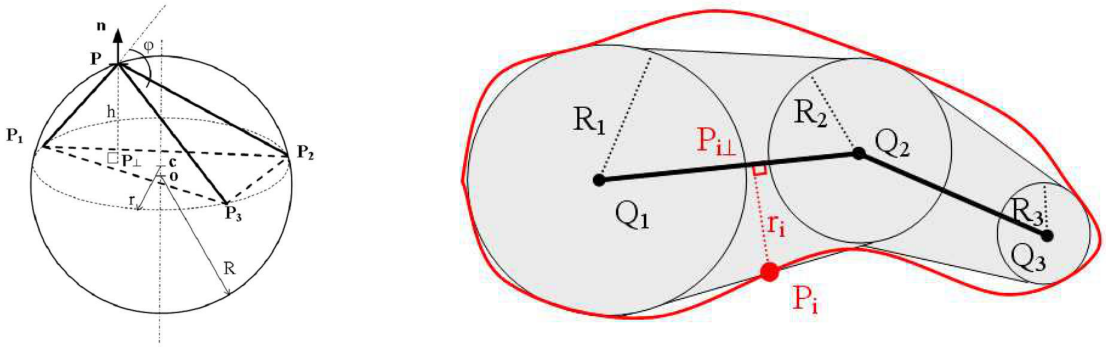

Figure 2.1.2 – (left) Local parametrization of simplex meshes (Delingette, 1999); (right) medial representation for constraining the model shape (Gilles and Magnenat-Thalmann, 2010).

bridging gaps between computer graphics, biomechanical simulation and computer vision.

MUSCLE SEGMENTATION USING SIMPLEX MESHES WITH MEDIAL REPRESENTATIONS Gilles and Magnenat-Thalmann (2010) presented a skeletal muscle segmentation method based on simplex meshes (Delingette, 1994). Considering a 3D surface model, simplex meshes are discrete meshes where each vertex has exactly 3 neighbors. Having a constant connectivity allows to simply parametrize the location of one vertex with respect to its neighbors, and thus parametrize deformation of the shape – translation, rotation, scaling – in a local manner. Indeed, the location of a pixel, denoted as $\mathbf{x}$ can be expressed as a linear combination of the locations of the three neighbors plus a local elevation term parallel to the local normal (see figure 2.1.2): $\mathbf{x} = \varepsilon_1 \mathbf{x}_1 + \varepsilon_2 \mathbf{x}_2 + (1 - \varepsilon_1 - \varepsilon_2)\, \mathbf{x}_3 + h\mathbf{n}$. As a result, many local measurements – including curvature and cell surface – can be computed efficiently and global energy terms enforcing local constraints come up naturally.

Here, the authors impose local smoothing via curvature averaging, which does not tend to reduce the surface like 1-order operators typically do. Prior knowledge is imposed by constraining the local scale changes on the elevation parameter with respect to a reference shape. Denoting the surface of the triangle formed by the three neighbors of a pixel as S, given the reference shape parameters $\left(\tilde{\varepsilon}_1, \tilde{\varepsilon}_2, \tilde{h}, \tilde{S}\right)$, the new location of the considered pixel is expressed as:

$$\mathbf{x} = \tilde{\varepsilon}_1 \mathbf{x}_1 + \tilde{\varepsilon}_2 \mathbf{x}_2 + (1 - \tilde{\varepsilon}_1 - \tilde{\varepsilon}_2)\, \mathbf{x}_3 + \tilde{h} \left(S/\tilde{S}\right)^{1/\beta} \mathbf{n}, \tag{2.1.10}$$

where $\beta \in [2, +\infty[$ is a parameter which sets the amount of allowed local deformation: with $\beta = 2$ this definition is similitude invariant; with $\beta = +\infty$ this definition is invariant through rigid transformations only. The model is attached

to the target image through either gradient norm maximization in the direction of the gradient at the location of the vertices, or maximization of similarities between the reference and the target images at the vertices location.

A medial representation – similar to the M-reps of Pizer et al. (2003) – is combined with the simplex parametrization to exploit the specific tubular shapes of the skeletal muscles. Medial vertices are added to the model, constrained to remain on the medial axis of the tubular objects. This is achieved by connecting the new vertices to the surface vertices through spring-like forces. This constrains the global structure to resemble its initial reference shape, thus acting as a global shape prior (see figure 2.1.2). This medial axis representation also allows efficient collision handling. The model is fit to the image through an iterative process of successive local evolutions. Such model appear to always yield a valid solution, sometimes at the price of an excessive regularization or lack of adaptability to the specifics of the target image.

MUSCLE SEGMENTATION USING DEFORMABLE MODELS AND SHAPE MATCHING A shape prior for muscle segmentation in 3D images was derived from a computer animation technique, called *shape matching*, used to efficiently approximate large soft-tissue elastic deformations (Gilles and Pai, 2008). This method was applied to muscle segmentation with some success. In this approach, discrete meshes are used to parametrize the moving surface. Let $\mathbf{x}^0$ be the vector containing the initial position of the control points of the parametric surface. Clustering is performed on $\mathbf{x}^0$ such that each cluster ζ_i contains at least a certain number of vertices (set by the user). During segmentation, the evolution of the active surface is performed according to the following iterative procedure:

1. Shift vertices according to the external force: $\tilde{\mathbf{x}}^t = \mathbf{f}_{\text{ext}} + \mathbf{x}^t$. The external "force" $\mathbf{f}_{\text{ext}}$ is computed as the maximal gradient search in the gradient direction.

2. Regularize vertex positions:

   a) Compute rigid registration for each clusters:
   $\mathbf{T}_i = \arg\min \sum_{j \in \zeta_i} \left\| \mathbf{T}_i \mathbf{x}_j^0 - \tilde{\mathbf{x}}_j^t \right\|^2$,

   b) Average target position for each vertex:
   $\mathbf{x}_i^{t+1} = \frac{1}{|\zeta_i|} \sum_{j \in \zeta_i} \mathbf{T}_j \mathbf{x}_j^0$.

Single reference prior models are convenient in that they require only one annotated example of the objects of interest. However, when segmenting a class of objects whose shape varies a lot, such approach becomes too constraining and does not allow the model to adopt valid shapes which are too different from the single reference.

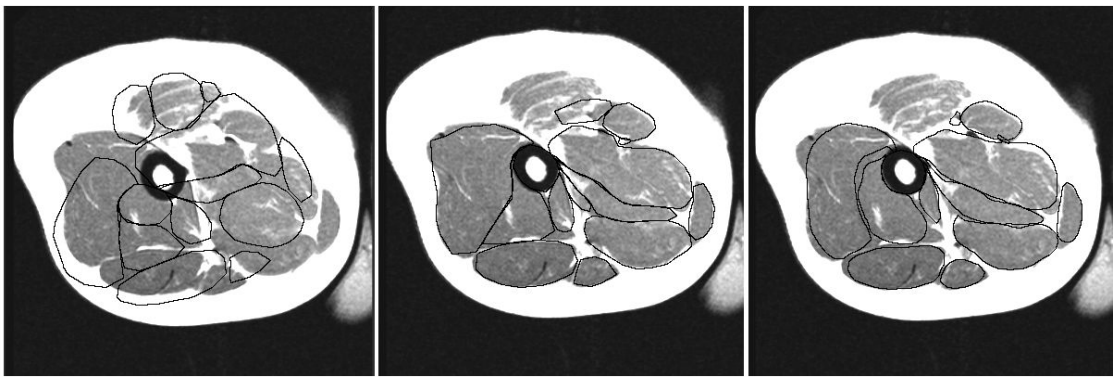

Figure 2.1.3 – Muscle segmentation using deformable models with a shape matching prior (Gilles and Pai, 2008): from the first iteration (left) to the last (right).

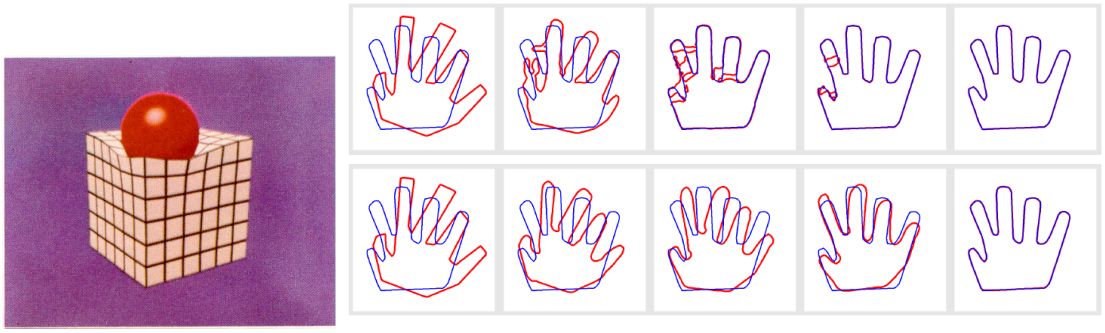

Figure 2.1.4 – Single reference shape priors. (left) An elastic cube model Terzopoulos et al. (1987). (right) The advantages of generalized gradients (bottom) over standard gradient descent (top) using the same deformable model: a realistic shape is kept all along the deformation process (Charpiat et al., 2007).

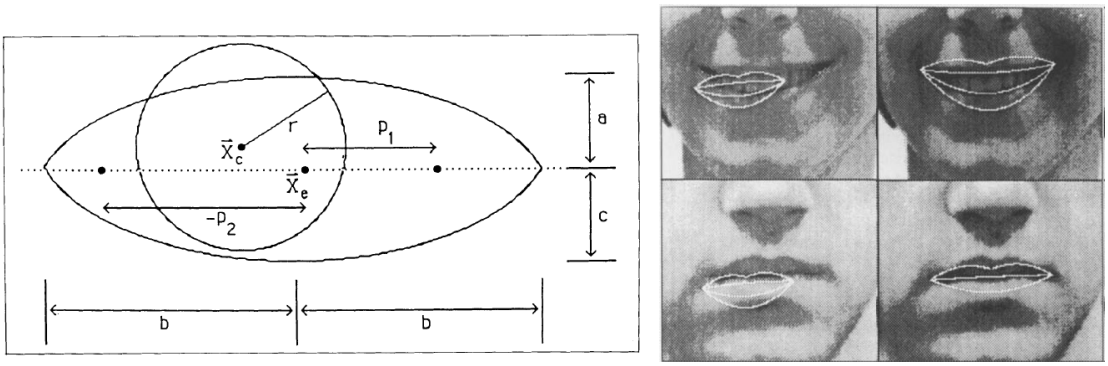

Figure 2.1.5 – Deformable templates: (top) a template eye made of a few simple mathematical shapes (Yuille, 1990); (bottom) segmentation of a person's mouth, using deformable mouth template (ibid).

In a different approach, a shape prior was derived from a generalization of the fundamental variational equation 2.1.1 (Charpiat et al., 2007). Such generalization is based on the observation that the canonical L^2 inner product is always implied when taking the gradient of the energy. The proposed method relies on replacing this canonical inner product by new inner products, designed to enforce spatially coherent motions with respect to the initial template shape. As a consequence, the trajectory of the boundary is optimized so the energy does not have to include regularization terms. This method can be used both with level sets (see section 2.1.4) and standard Active Contours.

Deformable templates

Deformable templates (Yuille, 1990) are combinations of simple geometric shapes requiring very few parameters – e.g. a center location and a radius for a circle. While their simplicity makes them robust and fast to optimize, these models are not suited to segment complex shapes and large deformations.

Superquadrics, hyperquadrics

Superquadrics (Terzopoulos and Metaxas, 1990; Bardinet et al., 1996) are geometric shapes defined by parametric equations, generalizing simpler objects such as ellipsoids. Like deformable templates, they require very few parameters, and have similar pros and cons: simple and fast to optimize, but the space of available shapes is too small to model complex shapes.

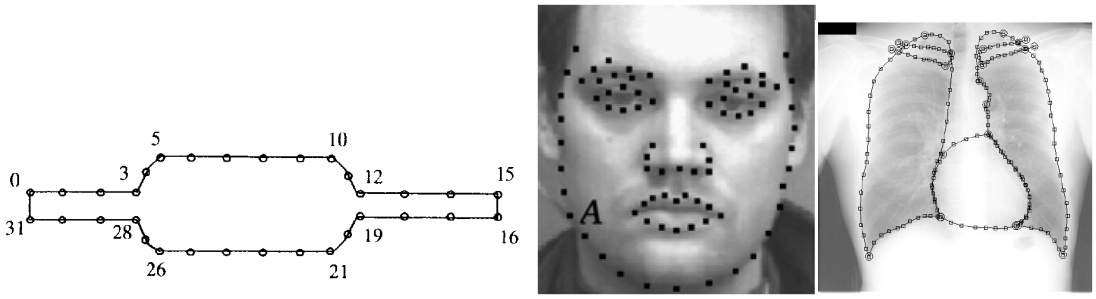

Figure 2.1.6 – Examples of landmarks: (left) 32 manually placed landmarks on the surface of a resistor Cootes et al. (2001); (middle) manually placed landmarks on a person's face (Romdhani et al., 1999); (right) manually and interpolated landmarks on the surface of lungs in an X-ray image (Ginneken et al., 2006).

Hyperquadrics (Cohen and Cohen, 1996; Hanson, 1988) also defined by parametric equations, extend superquadrics by allowing a wider range of shapes. A hyperquadric is defined by the points (x, y, z) satisfying the equation:

$$H(x, y, z) = \sum_{i}^{N} |a_i x + b_i y + c_i z + d_i|^{\gamma_i} = 1. \tag{2.1.11}$$

Unlike superquadrics, hyperquadrics are not restrained to symmetric shapes. Hybrid hyperquadrics (Cohen and Cohen, 1996) further extend superquadrics by allowing to create some local concavities in hyperquadrics. However, small local details remain inaccessible to these models and many authors have used them as coarse starting points to be refined in a second stage (e.g. using free-form deformations).

Landmark-based models

Statistical shape priors are built from sets of training annotated images, and aim at generating a subspace $\mathcal{S}$ of valid solutions capturing most of the shape variability while remaining as small as possible. Once such a subspace is generated, the model is constrained to evolve within it or to remain close to it, thus ensuring a valid shape is always kept. For contour models, such priors generally require to define a set of consistent boundary points, called *landmarks* (cf. figure 2.1.6).

The Active Shape Model (ASM) (Cootes et al., 1995) uses a Principal Component Analysis (PCA) performed on the position of the landmarks to generate a sub-space $\mathcal{S}$ of solution containing the main variation modes (cf. figure 2.1.7). Let $\{\mathbf{x}_k\}_{k=1\ldots K}$ be

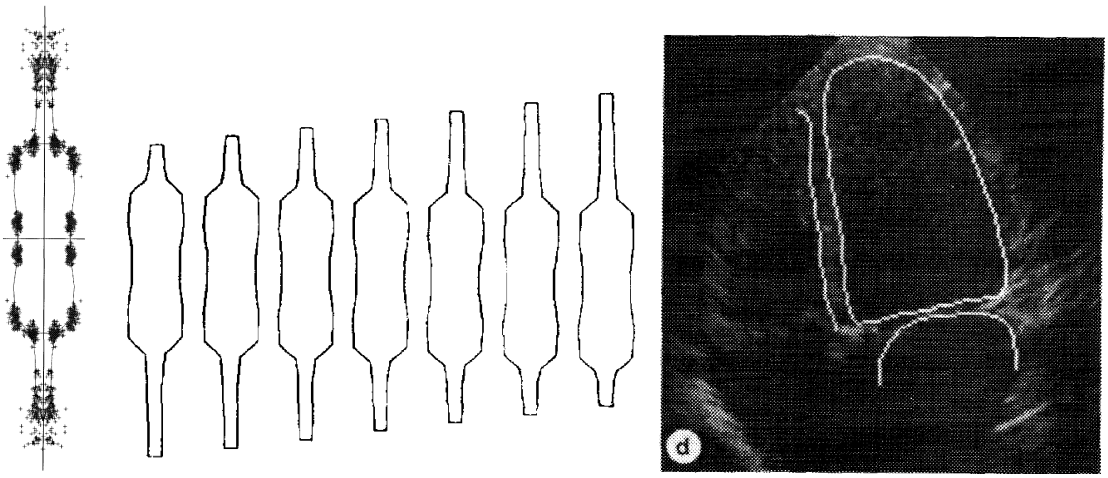

Figure 2.1.7 – Active Shape Model: (left): scatter plot of the landmarks of the training examples (Cootes et al., 1995); (middle) synthesized models using: $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{U}\mathbf{p}$ for different values of $\mathbf{p}\,[0]$ (ibid); (right) segmentation of a heart ventricle in an echo-cardiogram using ASM (ibid).

the set of vectors containing the landmark positions for the training data. The average models and its covariance are:

$$\bar{\mathbf{x}} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k, \tag{2.1.12}$$

$$\Sigma = \frac{1}{K}\sum_{k=1}^{K}\left(\mathbf{x}_k - \bar{\mathbf{x}}\right)\left(\mathbf{x}_k - \bar{\mathbf{x}}\right)^T. \tag{2.1.13}$$

Then, PCA on Σ gives:

$$\Sigma \approx \mathbf{U}^T \mathrm{diag}\left(\Lambda\right)\mathbf{U}, \tag{2.1.14}$$

where Λ is the vector of retained eigenvalues and $\mathbf{U}$ is the corresponding set of retained eigenvectors stored in columns. Then, any shape lying within the prior subspace of solutions as determined by the set of boundary landmarks writes:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{U}\mathbf{p}, \tag{2.1.15}$$

where $\mathbf{p}$ is the coordinate vector of $\mathbf{x}$ in $\mathcal{S}$. Such formulation assumes that the distribution of realistic shapes can be effectively approximated by a Gaussian distribution. During model evolution, which is similar to a gradient descent, the boundary is regularly projected into subspace $\mathcal{S}$ to ensure the model shape always remain realistic with respect to the training examples.

A similar strategy can be adopted, but using the full covariance matrix Σ instead of a subspace via a selection of the main variation modes (Cremers et al., 2002). Shape

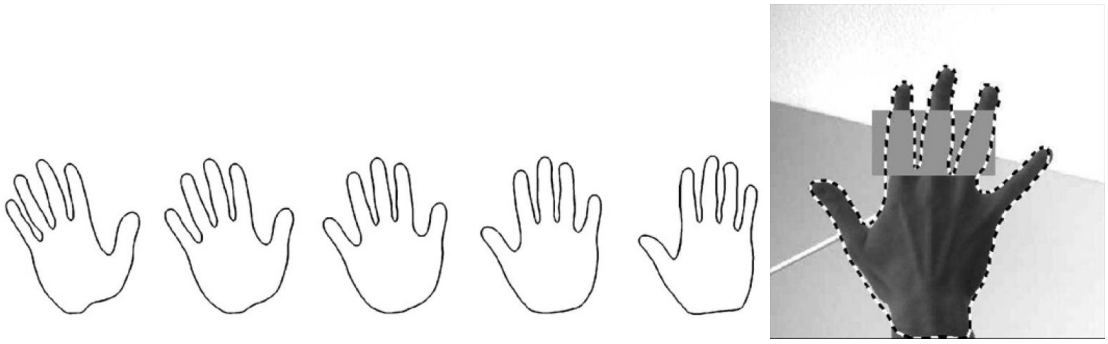

Figure 2.1.8 – (left) Invalid Gaussian hypothesis: variations of the first mode of a linear ASM of a hand; the most probable shape (center) is the least realistic (Cremers et al., 2003).
(right) A kernel-ASM trained on both right and left hand images allows to segment a right hand in a new image (ibid).

correctness is enforced by minimizing an energy term in the form of a Mahalanobis distance, in addition to the external energy term:

$$E_{\text{shape}}(\mathbf{x}) = (\mathbf{x} - \bar{\mathbf{x}})^T \Sigma_{\perp}^{-1} (\mathbf{x} - \bar{\mathbf{x}}), \tag{2.1.16}$$

where $\Sigma_{\perp}$ is the regularized covariance matrix, authorizing inversion. Like ASM, such model implies that the distribution of shapes be Gaussian, and penalizes – but does not forbid – more strongly shapes having a low probability.

However, for many classes of shapes, the Gaussian hypothesis leading to linear shape statistics is not valid (cf. figure 2.1.8). In order to circumvent this limitation, a typical solution is found in using constraints in a *feature space* via kernel methods. Let $\phi : \Omega \to \mathcal{F}$ be a non-linear mapping from the input space to a higher-dimensional feature space, potentially unknown. Using a Kernel Principal Component Analysis (KPCA), a *Kernel Active Shape Model* was proposed (Romdhani et al., 1999; Twining and Taylor, 2001). As in the linear ASM, the optimization procedure of Romdhani et al. (1999) consists in regularly projecting the mapped boundary back into the feature space. Twining and Taylor (2001) enforced the shape prior by minimizing an energy based on a "proximity to data" measure in the feature space. Using a formulation similar to that of equation 2.1.16, a non-linear shape prior was proposed (Cremers et al., 2006a) such that the empirical mean and covariance are computed on the mapped training vectors.

MUSCLE SEGMENTATION USING A HIERARCHICAL STATISTICAL SHAPE MODEL A hierarchical prior model using Diffusion Wavelets was proposed to segment organs (Essafi et al., 2009) – including one calf muscle – in MRI. This model builds on the formulation of the ASM (cf. equation 2.1.15), using a differ-

ent basis for the subspace of valid solutions. One of the main drawbacks of ASMs, is that they often require a large number of training data in order to obtain relevant decomposition modes. Indeed, some non-correlated shape features – such as global and local shape configurations – are often modeled by the same deformation modes. Thus, desired shape behaviors are often mixed with unwanted shape behaviors when optimizing the shape parameters for segmenting a new image. The hierarchical approach allows to uncorrelate small and large scale shape behaviors. Moreover, the presented method also uncorrelates long-range shape behaviors, thus ensuring that deformation mode are spatially localized.

We give a brief summary of this method. First, a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is built on the set of landmarks: $\mathcal{V}$ is the set of nodes and each landmark corresponds to a node in $\mathcal{V}$; $\mathcal{E}$ is the set of edges, whose weights are determined through a statistical analysis of the mutual relations between the landmarks in the training set $\{\mathbf{x}_k\}_{k=1...K}$ (cf. Shape Maps (Langs et al., 2008)). As a result, landmarks with independent behaviors will be connected by edges with a small weight, whereas nodes with strongly related behaviors – such as neighboring points – will be connected by large weight edges.

Second, a Diffusion Wavelet decomposition of $\mathcal{G}$ is performed. This process involves computing the diffusion operator $\mathbf{T}$ of graph $\mathcal{G}$, which is the symmetric normalized Laplace-Beltrami operator, and computing and compressing the dyadic powers of $\mathbf{T}$. The output of this decomposition is a hierarchical orthogonal basis $\{\Gamma_i\}$ for the graph, whose vectors correspond to different graph scales; considering the vector of landmark positions when decomposed on the new basis:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{\Gamma}\mathbf{p}, \tag{2.1.17}$$

global deformations – i.e. global relations between all the nodes – are controlled by some of the coefficients in $\mathbf{p}$, while local interactions – i.e. local interactions between close-by nodes – are controlled by some other coefficients in $\mathbf{p}$. Projecting all the training examples onto this new basis, a PCA is performed *at each scale* of the decomposition. Finally, during the segmentation process, the landmarks are positioned on the target image in an iterative manner: 1) the position of the landmarks is updated according to a local appearance model; 2) they are projected into the hierarchical subspace defined previously.

2.1.3 *Implicit deformable models: level sets*

Level set methods, also called *Geometric Deformable Models*, are a special class of Deformable Models. They were designed to solve certain issues of standard Active Contours (also called *Parametric Deformable Models* or *Snakes*): the explicit parametrization of Snakes necessitates frequent re-parametrization in order to preserve accuracy; topo-

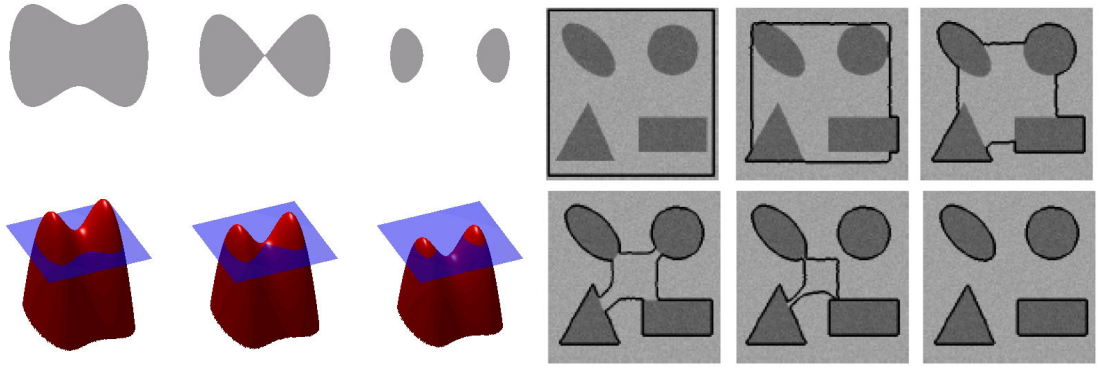

Figure 2.1.9 – Level set segmentation: (left) illustration of the level set method (from Wikipedia); (right) topological changed during segmentation of synthetic images (Paragios, 2005).

logical changes in snakes are difficult to achieve; the output depends heavily on the contour's initialization (position and shape); region features (color, texture) are difficult to take into account.

The level set method, proposed by Dervieux and Thomasset (1980); Osher and Sethian (1988) in the field of fluid dynamics, uses an implicit representation of the evolving boundaries $\mathcal{C}$ as the location of the zero level line of an implicit function $\phi : \Omega \rightarrow \mathbb{R}$ (cf. figure 2.1.9):

$$\mathcal{C} = \{x \in \Omega | \phi(x) = 0\}. \tag{2.1.18}$$

Such implicit representation allows *natural topological changes* of the implied boundary (cf. figure 2.1.9). This representation is then combined with the general energy functional for Active Contour models using parametric curves $\mathcal{C} : [0,1] \rightarrow \Omega$ (Paragios, 2005; Cremers et al., 2006b) (cf. section 2.1.1):

$$E(\mathcal{C}) = \int_0^1 \alpha E_{int}(\mathcal{C}(p)) + \beta E_{img}(\mathcal{I}(\mathcal{C}(p))) + \gamma E_{ext}(\mathcal{C}(p))\, dp, \tag{2.1.19}$$

where $\mathcal{I}$ is the image, E_{int} imposes curve smoothness (regularization term), E_{img} pushes the curve towards boundaries (data term), E_{ext} encodes user knowledge (interaction or prior term), and α, β, γ are balancing parameters. Such cost functional can be minimized within a finite difference approach on the variational form:

$$\frac{\partial \mathcal{C}}{\partial t} = -\frac{\partial E(\mathcal{C})}{\partial \mathcal{C}} = \mathbf{F} \cdot \mathbf{n}, \tag{2.1.20}$$

where F is a speed function and n is the contour unitary normal vector. Incorporating the level set representation to the previous formula yields:

$$\frac{\partial}{\partial t}\phi = -\left|\nabla\phi\right|\mathbf{F}. \tag{2.1.21}$$

Such a flow specifies the evolution of the implicit function ϕ at the location of the boundaries. Among the first applications of the level set formalism, a notable contribution is the *Geodesic Active Contours* model (Caselles et al., 1997; Kichenassamy et al., 1995), with evolution equation:

$$\frac{\partial}{\partial t}\phi = \left|\nabla\phi\right| \operatorname{div}\left(g\left(\mathcal{I}\right)\frac{\nabla\phi}{\left|\nabla\phi\right|}\right), \tag{2.1.22}$$

where g is an edge function, increasing with the strength of the edges in $\mathcal{I}$. Such equation models an inward (respectively outward) flow which shrinks (respectively expands) the initial curve towards the detected edges. For implementation, ϕ is generally considered as a signed distance function to the evolving curve:

$$\phi\left(x\right) = \begin{cases} 0 & , \ x \in \mathcal{C}, \\ D\left(x,\mathcal{C}\right) & , \ x \in \mathcal{C}_{in}, \\ -D\left(x,\mathcal{C}\right) & , \ x \in \mathcal{C}_{out}, \end{cases} \tag{2.1.23}$$

where $D(x,\mathcal{C})$ is the Euclidean distance from x to $\mathcal{C}$, $\mathcal{C}_{in} \subset \Omega$ is the region inside $\mathcal{C}$ and $\mathcal{C}_{out} \subset \Omega$ is the region outside $\mathcal{C}$.

Having to impose the direction of the flow, as well as relying only on edge detection make such formulation very dependent on the initialization and non-robust to undesired local extrema. To overcome these problems, a global regional term was introduced. For a two-phase segmentation, such model has an energy of the form (Chan and Vese, 2001; Cremers et al., 2006b):

$$\begin{aligned} E\left(\phi,\theta_1,\theta_2\right) \quad = \quad & \int_{\Omega} -H\left(\phi\right)\log p\left(\mathcal{I}|\theta_1\right) \\ & -\left(1-H\left(\phi\right)\right)\log p\left(\mathcal{I}|\theta_2\right) + w\left|\nabla H\phi\right| dx, \end{aligned} \tag{2.1.24}$$

where $p\left(\mathcal{I}|\theta_i\right)$ is the posterior probability to observe $\mathcal{I}$ given region model parameter θ_i, H is the Heaviside step function such that: $H\left(x\right) = 1, \ x \in \mathcal{C}_{in}$, and $H\left(x\right) = 0$ otherwise – in practice, H is a smooth differentiable approximation of the Heaviside distribution. Region parameters θ_i can either be set, as a prior knowledge of statistical properties of the objects region (Paragios and Deriche, 1998), or estimated in an alternate optimization method, assuming intensity piece-wise constant objects (Chan and Vese, 2001). Many variations on this formulation were developed, for multi-phase seg-

mentation (Zhao et al., 1996; Paragios and Deriche, 2004; Yezzi et al., 1999), for using non-parametric approximation of density functions (Kim et al., 2002; Rousson, 2002), instead of Gaussian approximation functions.

2.1.4 *Level sets and shape priors*

Incorporating shape prior knowledge within the level set framework has been done in multiple ways. Some methods (Chen et al., 2002; Rousson and Paragios, 2002) use the statistical average curve $\mathcal{C}_{\mathcal{M}}$, and associated signed Euclidean distance function $\phi_{\mathcal{M}}$ to guide the deformation of the model. For example, when added to a level set energy functional, minimizing the term:

$$E_{\text{shape}}(\phi, \mathcal{A}, s) = \int_{\Omega} \delta(\phi) \left(s\phi(x) - \phi_{\mathcal{M}}(\mathcal{A}(x))\right)^2 dx \tag{2.1.25}$$

imposes the implicit function ϕ to be similar to $\phi_{\mathcal{M}}$, where δ is a smooth approximation of the Dirac distribution, $\mathcal{A}$ is a linear or non-linear transform to be estimated, s is the scale factor between ϕ and $\phi_{\mathcal{M}}$ to be estimated.

As seen previously with explicit models, single reference models can only recover small deviations from the prior shape. In order to build a model able to recover important shape variations, methods using a PCA on a set of distance functions were developed (Leventon et al., 2000; Tsai et al., 2001), similar in principle to the Active Shape Model (cf. section 2.1.2):

$$\phi_{\lambda} = \phi_{\mathcal{M}} + \sum_{j} \lambda_j \mathbf{U}_j, \tag{2.1.26}$$

where λ_j is the parameter weighting the eigenvectors $\mathbf{U}_j$. Similar to 2.1.25, an energy term enforcing a shape prior can be formulated:

$$E_{\text{shape}}(\phi, \mathcal{A}, s, \lambda) = \int_{\Omega} \delta(\phi) \left(s\phi - \phi_{\lambda}(\mathcal{A})\right)^2 dx. \tag{2.1.27}$$

Such an approach assumes that the distribution of training shapes is Gaussian, which is often not true in practice, and thus necessitates a large number of eigenvectors in order to capture enough details of the shape. Approaches for modeling non-Gaussian shapes distributions were considered, using nonparametric density estimators (Cremers et al., 2006a; Rousson and Cremers, 2005), e.g. yielding prior energy terms of the form (Rousson and Cremers, 2005):

$$E_{\text{shape}}(\phi, \boldsymbol{\lambda}) = -\log \frac{1}{N} \sum_{i=1}^{N} K\left(\frac{\boldsymbol{\lambda} - \boldsymbol{\lambda}_i}{\sigma}\right), \tag{2.1.28}$$

where $\boldsymbol{\lambda}_i = (\lambda_1, \ldots, \lambda_n)$ is the shape vector (e.g. projection into a PCA space) for training shape i, $\boldsymbol{\lambda}$ is the shape vector of the model, K is a kernel function (e.g. Gaussian kernel $K(x) \propto \exp -x^2$) and σ is a scaling parameter. Such approach ensures the model remains in the vicinity of the training shapes.

2.1.5 *Graph-based surface models*

Unlike most surface-based models, which are usually optimized with continuous local methods, these models are fit to the image via discrete graph-based optimization procedures. The graph frameworks has the advantage of allowing to enforce similarity invariant statistical shape priors while searching for a globally optimal solution.

SURFACE MODELS AND PAIR-WISE DISCRETE OPTIMIZATION A landmark-based surface model was proposed by Besbes et al. (2009); Besbes (2010) for a 2-class segmentation methods with a shape prior based on learning joint probabilities between pairs of control points. Let us call x_i the spatial location of the i-th control points, $\mathcal{X} = \{x_1, \ldots, x_N\}$ the set of control points. Given an annotated training set of sample images, the joint probability for the relative location of two control points is learned based on the normalized distance between pairs of points:

$$d_{ij} = \frac{\|x_i - x_j\|}{d}, \tag{2.1.29}$$

where d is the estimated scale of the object. Since prior information relies only on a normalized distance between the control points, it is rotation, translation and scale invariant – although scale-invariance depends on a separate scale estimation. From these distance, a joint probability distribution is estimated: $p_{ij} = p(d_{ij})$ given a statistical model (e.g. a Gaussian model). A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is then built over the control points. The following functional is defined:

$$E(\mathcal{X}) = \sum_{v_i \in \mathcal{V}} V_i(x_i) + \sum_{e_{ij} \in \mathcal{E}} V_{ij}(x_i, x_j), \tag{2.1.30}$$

where: $V_i(\cdot)$ is the data inference term for each control point. This measures the adequacy of the segmentation with the pixels intensity; $V_{ij}(\cdot, \cdot)$ is the shape prior term, measuring the adequacy of the surface's shape with the prior knowledge:

$$V_{ij}(x_i, x_j) = -\log p_{ij}. \tag{2.1.31}$$

The most appropriate configuration for the set $\mathcal{X}$ can be obtained by minimizing the discrete energy E: each control point is limited to a finite number of displacements. As formulated, this problem is NP-hard and can only be approximately solved, e.g. via an efficient primal-dual algorithm (Komodakis et al., 2007). Due to the limited set

of locations for each control point, this optimization method remains non-global, but nonetheless covers a greater range of solutions than variational methods.

MUSCLE SEGMENTATION USING HIGHER-ORDER CLIQUES In the previous method, a shape prior is enforced via learning distribution models for pairs of control points. While such modeling makes a rotation and translation invariant model, it is only approximately scale invariant due to the normalization of distances by an estimation scale of the model. Moreover, the iterative approach for minimizing the functional imposes an initial guess for the control point location, e.g via user interaction.

A surface model using prior distributions over higher-order cliques – as triplets – was proposed by Wang et al. (2010). This allows to define an exact scale-invariant model and allows global search for control point locations due to the more constraining effect of imposing triplet configurations. Let us denote the spatial distribution for triplet (i, j, k) as p_{ijk} , which is learned for all triplets of points, using a mixture of Gaussians statistical model.

Unlike the previously presented method, control point possible locations are searched in a first stage, within the entire image. Such approach is achieved by capturing the local image appearance around the control points in the training set images. A series of Gabor filters are computed over patches surrounding the landmarks and fed to a Randomized Forest Classifier. Then the trained classifier is run over the target image to determine a set of best matches for each landmark and a score for each match. The k-th match for landmark i represents the assignment label l_i^k for node v_i. Defining $x_i(l_i)$ as the location of point i for label l_i, the following functional is defined:

$$E(\mathcal{L}) = \sum_{v_i \in \mathcal{V}} V_i(x_i(l_i)) + \sum_{e_{ij} \in \mathcal{E}} V_{ijk}\left(x_i(l_i), x_j\left(l_j\right), x_k\left(l_k\right)\right), \tag{2.1.32}$$

where: $V_i(\cdot)$ is the data inference term for each control point. This is defined as a decreasing function of the classification score for the corresponding point match; $V_{ij}(\cdot,\cdot,\cdot)$ is the shape prior term, measuring the adequacy of the surface's shape with the prior knowledge:

$$V_{ijk}\left(x_i(l_i), x_j\left(l_j\right), x_k\left(l_k\right)\right) = -\log p_{ijk}\left(x_i(l_i), x_j\left(l_j\right), x_k\left(l_k\right)\right). \tag{2.1.33}$$

This functional contains high-order cliques and can be minimized using a novel Dual-decomposition method (Komodakis, 2011). Since the possible location for each landmark are searched for globally, no iterative approach is required and the model does not necessitate an initial guess.

2.2 REGION-BASED METHODS

Apart from methods relying on surface representations, there also exists segmentation methods using regions models: each point in the image domain is parametrized by an assignment value, which can be hard – a *label* – or soft – e.g. a *probabilistic label assignment*. In general, these methods achieve good accuracy at segmenting the boundaries of the objects, but global shapes and topology are much more difficult to constrain. This is the opposite with surface-based methods which provide natural means for enforcing shape constraints and topology but often fail to segment the boundaries accurately. In the case of multi-object segmentation, surface-based methods necessitate complex computer animation algorithm to deal with interacting objects – e.g. to solve intersection issues. Region-based methods offer a simpler framework in that matter, since one point in the image domain can only belong deterministically to one label. Moreover, fuzzy or probabilistic assignment to different labels is also simpler with such methods.

2.2.1 *Continuous methods*

A continuous segmentation model based on probabilistic atlases was introduced by Cremers et al. (2008). This approach was motivated by the well-known limitations of level sets with shape prior methods: a) the cost functionals are generally non-convex, leading to non-optimal solutions and b) it was noticed that linear combinations of signed distance functions – used in building statistical shape priors – do not yield valid shapes and thus poor shape priors.

This approach is based on a probabilistic assignment model. Let us denote $q : \Omega \rightarrow [0, 1]$ such that $q(x)$ is the probability that x belongs of the target object. Assuming we own a set of annotated training images, denoted $\mathcal{T} = \{q_k\}_k$, a PCA is performed on $\mathcal{T}$. This operation involves computing the average probabilistic segmentation:

$$\mu = \frac{1}{|\mathcal{T}|} \sum_k q_k, \tag{2.2.1}$$

and the covariance matrix of the centered samples:

$$\Sigma = \frac{1}{|\mathcal{T}|} \sum_k (q_k - \mu)(q_k - \mu)^T. \tag{2.2.2}$$

Then a segmentation in the subspace spanned by the first n eigen-modes is:

$$q_\gamma = \mu + \Gamma\gamma, \tag{2.2.3}$$

where $\Gamma \in \mathbb{R}^{N \times n}$ denotes the matrix whose columns are the n retained eigenvectors. With these definitions, the following general cost functional is proposed:

$$E(q_\gamma) = E_i(q_\gamma) + \alpha E_s(q_\gamma). \tag{2.2.4}$$

where:

$$\begin{aligned} E_i(q) &= \int f(x)\, q(x)\, dx \\ &\quad + \int g(x)\,(1 - q(x))\, dx + \int h(x)\,|\nabla q(x)|\, dx, \end{aligned} \tag{2.2.5}$$

where $f(x)$ and $g(x)$ are functions binding the model to the image, and $h(x)$ a function constraining the length of the boundary. Such functional is a generalized version of the typical two-phase level set cost functional 2.1.24 (Chan and Vese, 2001). However, unlike the level set approaches, this functional is convex in q (whose domain is also convex). The shape prior functional is:

$$E_s(q_\gamma) = \gamma^T \Sigma^{-1} \gamma, \tag{2.2.6}$$

which is convex in γ. Such approach has the advantage that the shapes spanned by the PCA are always valid, unlike the level set case. It should be noted that this approach assumes a Gaussian model for the probabilistic assignment variables, which is a rather rough approximation and, as a consequence, the PCA does not generate probability vectors.

MUSCLE SEGMENTATION USING A CONTINUOUS REGION MODEL The previous method was extended to multi-label segmentation by Andrews et al. (2011b), and applied to skeletal muscle segmentation (Andrews et al., 2011a). Before performing the PCA on the training samples, an Isometric Log-Ratio (ILR) transform is applied to the assignment vectors. The reason for using this transform is that multi-label segmentation requires to have probabilities at all times, which the previous method does not achieve. Here, the PCA is performed in the ILR space and its output is projected back into the initial probability space. Denoting $\eta_\gamma = \mu + \Gamma\gamma$ a segmentation in the subspace of valid solution spanned by the PCA in the ILR space, the following functional is proposed:

$$E(\eta_\gamma) = d(\eta_{BG}, \eta_\gamma)^2 + \int (1 - h(x))\,|\nabla \eta_\gamma|^2 + \gamma^T \Sigma^{-1} \gamma, \tag{2.2.7}$$

where $d(\eta_{BG}, \cdot)$ is an intensity prior functional for separating muscle voxels from background voxels, and $h(x)$ is and edge-map of the target image such that the

energy is minimal when the boundaries of the model match the edges in the image.

2.2.2 *Graph-based methods*

In the following we review region-based methods relying on a graph formalism. Digital images are intrinsically discretized: in space into pixels and in time into frames (videos) – amplitude quantization is generally too fine to be considered. For this reason, pixels/voxels decomposition provides a convenient solution domain and are often used as such. However, other sub-domains have been used by further reducing the search domain into sets of pixels with the same assignment value (super-pixels) . These graph-based methods are intrinsically "bottom/up", since they rely on pixel-scale features to provide a segmentation of larger scale objects. More importantly, graph-based methods offer means of obtaining globally optimal segmentations, which is not the case for most surface-models, whose results depend heavily on the initial conditions.

NOTATIONS Let us consider an image I with N pixels. A segmentation on I is formulated as a labeling problem of an weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges. The i-th node v_i corresponds to the i-th pixel of image I. An edge spanning two vertices v_i and v_j is denoted e_{ij}, and its weight w_{ij}. The set of edges is composed only of pairs of adjacent pixels – e.g 4-connected neighborhood structure in 2D, and a 6-connected structure in 3D.

Given a set of labels $\mathcal{S}$ (e.g. the indices of the muscles), segmenting I consists of assigning a label $s \in \mathcal{S}$ to each node $p \in \mathcal{V}$. In the binary case, $\mathcal{S} = \{0, 1\}$. We denote $x_i \in \mathcal{S}$ the assignment label of pixel i.

2.2.3 *Segmentation with Graph Cuts*

The *Graph Cuts* method was originally developed for binary clustering of images (Wu and Leahy, 1993; Shi and Malik, 2000b), and for interactive segmentation (Boykov and Jolly, 2001) where a user provides annotated pixels, the *seeds*, as an initialization for the algorithm. In the following, we briefly explain the principles of the graph cut theory for the binary case – i.e. a foreground/background segmentation.

Two *terminal nodes*, the source S and the sink T are added to the set of nodes $\mathcal{V}$. A *cut* on graph $\mathcal{G}$ is a separation of the nodes into two subsets, one $\mathcal{V}_S$ containing the source, and the other $\mathcal{V}_T$ the sink (cf. figure 2.2.1). We call $C \subset \mathcal{E}$ the subset of edges

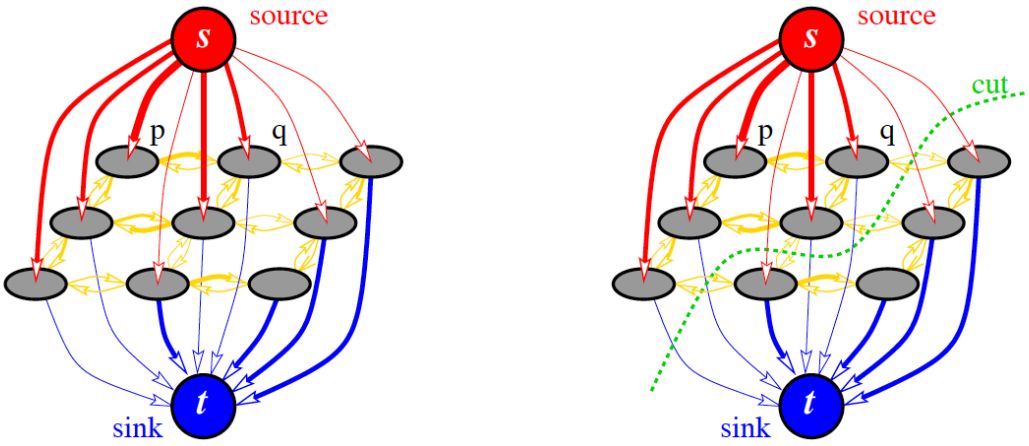

Figure 2.2.1 – Graph-cut: (left) a directed graph $\mathcal{G}$; (right) a cut on $\mathcal{G}$ (Boykov and Kolmogorov, 2004).

crossing the separation – i.e. with one vertex in $\mathcal{V}_S$ and the other in $\mathcal{V}_T$. The cost $|C|$ of this cut is the sum of the weight of the edges in C:

$$|C| = \sum_{(i,j)\in C} w_{i,j}. \tag{2.2.8}$$

The Ford-Fulkerson *max-flow/min-cut* algorithm can be used to find the minimum cut in polynomial time.

Segmenting image I is a (binary) labeling problem which amounts to defining a cut on graph $\mathcal{G}$, where each node in $\mathcal{V}_S$ are assigned to one label (e.g. the background) and each node in $\mathcal{V}_T$ are assigned to the other label (e.g. the foreground). An optimal segmentation is one which verifies certain criteria: e.g. intensity homogeneity ("I know this object has the same color everywhere"), intensity prior ("I know this object is dark"). These simple criteria can be expressed at the pixel scale in terms of edge weights:

- homogeneity of intensity between two pixels can be measured using a Gaussian weighting of intensity differences:

$$\forall\,(i,j) \in \mathcal{E}\setminus\{S,T\}\,,\; w_{ij} = \exp\left(-\beta\left(I_i - I_j\right)^2\right). \tag{2.2.9}$$

  where β is a contrast parameter;

- prior knowledge over intensity of one pixel can be measured using the negative log-likelihood:

$$\forall v_i \in \mathcal{V},\quad \begin{aligned} w_{iS} &= -\log \Pr\left(I_i|T\right), \\ w_{iT} &= -\log \Pr\left(I_i|S\right), \end{aligned} \tag{2.2.10}$$

where $\Pr(I_i|T)$ (respectively $\Pr(I_i|S)$) is the intensity distribution for the object attached to the sink (respectively the source). Note that if $\Pr(I_i|T) \approx 0$, then $w_{iS} \gg 0$, which implies that cutting the edge e_{iS} is very costly.

As a consequence, an optimal segmentation according to pair-wise criteria can be obtained by finding the *minimum cut* $\hat{C}$, which is a cut of $\mathcal{G}$ which has a minimal cost. Indeed, minimizing the cost of the cut implies cutting the edges with the smallest weight on average, which is equivalent of satisfying the selected criteria as well as possible. An optimal segmentation is obtained via minimizing the following functional with respect to the label assignments:

$$E(x) = \sum_{(i,j)\in\mathcal{E}} w_{ij} \left| x_i - x_j, \right| , \tag{2.2.11}$$

where $x = \{x_S, x_T\} \cup \{x_i\}_{i=1\ldots N}$ and where x_i is a binary assignment variable:

$$x_i = \begin{cases} 1 & \text{if } v_i \in \mathcal{V}_S, \\ 0 & \text{if } v_i \in \mathcal{V}_T. \end{cases} \tag{2.2.12}$$

Under this formulation, the assignment value for terminal nodes is of course already known: $x_S = 1$ and $x_T = 0$. Additional information can be provided in the form of seeds, i.e. used-assigned pixels. Let us call $\mathcal{V}_M$ the set of marked nodes, then finding the optimal segmentation is achieved by minimizing 2.2.11 with respect to the assignments for un-marked nodes $v \in \mathcal{V} \backslash \mathcal{V}_M$.

In the general case, $w_{ij}(x_i, x_j)$ can depend on the labels x_i and x_j. It was shown that Graph Cuts achieve a globally optimal solution providing that $w_{ij}(\cdot,\cdot)$ is a sub-modular function (Kolmogorov and Zabih, 2004), i.e. that it verifies:

$$w_{ij}(0,0) + w_{ij}(1,1) \leq w_{ij}(1,0) + w_{ij}(1,0). \tag{2.2.13}$$

When one wishes to segment several objects at once, the multi-label case can be formulated as a multi-way cut problem (Boykov et al., 1999). However, such formulation was proven to be NP-hard and only good approximate solutions can be found (Boykov et al., 1999; Komodakis et al., 2007).

A different approach with no required intensity prior (Wu and Leahy, 1993; Shi and Malik, 2000b) can be formulated by setting: $\forall i,\ w_{iS} = w_{iT} =$ constant. Moreover, seeds are no longer required, so the resulting segmentation depends only on the intensity difference between pixels. Such method is called *perceptual grouping*. However, the output segmentation tends to be biased towards very small segments, because small segments have less edges and induce a lower cut cost (Wu and Leahy, 1993). This problem can be solved with normalized cuts (Shi and Malik, 2000b) (cf. figure 2.2.2),

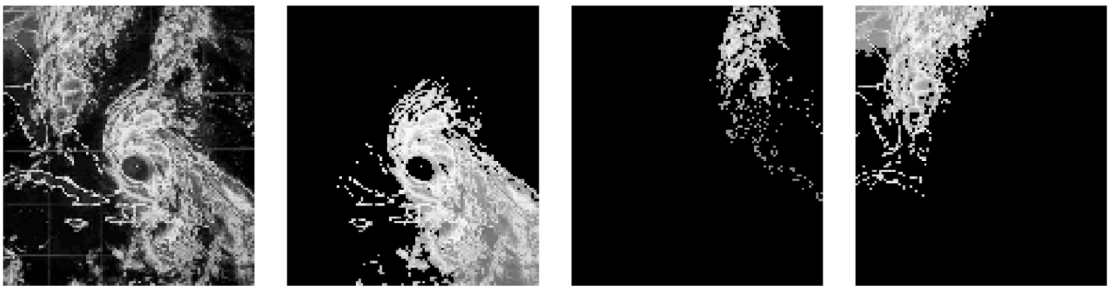

Figure 2.2.2 – Automatic image partition using normalized cuts (Shi and Malik, 2000b).

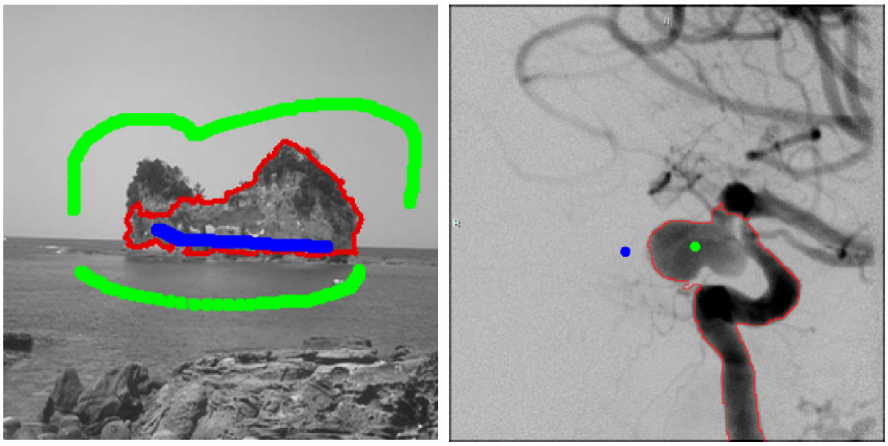

Figure 2.2.3 – Two examples of graph-cut segmentation (red curve) using user-provided seeds (green and blue marks) (Sinop and Grady, 2007).

where a different functional is minimized, consisting in a normalized cut cost by the cost of all edges within the parts:

$$E_{\mathrm{nC}}(x) = \left(\sum_{(i,j)\in\mathcal{E}} w_{ij} \left| x_i - x_j \right| \right) \left(\frac{1}{\sum_{(i,j)\in\mathcal{E}} w_{ij} x_i} + \frac{1}{\sum_{(i,j)\in\mathcal{E}} w_{ij} (1 - x_i)} \right). \tag{2.2.14}$$

However, minimizing the normalized cut is a NP-hard problem and can only be optimized approximately.

An interactive scheme can be adopted, where the user has to provide seeds to initialize the algorithm (Boykov and Jolly, 2001) (cf. figure2.2.3). As a result, the segmentation consists in a optimal separation of the seeded regions and is not biased towards small cuts. No intensity prior is required, but one can be estimated simply by computing the histogram of the marked regions, thus enforcing intensity coherence as well as spatial coherence.

2.2.4 *Shape priors and graph-cuts*

A "star shape prior" was proposed by Veksler (2008), allowing to constrain the shape of objects with *star shapes*. An object O has a star shape if there exists a point c such that any point p in O can be reached in straight line from c without stepping out of O. The shape prior is enforced by setting special weights s_{ij} to edges connecting pixels i and j which both lie on a line passing through c. Assuming pixel j is between c and pixel i, these special weights have the following form:

$$s_{ij}\left(x_i, x_j\right) = \begin{cases} 0 & \text{if } x_i = x_j, \\ \infty & \text{if } x_i = 1,\ x_j = 0 \\ \beta & \text{if } x_i = 0,\ x_j = 1, \end{cases}, \tag{2.2.15}$$

to be added to the original weights w_{ij}. This way, if pixel i belongs to the target object, then pixel j has to belong to the object as well. The center point c is provided by the user.

A "topological" prior was proposed by Delong and Boykov (2009), . Topological constraints are, for instance: containment ("object A contains object B"), exclusion ("object A and object B are not superimposed"), etc. These constraints are enforced in the binary graph-cut framework by solving simultaneous binary problems – one layer per label – and adding edges between the pixels of the different layers. These additional edges enforce the aforementioned topological constraints. Similarly, a topological "location" prior was proposed by Chittajallu et al. (2009), relying on prior knowledge on the location of other objects: e.g. having segmented the lungs in CT-scan of the torso, the position of these organs provides a prior on the location of the heart.

A shape prior based on a single reference segmentation was introduced to the graph-cut framework by Freedman and Zhang (2005). Let us denote $\mathcal{C}$ the boundary of this template shape, and ϕ the unsigned distance function whose o-level set is $\mathcal{C}$: $\mathcal{C} = \{p \in \Omega | \phi(p) = 0\}$. Then, with $p(i)$ denoting the spatial location of pixel i, the additional binary weight function is used:

$$s_{ij}\left(x_i, x_j\right) = \begin{cases} 0 & \text{if } x_i = x_j, \\ \phi\left(\frac{p(i)+p(j)}{2}\right) & \text{otherwise}, \end{cases} \tag{2.2.16}$$

which gives a low cost if the label transition (i.e. the boundary of the object) is located near the boundary of the reference shape.

2.2.5 *Segmentation with Random Walks*

We refer the reader to chapter 4 for detailed explanations on this approach.

2.3 PRIOR ART LIMITATIONS

In this chapter, we have tried to review all existing segmentation methods which could adapt to muscle segmentation, highlighting those which were actually applied to this task. Although there were few attempts at segmenting striated skeletal muscles, the existing methods adopted very different approaches. Among the surface-based models, Gilles and Pai (2008); Gilles and Magnenat-Thalmann (2010) adopted the active surface approach: a reference shape is deformed according to internal regularization forces and external image-driven forces in a local optimization process (pp. 17). Shape prior is enforced by a "shape memory" term (Gilles and Magnenat-Thalmann, 2010) or via a soft tissue simulation algorithm called shape-matching (Gilles and Pai, 2008). Both constraints enforce a strong prior, and the results is always valid in terms of anatomy. However, the local optimization and the possibly unreliable data-terms (gradient driven) sometimes yield over-regularized solutions. The method of Essafi et al. (2009) also uses a surface model, and thus suffers from the same problems. However, due to using a hierarchical statistical prior models, one can expect the surface model to adapt more easily to the specifics of the target image (pp. 23). The method of Wang et al. (2010) has the advantage of being optimized globally, as candidate targets for the surface vertices are searched in all the image. Second, the higher-order clique approach allows to learn a statistical shape prior and to enforce local similitude invariance in a practical way. However, like other landmark-based approaches, it relies on finding consistent detectable anatomical points, which we believe is very difficult with skeletal muscles (pp. 29). Like most surface models, the latter two methods may also suffer from topological issues such as intersection during optimization in a multi-object context, since this case was not addressed. Finally, the method of Andrews et al. (2011a) benefits from its region-based approach: global optimization and no topological issues (pp. 31). Shape prior is efficiently enforced for the entire limb by imposing the shape to evolve in a statistical shape space of valid solutions, built from a training atlas base through principal component analysis. The presented results demonstrate excellent performances. However, one weakness may remain in the image term, which relies on contour detection, an unreliable process in MR images of muscles where contours are often missing or often confused with spurious structures.

In chapter 4, we present some quantitative comparisons of the methods of Gilles and Pai (2008) and Andrews et al. (2011a) with our own proposed methods.

3

AUTOMATIC SEED PLACEMENT

3.1 INTRODUCTION

The approach presented in this part builds upon the observation that certain region-based methods can achieve excellent results, providing a substantial manual initialization. Among them, the Random Walks (RW) segmentation algorithm (Grady, 2006) has the advantage of addressing the issue of incomplete contours, which is one of the main difficulties in our task. The RW algorithm is a pixel-wise graph-based method which amounts to computing the probabilities of assignment of each pixel to each label (e.g. the index of a muscle). In this approach, a quadratic energy functional measuring the conformity of the segmentation with the image is defined and minimized by solving a large sparse linear system. Since efficient optimization algorithms exist to solve such problem – e.g. the conjugate gradient method – large volumes can be entirely segmented in a few minutes (see chapter 4 for a more in-depth view of the RW method). The interactive phase of the RW method consists in manually annotating some pixels of each label: the *seeds* (cf. figure 3.1.1). However, while this approach offers a considerable time gain by avoiding to manually annotate all the pixels, manually positioning the seeds remains a time consuming process, especially when large volumes are to be segmented.

In this chapter, we propose a method for generating the seeds automatically. While the large inter-subject variations of skeletal muscles prevent using a deformable model based on a shape template, the relative positions of the muscles are consistent among individuals. Therefore, it is reasonable to develop a segmentation method whose prior information is not based on the position of the boundaries and on the shape of the objects, but rather on the approximate relative position of the objects and the existence of boundaries between them. Seeds are compatible with this idea: it is sufficient that they are placed well inside the target object and the Random Walks algorithm will segment the contours accurately. Through a sampling process, unlabeled seeds are generated in the image domain. The clustering of these seeds with respect to the different muscle classes is achieved through a MRF graph-based approach: each seed is considered as a node and is to be assigned to a label – i.e. a muscle index. The weight of the edges between the seeds is set to reflect the topological information: the presence – or absence – of a boundary between two seeds and the likelihood of the orientation of an edge with respect to the assigned muscle indices. A discrete energy functional is defined to measure the the quality of a global label assignment. Due to being NP-hard, this problem can only be approximately solved; however, we obtained

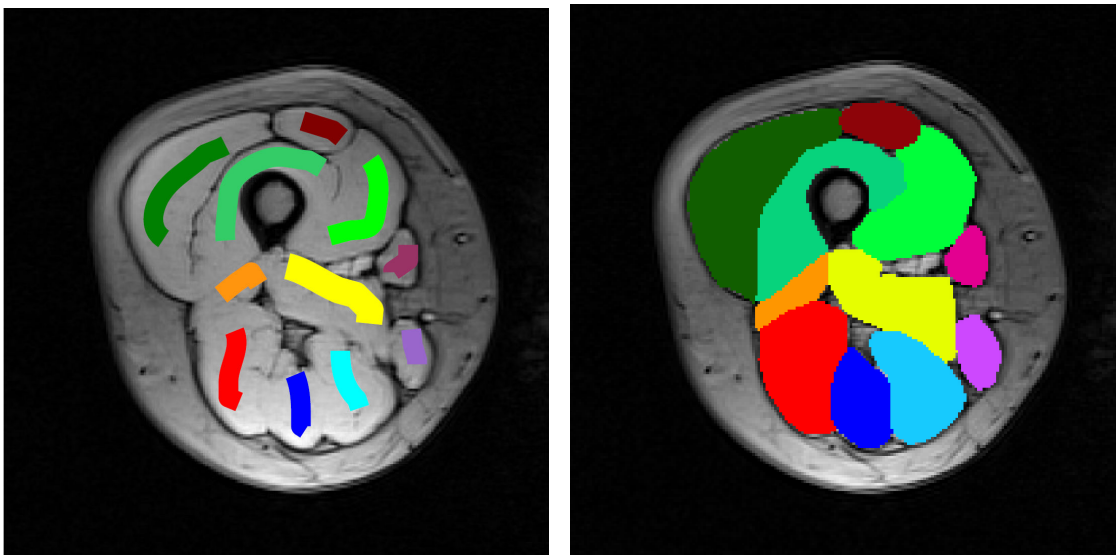

Figure 3.1.1 – (left) Axial cross-section of a right thigh with manually drawn seeds; (right) segmentation.

satisfying results using an efficient message passing algorithm (Kolmogorov, 2006) on a real clinical dataset.

3.2 RELATED WORK

To our knowledge, this is the first method proposing an automatic seed generation based on a graph formulation of the labeling problem. Previous automatic seeding methods may be associated with graph-cuts (Boykov and Jolly, 2001), seeded region growing (Adams and Bischof, 1994) or Random Walks. The more straightforward approaches rely on appearance filtering for selecting seeds: only pixels of a certain predetermined intensity or color are kept as seeds (Wyatt et al., 2000; Wighton et al., 2009). Similarly, some methods (Fan et al., 2001) first detect certain images features (e.g. edges) and generate seeds according to geometrical considerations (e.g. estimated centroids). Integration of knowledge of the anatomy can be based on pre-defined distances and relative positions (Wyatt et al., 2000; Maier et al., 2008), but such approaches require small inter-subject variations or few objects to segment, with simple geometries. Finally, there also exists "pre-segmentation" methods, destined to partition the image into several regions to be latter merged to obtain semantically correct regions (Mičušík and Hanbury, 2006); such methods do not rely on any predefined topology, segmenting an unknown number of objets, with unknown shapes.

Figure 3.3.1 – Binary mask (right) generated from water map image (left). This mask is used for registering the reference image onto the test image.

3.3 PREREQUISITES

We need a binary mask of the muscle tissues, separating them from the background and the fat tissues (see figure 3.3.1). Such a mask can be obtained via intensity thresholding – or in our case, more efficiently, from the "water-map" images we based on for our experiments. This water-map represents the fraction of water protons in the organic tissues; as a result, muscles – made of water for a large part – have high intensity values, whereas fat tissues and bones have intensity values close to zero (cf. appendix A for more details). This mask is used to ensure the seeds are only generated within the muscles.

This approach assumes we have a *reference atlas* rigidly registered with the image to be segmented (see figure 3.3.2). We applied rigid-registration (translation, rotation, scaling) using the free registration software medInria. This software uses the block matching rigid registration algorithm (Ourselin et al., 2000). To reduce the possibility of registration errors, we used the binary masks of the muscle tissues as input images for the registration software. This way, only the muscle/fat boundaries are taken into account for leading the registration algorithm.

3.4 UNLABELED SEEDS SAMPLING

The first step of our algorithm is to generate the seeds that will be used as initialization by the RW algorithm once labeled. We are going to place the seeds well inside the muscles, and not too close to the contours since the task of segmenting the contours

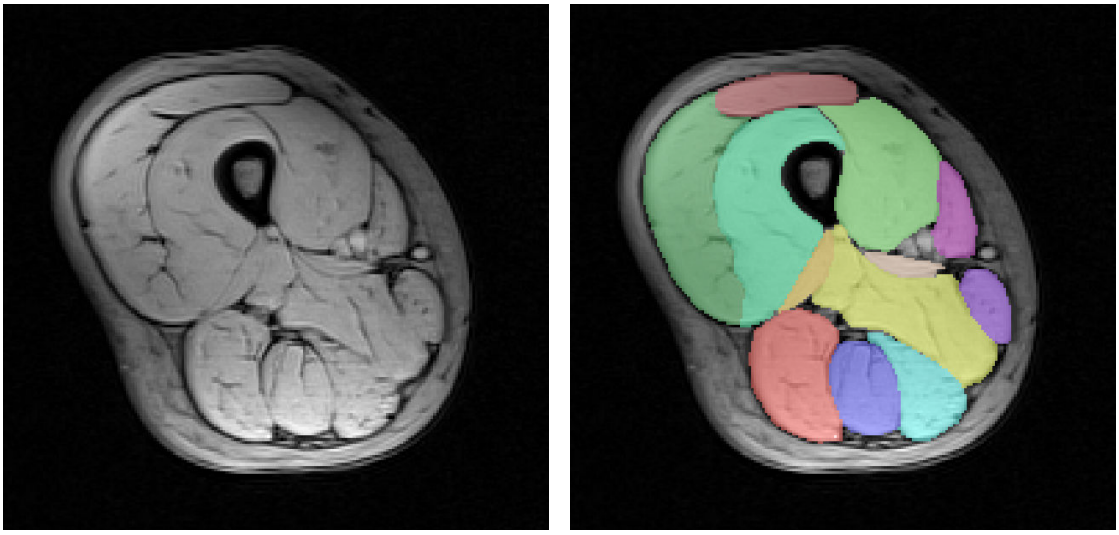

Figure 3.3.2 – (left) Axial cross-section of a right thigh; (right) manual segmentation. The reference segmentation will serve as a reference atlas when segmenting a new image.

accurately is left to the RW method. To achieve this, we first compute an edge map of our volume – e.g. by computing the local variance of the intensity on a small patch centered on each pixel. Then, we sample the seeds on a regular grid, with a high enough density to ensure all the muscles contain several seeds. Finally, we move the seeds in the opposite direction of the gradient of the edge map for a few iterations. More precisely, if the spacing of the grid is k, then the seeds are allowed to move by $k/2$ from their original position, to prevent any pair of moved seeds to take the same location. This way, seeds are moved away from the close-by edges by a few pixels (see figure 3.4.1).

The seed displacement algorithm can be summed-up by the following algorithm:

1. let us denote the initial location of voxel p as $\mathbf{z}_p^0$;
2. for $i = 1 \dots \frac{k}{2}$ do:
   a) compute normalized displacement vector $\mathbf{v}_p^i = -\left\lfloor \nabla I\left(\mathbf{z}_p\right) / \|\nabla I\left(\mathbf{z}_p\right)\| + 0.5 \right\rfloor$.
   b) shift seed by $\mathbf{v}$: $\mathbf{z}_p^i \leftarrow \mathbf{z}_p^{i-1} + \mathbf{v}^i$;

The next step is the label assignment.

3.5 MRF FORMULATION

We now describe how to assign a label to the seeds we generated, according to the similarities between the test image and the reference segmentation.

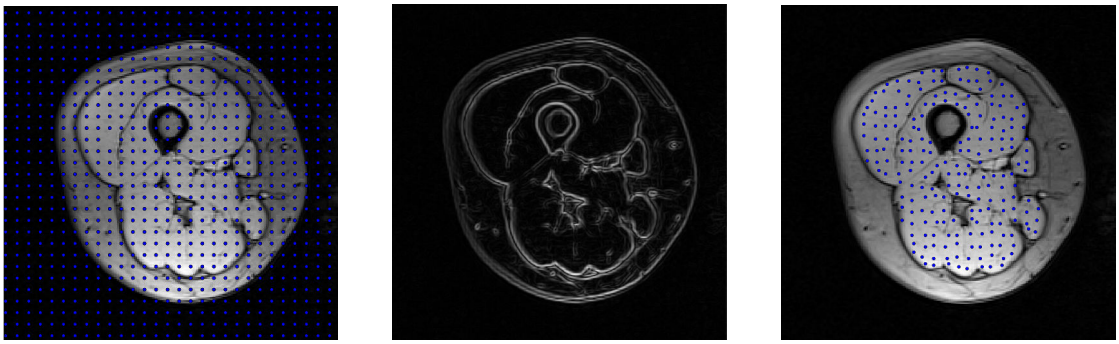

Figure 3.4.1 – Generating the unlabeled seeds: (left) points are sampled regularly over the test image with a spacing of k pixels; (center) edge map; (right) unlabeled seeds, after removal of the points lying outside of the mask of the muscles, and after displacement of $k/2$ pixels in the opposite direction of the gradient of the edge map.

3.5.1 *Energy form*

We formulate our problem as a labeling problem, that we will solve by minimizing a first-order Markov Random Fields energy functional. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. Given a set of labels $\mathcal{L}$, we want to assign a label $l \in \mathcal{L}$ to each node $p \in \mathcal{V}$. We denote x_p the label assigned to p, and $\mathbf{x}$ the collection of all assignments. A first order MRF functional has the form:

$$E(\mathbf{x}) = \sum_{p \in \mathcal{V}} \theta_p(x_p) + \sum_{(p,q) \in \mathcal{V}} \theta_{p,q}(x_p, x_q), \tag{3.5.1}$$

where the unary potential $\theta_p(\cdot)$ depends only on the label assigned to node p, and the binary potential $\theta_{p,q}(\cdot,\cdot)$ depends on the labels assigned to each nodes of edge (p,q). Unary potentials (or "costs") measure the quality of the label assignment of each node. Similarly, Binary potentials measure the quality of the label assignment of a pair of nodes. The quality measures we are referring to are specific to the desired application. The interest of the MRF formulation is that the assignment of each node depends only on the neighborhood of this node. In the simple case where only pairwise dependencies are considered, efficient optimization algorithm can be used to minimized the previous functional.

In this framework, the nodes in $\mathcal{V}$ are the unlabeled seeds and the labels in $\mathcal{L}$ are the indices of the muscles in the reference segmentation. In the following, we define the unary potentials so that seeds close to (respectively far from) a muscle in the reference image have a small (respectively large) assignment cost. Binary potentials will reflect the topological properties we seek to exploit: when assigning two different labels (i.e. muscles), unlikely orientations and the absence of boundary between the connected seeds will yield a large binary cost; reciprocally, when assigning the same label to both

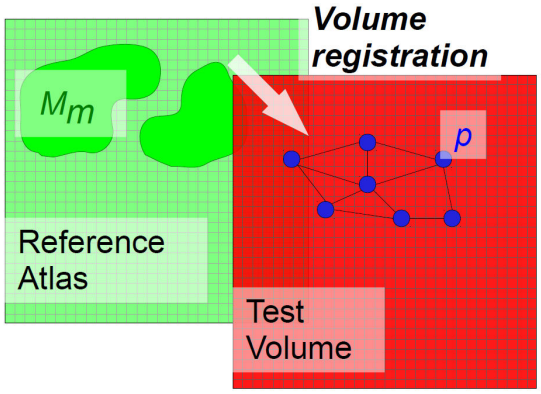

Figure 3.5.1 – The reference atlas is registered onto the test volume, providing prior-knowledge on label assignment likelihood for each node.

ends of an edge, it is the presence of a boundary between the connected seeds which will yield a large cost (orientations do not matter is this case). As we will see in a later part, the set of edges will be set to model the spatial proximity of the nodes.

3.5.2 *Unary potential*

We define a unary cost $\theta_p(\cdot)$ based on the knowledge that the reference atlas is rigidly registered onto the test image:

$$\theta_p(m) = d(p, M_m), \tag{3.5.2}$$

where M_m is the set of pixels with label m in the reference segmentation and $d(\cdot,\cdot)$ is a distance function between a point and a set:

$$d(p,R) = \begin{cases} \min\{\|p-r\|, r \in R\} & \text{if } p \notin R, \\ 0 & \text{otherwise.} \end{cases} \tag{3.5.3}$$

The rigid registration ensures that, if we superimpose the test image and the reference segmentation, an unknown muscle in the test image is likely to correspond to a geometrically close muscle in the reference segmentation. Therefore, this term will favor the assignment to a node of a label corresponding to a geometrically close muscle, over the label of a more distant muscle (see figure 3.5.1).

Geodesic distance potential and graph connectivity

The unary term does not take into account the structure of the set of muscles. As we saw previously, the shape of the muscles can vary quite extensively between in-

dividuals. On the other hand, the topology of the set of muscles is consistent: muscle a is *always* "close to" muscle b, but "far from" muscle c, there is *most of the time* a visible contour between muscle a and b. We designed binary terms to account for this knowledge. The binary potential is the sum of two potentials: $\theta_{p,q}(\cdot,\cdot) = w_{geod}\theta^{geod}_{p,q}(\cdot,\cdot) + w_{orient}\theta^{orient}_{p,q}(\cdot,\cdot)$, weighted by parameters w_{geod} and w_{orient}.

The term $\theta^{geod}_{p,q}(\cdot,\cdot)$ is derived from the *geodesic distance* $g(p,q)$ between the seeds p and q. This geodesic distance between two points is understood as the length of the shortest path from one point to another over a curved space. Here, we consider the edge-map of the test image as a curved space: a ground with "hills" – i.e. edges – and "valleys" – i.e. homogeneous regions. The interest of such measure is the following: considering two edges of the same Euclidean length – an edge crossing a boundary has a larger geodesic length than a non-boundary crossing edge. We propose the following definition:

$$\theta^{geod}_{p,q}(x_p, x_q) \propto \begin{cases} 1/\left(1+\exp\left(\gamma - g(p,q)\right)\right) & \text{if } x_p = x_q, \\ 1/\left(1+\exp\left(g(p,q) - \gamma\right)\right) & \text{otherwise.} \end{cases} \tag{3.5.4}$$

where γ is a free parameter. If $x_p = x_q$, i.e. both seeds belong to the same muscle, $\theta^{geod}_{p,q}(x_p, x_q)$ will penalize large geodesic distances, as there should be no contour between p and q. On the other hand, $\theta^{geod}_{p,q}(x_p, x_q)$ will favor large geodesic distances if $x_p \neq x_q$ (see figure 3.5.2).

We compute an approximate geodesic distance between the seeds using the Fast Marching algorithm (Sethian, 1996) on the test image. The Fast Marching method solves the general static Hamilton-Jacobi equation for a given speed function. Here, the speed is determined by the value of the edge-map (e.g. the gradient magnitude). A propagating discrete front is started at each node and marches in the outward direction at the given spatially-dependent speed. From a technical view, an efficient heap-sort is used to select the best front pixel to update, so that previously evaluated pixels are seen only once. Since there are several fronts evolving simultaneously, the origin point of each front is stored for each visited pixel. Once all pixels have been visited, the arrival time of the front at a given pixel can be seen as a geodesic distance from this pixel to the origin pixel of the front.

We supply the algorithm with the unlabeled sampled seeds. We obtain in return a partition of the image, where each seed in included in a different region, and where the boundary between two regions is equidistant to the seeds in the regions, in the geodesic sense (i.e. a geodesic Voronoi partition). The connectivity of the graph is determined by the Voronoi partition: for each adjacent region, we add an edge between the corresponding nodes. Note that we do not obtain an edge between every pair of seeds, but only between geometrically close seeds. This is not a problem, since $\theta^{geod}_{p,q}$ is designed to account for the presence or the absence of a contour between two seeds,

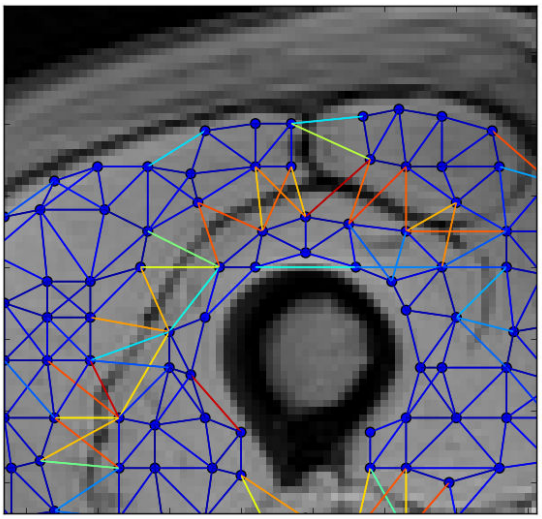

Figure 3.5.2 – Visual representation of the geodesic distance between seeds: the color-map goes from blue to red as the geodesic distance increases. Notice that edges crossing a visible boundary have a large geodesic distance whereas edges crossing a homogeneous area have a small geodesic distance.

and thus is mostly a local term; our unary cost coarsely handles the labeling problem at a larger scale.

Relative orientation potential

The term $\theta_{p,q}^{orient}(\cdot,\cdot)$ is intended to ensure that the relative position of a pair of muscles is maintained, e.g. muscle a is located "top right of" muscle b. In a pre-processing stage using the reference atlas, we computed the distribution of the orientations for each pair (m,n) of muscles. More precisely, we estimated the probability density $p_{m,n}(\cdot)$ of random variable

$$\mathbf{U} = (P_m - P_n) \,/\, \|P_m - P_n\|\,, \tag{3.5.5}$$

where P_m is a randomly chosen point in muscle M_m, and

$$P_n = \arg\min_{p \in M_n} \|P_m - p\| \tag{3.5.6}$$

(see figure 3.5.3). Thus, $p_{m,n}(\mathbf{u})$ can be seen as the probability that, for any pixel in muscle m, the direction of the closest pixel in muscle n is $\mathbf{u}$. We define the orientation cost as:

$$\theta_{p,q}^{orient}(x_p, x_q) \propto \begin{cases} -\log P_{x_p,x_q}(p-q) & \text{if } x_p \neq x_q, \\ 0 & \text{otherwise.} \end{cases} \tag{3.5.7}$$

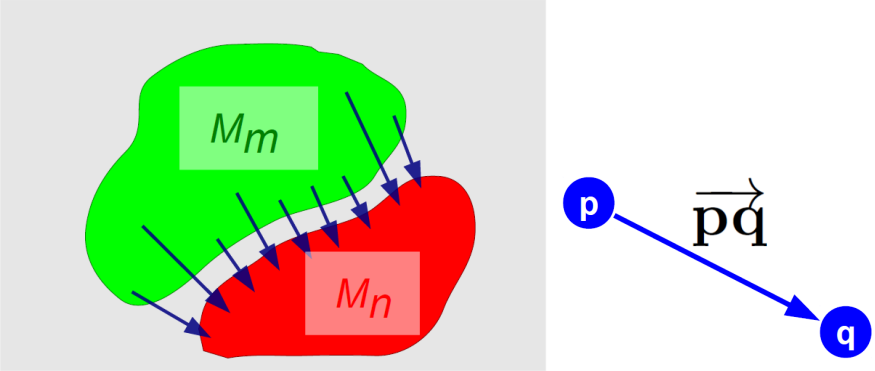

Figure 3.5.3 – (left) the probability density of the orientations between muscles M_m and M_n is estimated by sampling points within M_m and determining the closest point in M_n. (right) the cost of an edge between nodes p and q depends on the probability of the orientation of vector $\overrightarrow{pq}$.

This cost will be large if the orientation of the edge between two nodes is unlikely given the labels to which the nodes are assigned.

3.6 EXPERIMENTAL VALIDATION

3.6.1 *MRF Optimization*

Minimizing the energy 3.5.1 is a NP-hard problem for the binary potential function is arbitrary. However, due to the relatively small number of variables we can expect to reach a good approximate in reasonable time with an efficient message passing algorithm such as the convergent Tree-Reweighed (TRW) Message Passing algorithm (Kolmogorov, 2006). The TRW algorithm uses a linear programming relaxation technique and operates on the dual of the relaxed problem in order to find a lower bound to the energy of the original problem. Due to its state of the art performance among discrete optimization methods, TRW has been applied to solve many computer vision and medical imaging tasks.

3.6.2 *Experimental Results*

To evaluate our method, we used a dataset composed of 3D volumes of the right thigh of 15 healthy subjects, covering a wide range of morphologies (8 females, 7 males, ages range: 26 to 60), acquired with a 3T Siemens scanner and using 3pt Dixon sequence (TR=10ms, TE1=2.75 ms TE2=3.95 ms TE3=5.15 ms, rf. flip angle =3°; we used the out-of-phase image), with the an average image size of: 220px $\times$ 220px $\times$ 64px. Every volume was manually segmented to obtain the ground truth against which the

segmentation results are compared. We focused our evaluation on clinically relevant muscles of the thigh (13 muscles).

Sampling the seeds, as described in section 3.4, with grid spacing 7px × 7px × 7px, and 2 iterations to bring the seeds further away from the edges (filter radius = 1px), gave us between 2000 and 2500 seeds per test image. At the MRF stage, detailed in section 3.5, we obtained our best results with the following cross-validation estimated parameters $w_{geod} = 0.1$, $w_{orient} = 0.01$, $\gamma = 2 \times \text{median}_{p,q \in \mathcal{V}} g(p,q)$, and U contains 26 regularly spread unitary orientation vectors. On a 2.8 GHz Intel® processor with 4 GB of RAM, total processing time is under 5 min, which adds up to the RW segmentation time (5 min).

All our results are based on a leave-one-out validation protocol: each volume is used as the reference segmentation for segmenting all the other volumes. First, we evaluated the automatic labeling process by computing the labeling error rate for each muscle (figure 3.6.1) as box-plots[1]. For testing the efficiency of the binary potential, we also computed the error rate for an energy only composed of the unary potential. We observe that using the binary potential significantly reduce the error rate. Moreover, the combination of the geodesic potential with the orientation potential is more efficient than using only the geodesic potential, which confirms that both terms exploit complementary types of information.

Then, we computed Dice coefficients to evaluate the segmentation results given by the RW with the labeled seeds (figures 3.6.2 and 3.6.3). The expression of the Dice coefficient is: $D = 2|T \cap R| \,/\, (|T| + |R|)$, where T and R are the pixel sets of a given muscle in the inferred segmentation and the reference segmentation respectively. The presented method outperforms a simple Atlas-based segmentation method (see chapter 4 for details on this method).

In figure 3.6.4, we show cross-sections of segmentation results. We observe that small muscles tend to be more affected by segmentation errors than large ones, which points out the limitations of our model. In particular, the unary term, based on the superimposition of the registered reference atlas with the target image, is bound to be less effective for small muscles, because the distance term $d(.,.)$ is likely to be larger than in the case of large muscles. Besides, many errors are due to the absence of reliable contours between parts of muscles, which, in the case of large errors, indicates that the topology constraints are failing to insure topology or shape correctness.

3.7 CONCLUSION

In this chapter we presented a first method relying on the good properties of the RW algorithm when facing weak boundaries. Results on real clinical data show the valid-

1 Box-plot presentation: the boxes contain the middle 50% of the data and the median value, and the extremities of the lines indicate the min and max values, excluding the outliers (for more details, see the documentation of Matplotlib).

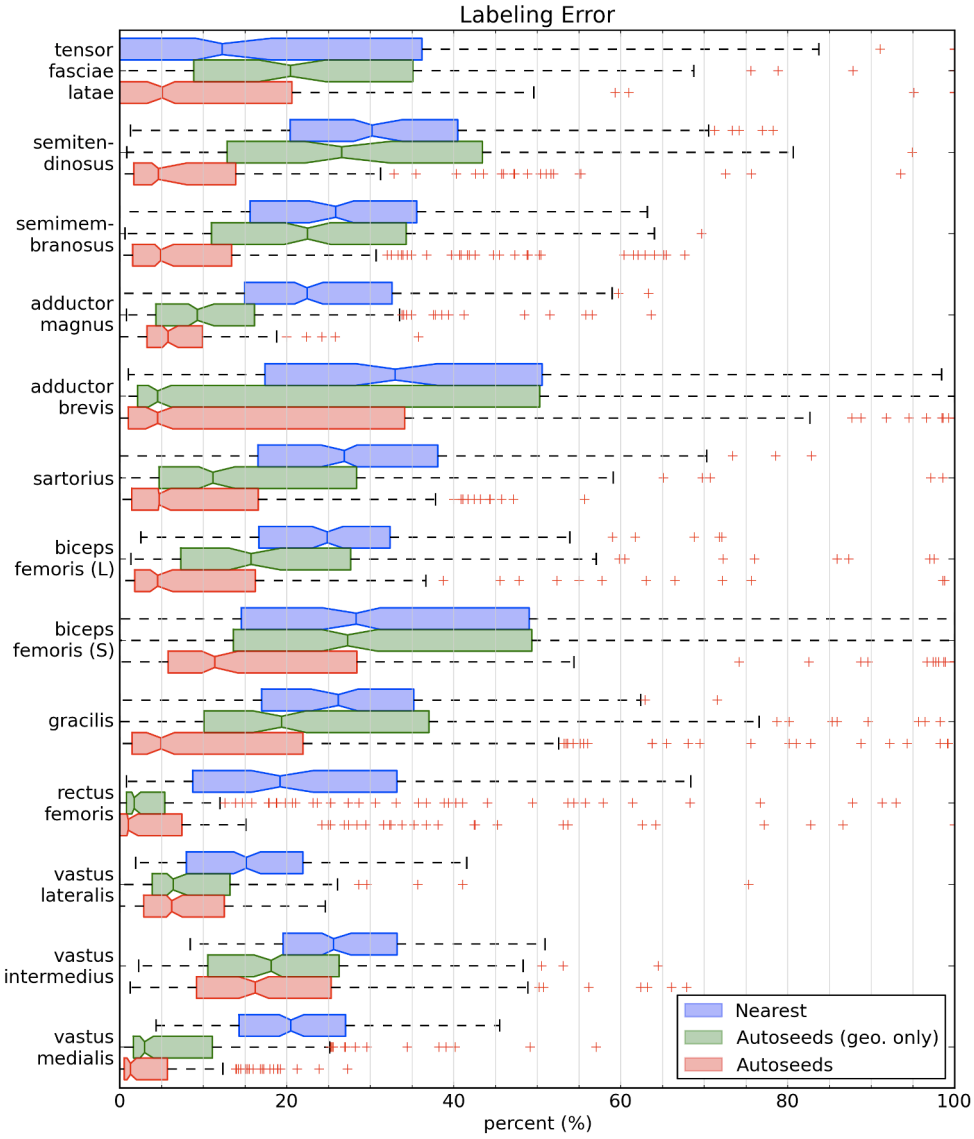

Figure 3.6.1 – Labeling error rate (in percent) as box-plots. "Nearest" is when using only the unary potential ($w_{\text{geod}} = w_{\text{orient}} = 0$). The label of the seeds is only determined by the geometrically nearest muscle in the reference atlas. "Autoseed (geo. only)" is the error rate when using only the geodesic distance term ($w_{orient} = 0$, $w_{geod} = 0.1$). "Autoseeds" refers to the proposed method. Using both binary potentials significantly reduces the error rates.

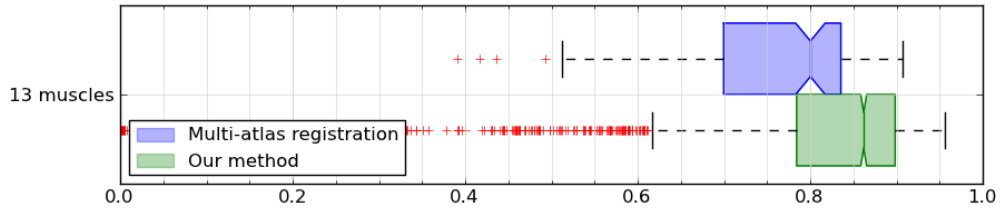

Figure 3.6.2 – Dice coefficients for all muscles as box-plots "Multi atlas registration" refers to a simple atlas registration-based segmentation method (see next chapter for more details). The expression of the Dice coefficient is: $D = 2\,|T \cap R| \,/\, (|T| + |R|)$, where T and R are the pixel sets of a given muscle in the inferred segmentation and the reference segmentation respectively.

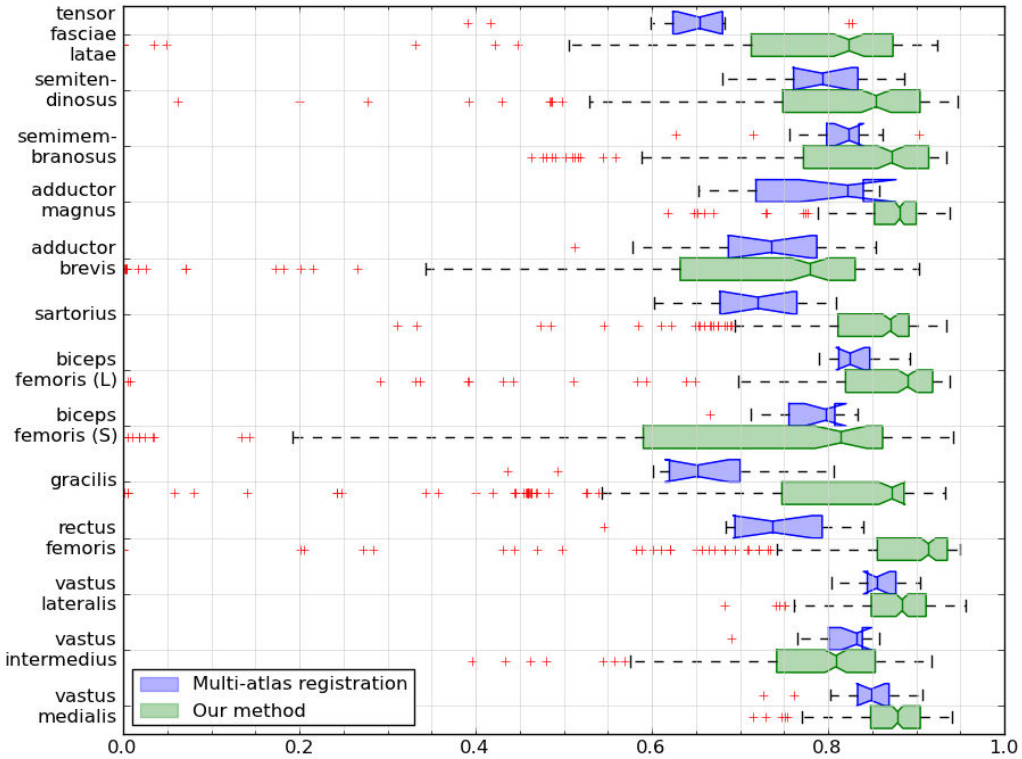

Figure 3.6.3 – Detailed Dice results for all muscles separately.

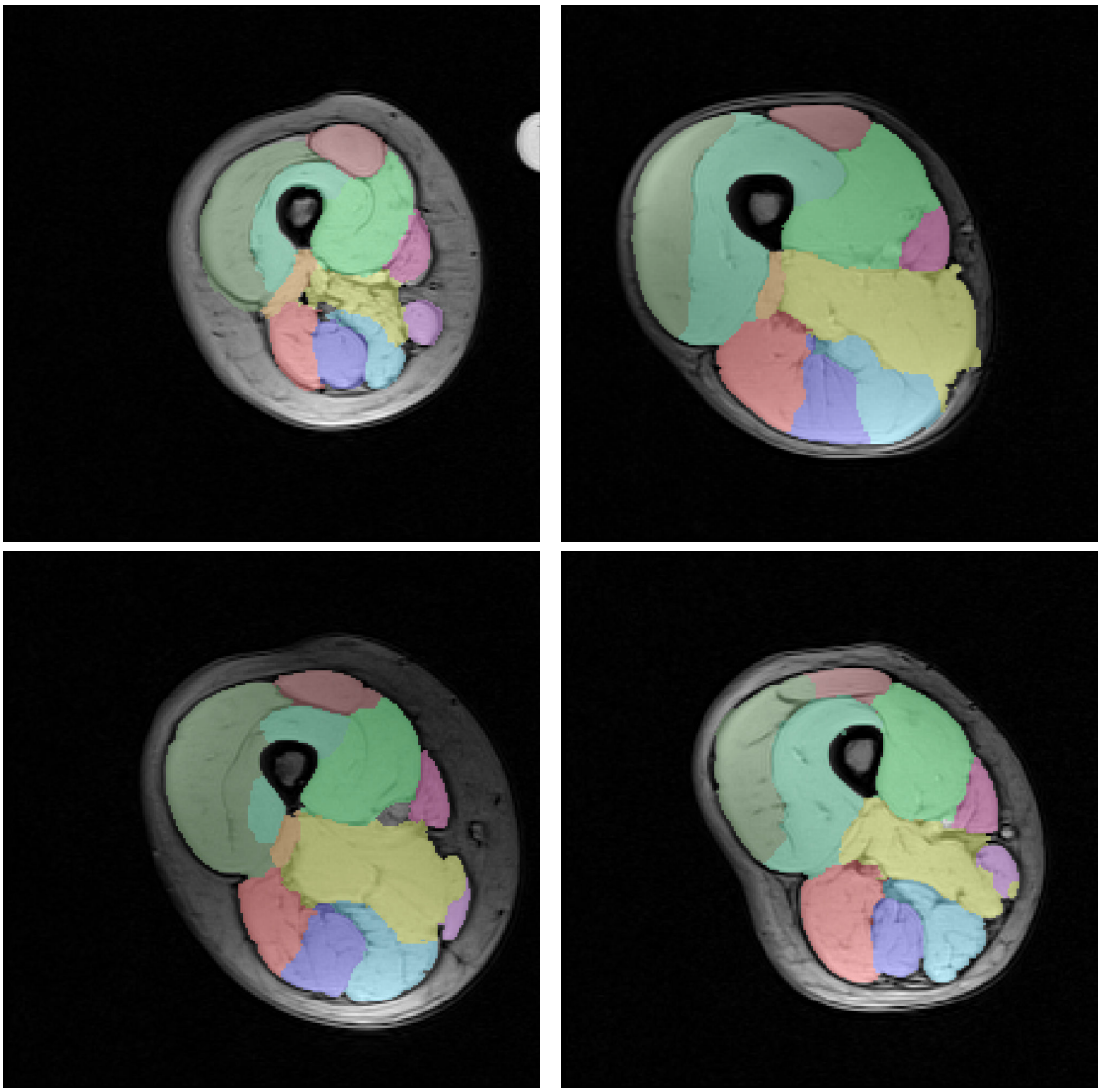

Figure 3.6.4 – Examples of cross-sections of 3d volumes of right thighs automatically segmented with the proposed method.

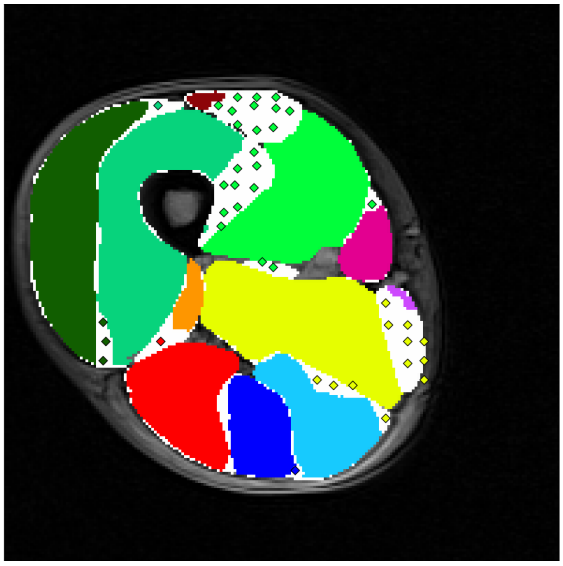

Figure 3.7.1 – Incorrectly labeled seeds (diamond markers) prevent the RW method to segment the corresponding regions correctly (white regions)

ity of this approach and the usefulness of the proposed two binary terms. However, an inherent limitation of this method lies in its two-stage formulation: 1) label the seeds; 2) segment the volume from the seeds. Indeed, as shown in figure 3.7.1, if the labeling fails, segmentation will inevitably fail in the same area since seeds act as constraints on the segmentation process (see the Random Walks algorithm detailed presentation in the next chapter). This issue may be revealed by the non-negligible amount of outliers in the Dice plots (Fig. 3.6.2), which means that the method totally failed to segment some muscles (in general, smaller ones, like the biceps femoris or the adductor brevis). In order to avoid such propagation of errors, we propose an integrated method in the next chapter, in which the Random Walks and prior knowledge of shape are combined into one formulation. Although this method is inherently flawed for the aforementioned reasons, some improvements could still be considered. In particular, using non-rigid registration could improve the performance if a better match of the atlas and the segmented image is achieved.

4

RANDOM WALKS SEGMENTATION AND PRIOR KNOWLEDGE

In this chapter, we present different approaches based on the general Random Walks Segmentation algorithm. The state of the art RW algorithm has several benefits over its competitors: (i) unlike discrete methods such as graph-cuts, it provides a probabilistic segmentation (that is, a distribution over all putative labels for each pixel/voxel); (ii) unlike surface-based methods like Active Contours, it has the ability to handle intersection issues in multi-label cases; (iii) it is robust to partial contours; and (iv) it is computationally efficient, since it only requires solving a sparse linear system of equations. Although it was initially proposed for the interactive setting where a user specifies the seeds of the segmentation, a fully automated extension using the contrast information was proposed by (Grady, 2006). However, contrast information does not permit accurate segmentation in the skeletal muscle case where all muscles share the same appearance. Therefore, we propose to incorporate prior knowledge of the *shape* of skeletal muscles into the RW framework. In a first published approach (Baudin et al., 2012a), a prior energy functional is added to the RW functional in the form of a proximal regularization of the probabilistic segmentation, whose deviations from an average reference segmentation are penalized. We propose to modulate the strength of the model constraints according to both prior information on the model's local accuracy and to the strength of the contours present in the segmented image. A second published approach (Baudin et al., 2012b), introduces some flexibility to the model by constraining the shape to remain close to a Principal Component Analysis shape space built from training examples. Using the PCA allows us to model complex non-rigid shape variations relying on a few eigen-modes. These methods benefit from the high performance of the Random Walks optimization process, as they require only a simple addition to the original objective function.

4.1 THE RANDOM WALKS SEGMENTATION ALGORITHM

4.1.1 *Preliminaries*

From an image I with N pixels, we build a undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V}$ representing the set of nodes, $|\mathcal{V}| = N$, and $\mathcal{E}$ the set of edges, where the i-th node v_i corresponds to the i-th pixel of image I. We denote the edge connecting the nodes with indices i and j as e_{ij}, and its weight as $w_{ij} \geq 0$. Since the graph is undirected, e_{ij} and e_{ji} denote the same edge and their weight is the same: $\omega_{ij} = \omega_{ji}$. The set of edges

$\mathcal{E}$ is only composed of pairs of adjacent pixels, such that graph $\mathcal{G}$ contains only cliques of order 0 and 1. We also denote the neighborhood of pixel i as $\mathcal{N}_i = \{v_j / e_{ij} \in \mathcal{E}\}$.

Given a set of labels $\mathcal{S}$ (e.g. the indices of the muscles), segmenting an image I is defined as a *graph partitioning* procedure, i.e. consisting in assigning a label $s \in \mathcal{S}$ to each node $v \in \mathcal{V}$. We refer to the assignment of node v_i to the label s as: $l(i) = s$.

4.1.2 *Random Walker analogy*

The Random Walks (or Random Walker) algorithm is a probabilistic method modeling the behavior of a inebriated walker on a non homogeneous ground. At each step, the walker chooses a random direction, where the chosen direction is more likely if it is an easy path – e.g. on a flat ground – than a difficult path – e.g. on a bumpy ground (see figure 4.1.1 for illustration). Suppose the ground is discretized, i.e. the walker can only take a finite number of steps. The ground can be represented by a graph, each node being one possible location. The edges of this graph are the possible paths from one location (node) to another.

The connection with graph partitioning – and thus image segmentation – is established as follows: suppose some of the locations on the ground are marked with a colored flag. The number of colors is finite and there can be several flags of the same color. In graph terms, some nodes in the graph are already assigned to one label, and the number of labels is finite. We refer to the marked nodes as *seeds*. Then, the Random Walks algorithm computes the probability that the walker reaches one flag before all the others, for *each starting location*. Finally, each starting location is assigned to the most probably reached-first flag; in graph terms: each node is assigned to the most probable label.

4.1.3 *Probabilistic explanation*[1]

In this framework, the global label assignment is modeled by a random variable. We denote the probability of assignment of voxel i to label s as $x_i^s = \Pr(l(i) = s)$. In this notation, $l(i) = \arg\max_s x_i^s$. We denote the transition probability from node i to node j as p_{ij}. In general, we have $p_{ij} \neq p_{ji}$.

Assume we possess a set $\mathcal{V}_M$ of seeds, i.e. pre-labeled nodes *for each label* (typically, manually marked voxels). We denote the set of unknown nodes as $\mathcal{V}_U$, such that $\mathcal{V}_M \cup \mathcal{V}_U = \mathcal{V}$ and $\mathcal{V}_M \cap \mathcal{V}_U = \emptyset$. For convenience in the notations, we denote the

1 Thanks to Danny Goodman for his help in this section.

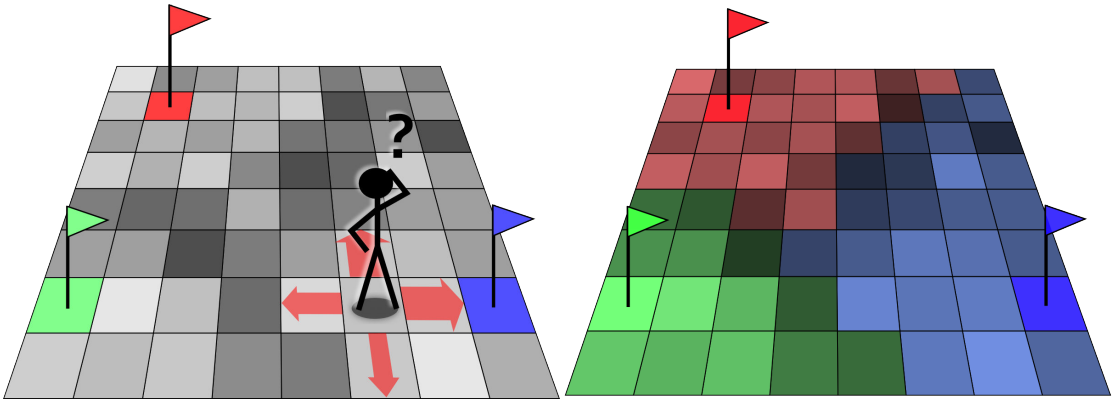

Figure 4.1.1 – (left) Random walker on a discretized ground with target seeds (flags). The probability of the walker of stepping onto an adjacent square/pixel can be different for each pixel, as figured by the red arrows of various sizes. (Right) Segmented ground/image: the label assigned to one pixel is the label of the most probably first reached seed during a random walk starting from that pixel.

sets of indices of the nodes in $\mathcal{V}_U$ and $\mathcal{V}_M$ as U and M respectively. Since the label of marked voxels is known for certain, their assignment probability is either 1 or 0

$$\forall i \in M,\ x_i^s = \begin{cases} 1 & l(i) = s, \\ 0 & l(i) \neq s. \end{cases} \tag{4.1.1}$$

We denote to the probability vector for label s as $\mathbf{x}^s$, where $\mathbf{x}^s$ contains the x_i^s for each voxel i. Then, without loss of generality, we can assume the variables in $\mathbf{x}^s$ are ordered so that we can write:

$$\mathbf{x}^s = \begin{bmatrix} \mathbf{x}_U^s \\ \mathbf{x}_M^s \end{bmatrix}, \tag{4.1.2}$$

where $\mathbf{x}_U^s$ (respectively $\mathbf{x}_M^s$) is the vector containing the assignment probabilities for unknown (respectively marked) nodes in $\mathcal{V}_U$ (respectively $\mathcal{V}_M$). Then, the assignment probability of any unknown node $v_i \in \mathcal{V}_U$ can be expressed with respect to the assignment probability of all the nodes (both unknown and marked) in its neighborhood $\mathcal{N}_i$:

$$\forall i \in U,\ x_i^s = \sum_{v_j \in \mathcal{N}_i} p_{ij} x_j^s. \tag{4.1.3}$$

The probability of assignment of one pixel depends only on the assignment probabilities of its neighbors. Since all nodes have the same property, the assignment

probabilities can all be represented at once in matrix form. We denote the transition probability matrix as Π:

$$\Pi = \begin{cases} p_{ij} & \text{if } e_{ij} \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1.4}$$

We denote the sub-matrix of Π containing the transition probabilities between two unknown nodes as Π_U (respectively: (i)Π_M; (ii)Π_{UM}; (iii)Π_{MU}) (respectively: (i) between two marked nodes: (ii) from an unknown node to a marked node; (iii) from a marked node to an unknown node):

$$\Pi = \begin{bmatrix} \Pi_U & \Pi_{UM} \\ \Pi_{MU} & \Pi_M \end{bmatrix}. \tag{4.1.5}$$

Thus, we can write:

$$\forall s,\ x_U^s = \Pi_U \mathbf{x}_U^s + \Pi_{UM} \mathbf{x}_M^s, \tag{4.1.6}$$

$$(I - \Pi_U)\, x_U^s = \Pi_{UM} \mathbf{x}_M^s, \tag{4.1.7}$$

with I being the identity matrix of size $|U| \times |U|$.

4.1.4 *Transition probabilities*

The transition probabilities are to be set accordingly to the desired properties of the segmentation. In general – and such is case of in muscle segmentation – one wishes to segment an image according to the visible boundaries. A boundary in an image is the result of an intensity pattern where many side-by-side pixels of very different intensities form a continuous curve In our probabilistic framework, this leads to have higher probability transitions for pairs of pixels having different intensities and lower probability transitions for pixels having similar intensities (cf. figure 4.1.2). A well-known choice for transition probability is the Gaussian kernel[2]:

$$w_{ij} = \exp\left(-\beta \left(I_i - I_j\right)^2\right), \tag{4.1.8}$$

where w_{ij} is thus defined as the weight for edge e_{ij}. Since p_{ij} is a probability, the transition probability is given by:

$$p_{ij} = \frac{\omega_{ij}}{\sum_i \omega_{ij}}. \tag{4.1.9}$$

2 we also implemented the following alternate formulation $w_{ij} = \frac{1}{\beta |I_i - I_j| + \varepsilon}$.

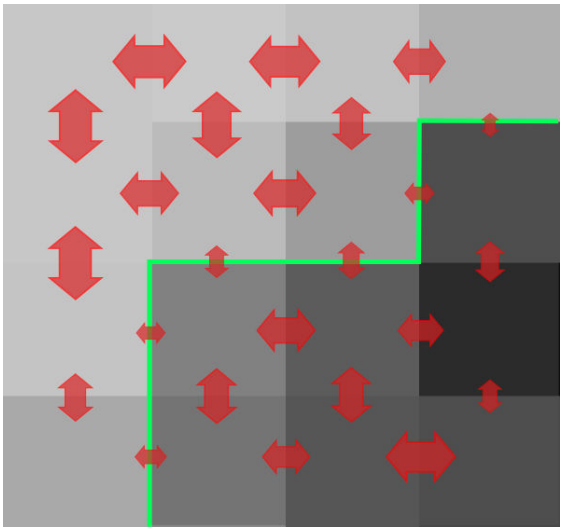

Figure 4.1.2 – Image contrast is chosen as the source for the transition probabilities (figured by red arrows). The green curve, following the sequence of lowest transition probabilities, is more likely to be selected as a boundary during the segmentation.

4.1.5 *Random Walks objective functional*

Let us define A, the non-normalized transition matrix – also designated as the *affinity matrix*:

$$A_{ij} = \begin{cases} \omega_{ij} & \text{if } e_{ij} \in \mathcal{E}. \\ 0 & \text{otherwise,} \end{cases} \tag{4.1.10}$$

and D, the diagonal matrix such that:

$$D_{ii} = \sum_j \omega_{ij}, \tag{4.1.11}$$

and L, the *un-normalized combinatorial Laplacian matrix*

$$L = D - A. \tag{4.1.12}$$

The *Random Walks objective functional* is defined as:

$$E^s_{\text{RW}}(\mathbf{x}^s) = \frac{1}{2}\mathbf{x}^{s\top} L \mathbf{x}^s. \tag{4.1.13}$$

In the following, we will see that minimizing this functional amounts to solving the probabilistic Random Walks equation 4.1.3, hence determining the assignment probabilities. Since $w_{ij} = w_{ji}$ (it is an undirected graph), the entries of L are:

$$L_{i,j} = \begin{cases} \sum_k \omega_{kj} & \text{if } i = j, \\ -\omega_{ij} & \text{if } e_{ij} \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1.14}$$

The Laplacian L has many properties (Von Luxburg, 2007), among which the most important ones are:

1. L is symmetric and positive semi-definite; the latter can be seen by reformulating 4.1.13 in a scalar form:

$$E^s_{\text{RW}}(\mathbf{x}^s) = \sum_{e_{ij} \in \mathcal{E}} \omega_{ij} \left(x^s_i - x^s_j \right)^2, \tag{4.1.15}$$

   which gives:

$$\forall \mathbf{a} \in \mathbb{R}^N, \ \mathbf{a}^\top L \mathbf{a} \geq 0; \tag{4.1.16}$$

2. the smallest eigenvalue of L is 0, and the corresponding eigenvector is the constant one vector $\mathbf{1}$.

We decompose L into sub-blocks for marked and unknown nodes, hence defining sub-blocks L_U, L_M and B:

$$L = \begin{bmatrix} L_U & B \\ B^\top & L_M \end{bmatrix},$$

which allows us to rewrite equation 4.1.13 as:

$$E_{\text{RW}}(\mathbf{x}^s) = \frac{1}{2} \mathbf{x}^{s\top}_U L_U \mathbf{x}^s_U + \mathbf{x}^{s\top}_U B \mathbf{x}^s_M + \frac{1}{2} \mathbf{x}^{s\top}_M L_M \mathbf{x}^s_M. \tag{4.1.17}$$

By differentiating 4.1.17 with respect to the unknown variables, we obtain:

$$L_U \mathbf{x}^s_U = -B \mathbf{x}^s_M. \tag{4.1.18}$$

We can show that the solution to this system exists and is unique if their exists at least one path composed of only nonzero-weight edges between each unknown node and a marked node (see appendix B.1).

The system of equation 4.1.18 is equivalent to the probabilistic Random Walks equation 4.1.7. This can be seen by noting that, if D has no zero entry – i.e. each pixel has

at least one edge with a non null weight – one can retrieve Π from equation 4.1.9, by normalizing A:

$$\Pi = D^{-1}A. \tag{4.1.19}$$

Thus, we can write

$$\begin{aligned} L &= D\left(I - \Pi\right), &(4.1.20)\\ &= DL_{rw}, &(4.1.21)\end{aligned}$$

where L_{rw}, defined as $L_{rw} = I - \Pi$, is called the *normalized Laplacian*. Decomposing L_{rw} into sub-blocks, we can write:

$$\begin{aligned} L_U &= D_U(I - \Pi_U), &(4.1.22)\\ B &= -D_U\Pi_{UM}, &(4.1.23)\end{aligned}$$

where D_U is the sub-matrix of D corresponding to unknown indices.

4.1.6 *Segmentation algorithm*

The Random Walks functional 4.1.13 provides a measure of the probabilistic segmentation $\mathbf{x}^s$, and the optimal segmentation is the one which minimizes it, i.e. which has the lower cost according to this measure. We have seen that minimizing the RW functional is equivalent to solving the initial RW system, which we recall here:

$$\forall v_i \in \mathcal{V}_U,\ x_i^s = \sum_{v_j \in \mathcal{N}_i} p_{ij} x_j^s.$$

It can be shown (see appendix B.2) that a solution to 4.1.18 is always a probability:

$$\begin{aligned} \forall i,\ \sum_s x_i^s = 1, &(4.1.24)\\ \forall i, \forall s,\ x_i^s \geq 0. &(4.1.25)\end{aligned}$$

Once one has inferred the probabilistic segmentation $\hat{\mathbf{x}}^s$ for each label s, the "hard" segmentation – i.e. the estimated label $\hat{l}\,(i)$ of each node v_i – is obtained via:

$$\hat{l}\,(i) = \arg\max_s \hat{x}_i^s. \tag{4.1.26}$$

4.1.7 *Related work*

The graph Laplacian is a fundamental tool for graph analysis, and is related to many other notions besides Random Walks, as we will see in the following.

SPECTRAL CLUSTERING In the machine learning community, one finds spectral clustering (Shi and Malik, 2000b; Ng et al., 2002; Meila and Shi, 2001), an extremely popular clustering technique. We refer the reader to Von Luxburg (2007) for a detailed overview of many aspects of spectral clustering.

There exists several versions of the spectral clustering technique, involving either the un-normalized Laplacian L, the normalized Laplacian (Ng et al., 2002) L_{rw}, or the symmetric normalized Laplacian (Shi and Malik, 2000a):

$$L_{sym} = D^{-1/2} L D^{-1/2}. \tag{4.1.27}$$

Assuming that the data consist of n points $x_1, \dots x_n$ in an arbitrary space, an affinity matrix S – also called similarity matrix – is built from some pairwise similarity function $S_{ij} = s\left(x_i, x_j\right) \geq 0$. A similarity graph $\mathcal{G}$ is constructed over the data points and their pair-wise similarities. The first k eigenvectors $u_1, \dots u_k$ of one of the Laplacians of $\mathcal{G}$ are computed and stored in a matrix $U \in \mathbb{R}^{n \times k}$ (in (Ng et al., 2002), the rows of U are normalized to norm 1). Then, denoting the points whose coordinate are the *rows* of U as $y_1, \dots, y_n$, clustering with the k-means algorithm (Hartigan and Wong, 1979) is performed on "abstract" points $\{y_i\}_i$.

NORMALIZED GRAPH CUTS Spectral clustering can be seen as an approximation of some graph-cut problem. Let us denote a partition of the graph with k subsets as $A_1, \dots, A_k$. With the notation: $W(A, B) = \sum_{i \in A, j \in B} \omega_{ij}$ and $\bar{A}$ as the complement of A, then the min-cut problem consists of minimizing the following quantity:

$$\operatorname{cut}(A_1, \dots, A_k) = \frac{1}{2} \sum_{i=1}^{k} W(A_i, \bar{A}_i). \tag{4.1.28}$$

As the solutions to the min-cut problem often tend to create "unbalanced" partitions, selecting excessively small subsets of nodes, some modifications of the objective were introduced; the best-known are the RatioCut (Hagen and Kahng, 1992) and the normalized cut Ncut (Shi and Malik, 2000b). Denoting the number of node in partition A as$|A|$ and the total weight of the edges in A as $\operatorname{vol}(A) = \sum_{i,j \in A} \omega_{ij}$, the definitions are:

$$\operatorname{RatioCut}(A_1, \dots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{|A_i|}, \tag{4.1.29}$$

$$\operatorname{Ncut}(A_1, \dots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{\operatorname{vol}(A_i)}. \tag{4.1.30}$$

However, unlike standard min-cut, both problems are NP-hard and can only be solved approximately.

It can be shown (Shi and Malik, 2000a; Hagen and Kahng, 1992) that these quantities can be expressed using the un-normalized graph Laplacian:

$$\begin{aligned} \text{RatioCut}\left(A_1, \ldots, A_k\right) &= \text{Tr}\left(H^\top L H\right), \\ \text{with} \quad H_{ij} &= \begin{cases} 1/\sqrt{|A_j|} & \text{if } v_i \in A_j \\ 0 & \text{otherwise} \end{cases}, \end{aligned} \tag{4.1.31}$$

and the symmetric normalized graph Laplacian:

$$\begin{aligned} \text{Ncut}\left(A_1, \ldots, A_k\right) &= \text{Tr}\left(H^\top L_{sym} H\right), \\ \text{with} \quad H_{ij} &= \begin{cases} 1/\sqrt{\text{vol}\left(A_j\right)} & \text{if } v_i \in A_j \\ 0 & \text{otherwise} \end{cases}. \end{aligned} \tag{4.1.32}$$

Thus, both problems can be reformulated as trace minimization problems under some constraints. It can be shown that spectral clustering using the un-normalized Laplacian L (respectively the symmetric normalized Laplacian L_{sym}) solves the relaxation of the RatioCut problem (respectively the Ncut problem (Shi and Malik, 2000a)) when the entries of H are allowed to take arbitrary real values. Please refer to the work of Von Luxburg (2007) for further details and proofs. Although the solutions of the relaxed problems via spectral clustering can be very different from the original problems, spectral clustering remains a popular approximation for its simplicity of use and the quality of its output in practice.

NCUT AND RANDOM WALKS A relationship exists between Ncut and Random Walks (Meila and Shi, 2001), which can be seen from the following result:

$$\text{Ncut}\left(A, \bar{A}\right) = P\left(\bar{A}|A\right) + P\left(A|\bar{A}\right), \tag{4.1.33}$$

where $P\left(A|B\right)$ is the probability that the random walker jumps from a node in subset B onto a node in subset A. Thus, one can say that minimizing Ncut amounts to finding a partition such that a random walker has the lowest probability of transitioning between parts.

COMMUTE TIME DISTANCE The commute time distance between two vertices $v_1$and v_2 is defined as the expected number of jumps of a random walker to travel from v_1 to v_2 and back. Among other interesting properties, the commute distance differs from the shortest path in that it considers all possible paths between the two nodes; as a result, two subsets separated by a partial gap – i.e. whose all but a few inter-connecting

edges have low weights – will be considered as "far apart" when using the commute distance, whereas they could be considered as "close-by" with the shortest path distance. The commute time distance can be computed using the generalized inverse of the un-normalized Laplacian L. Recalling the eigen-decomposition of the Laplacian $L = U\Lambda U^\top$, the generalized inverse of L is given by:

$$\begin{aligned} L^\dagger &= U\Lambda^\dagger U^\top, \\ \text{with } \Lambda^\dagger_{ii} &= \begin{cases} 1/\Lambda_{ii} & \text{if } \Lambda_{ii} > 0 \\ 0 & \text{otherwise} \end{cases}. \end{aligned} \tag{4.1.34}$$

Then, the commute time distance between nodes v_i and v_j (Saerens et al., 2004) denoted as c_{ij} , is given by:

$$c_{ij} = \text{vol}\,(V)\left(L^\dagger_{ii} - 2L^\dagger_{ij} + L^\dagger_{jj}\right). \tag{4.1.35}$$

As a consequence one can define an embedding of the data points $x_1, \ldots, x_n$, denoted as $z_1, \ldots, z_n$ and given by the rows of matrix $\left(\Lambda^\dagger\right)^{1/2} U$, such that the commute time distances between points x_i are converted to Euclidean distances between points z_i:

$$c_{ij} = \text{vol}\,(V)\left\|z_i - z_j\right\|^2. \tag{4.1.36}$$

This embedding is similar to that of the spectral clustering (where the data points are mapped to the rows of U), although it does not yield identical results in clustering.

LAPLACIAN EIGENMAPS The Laplacian eigenmaps algorithm (Belkin and Niyogi, 2001) is used to perform dimensionality reduction of low-dimensional manifolds embedded in a high-dimensional space. The embedding of the sample points $x_1, \ldots, x_n$ is given by the rows of U which contains the k eigenvectors corresponding to the first k smallest eigenvalues for L_{sym} (same embedding as the spectral clustering of (Shi and Malik, 2000a)). It can be seen (Ham et al., 2004) that the Laplacian eigenmap algorithm is equivalent to kernel Principal Component Analysis (PCA) using kernel $L^\dagger$.

4.2 RANDOM WALKS WITH PRIOR KNOWLEDGE

One can say that the standard formulation of the Random Walks segmentation method relies on prior knowledge in the form of pre-segmented nodes: the seeds. In general seeds are provided by the user, who manually segments a few pixels of each class. While this method is convenient and much more efficient than segmenting the whole image manually, it is still impractical for large datasets. For this reason, one can be interested in completely automatizing the segmentation process by using different, unsupervised types of prior knowledge such as statistical intensity of shape models.

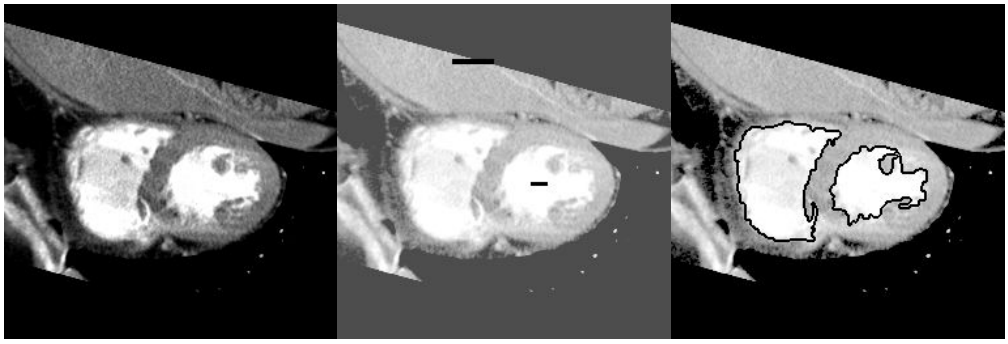

Figure 4.2.1 – Appearance model for RW in heart segmentation: (left) original image; (center) seeds as black lines; (right) segmented image. The appearance model reduces the amount of required user interaction (Grady, 2005)

Moreover, even in the case of manual seeding, incorporating additional information could improve the segmentation results and thus also decrease the user time.

4.2.1 *Random Walks segmentation with appearance prior*

Prior knowledge to the Random Walk formulation was introduced by Grady (2005), providing an estimate of the probability distribution of the gray-level intensity for each label. An an example of such technique can be found in figure 4.2.1, which allows using very few seeds to separate organs from the background, using the differences in intensity distributions.

Let us introduce $a_s(\cdot)$ as the probability density distribution for label s. Then, denoting I_i the intensity at pixel i, the Bayes' rule states that the probability of assignment to s for pixel i is:

$$x_i^s = \frac{a_s(I_i)}{\sum_t a_t(I_i)}. \tag{4.2.1}$$

Denoting the vector containing the $a_s(I_i)$ for each pixel i as $\mathbf{a}^s = [a_s(I_1), \ldots, a_s(I_N)]$ and $A^s = \operatorname{diag}(\mathbf{a}^s)$, the previous equality can be rewritten as

$$\left(\sum_t A^t\right)\mathbf{x}^s = \mathbf{a}^s. \tag{4.2.2}$$

The solution to this system is the minimum of the *aspatial functional*:

$$\mathbf{x}^{s\top}\left(\sum_{t\neq s} A^t\right)\mathbf{x}^s + (\mathbf{1}_N - \mathbf{x}^s)^\top A^s(\mathbf{1}_N - \mathbf{x}^s). \tag{4.2.3}$$

This prior-appearance functional is combined with the Random Walks functional 4.1.13 with the introduction of a free balancing parameter w_{app}:

$$\begin{aligned} E^s_{\text{RWapp}}(x^s) &= \mathbf{x}^{s\top} L \mathbf{x}^s \\ &+ w_{\text{app}} \left[\mathbf{x}^{s\top} \left(\sum_{t \neq s} A^t \right) \mathbf{x}^s + (\mathbf{1}_N - \mathbf{x}^s)^\top A^s (\mathbf{1}_N - \mathbf{x}^s) \right]. \end{aligned} \tag{4.2.4}$$

Intuitively, minimizing this functional will yield a segmentation influenced by both the Random Walk principle – segmentation along boundaries – and by the appearance prior – privileging label assignments that are consistent with intensity distributions. The optimal segmentation minimizing the functional 4.2.4 is obtained through solving:

$$\left(L + w_{\text{app}} \sum_t A^t \right) \mathbf{x}^s = w_{\text{app}} \mathbf{a}^s. \tag{4.2.5}$$

Since L is positive semi-definite, system of equations 4.2.5 can be solved even when all entries of x^s are unknown (no seeds). However, as noted by Grady (2005), it is still possible – and useful – to use seeds in combination with a prior to obtain more robust segmentations. In the case of muscle segmentation however, the intensity distributions of the labels (the muscles) are extremely similar – except for the background – and thus would make a very weak prior if used alone.

4.2.2 *Probabilistic framework*[3]

In this section, we show that the intensity prior of previous section 4.2.1 is consistent with the probabilistic framework of the Random Walks method, and allows incorporating prior knowledge of shapes as well. As we saw previously in the case of standard Random Walks, prior information is incorporated into the seeds, in the form of pre-segmented nodes. With the probabilistic notations, this is equivalent to setting the value of the assignment probability for marked nodes to either 0 or 1 – i.e. enforcing the certainty that each one of these pixels belongs to one chosen label.

At present, we release these constraints by providing no seed, but add an additional layer of nodes to the graph $\mathcal{G}$ with a one-to-one correspondence with the original layer of nodes (cf. figure 4.2.2). The nodes in this second layer $\mathcal{V}_0$ are called the *prior nodes*, while the nodes in the original layer $\mathcal{V}$ are the *primary nodes*. We set one *directed* edge from each primary node to its corresponding prior node. We denote the transition probability from primary node v_i to prior node v_{0i} as λ_i, and the assignment

3 Thanks to Danny Goodman for his help in this section.

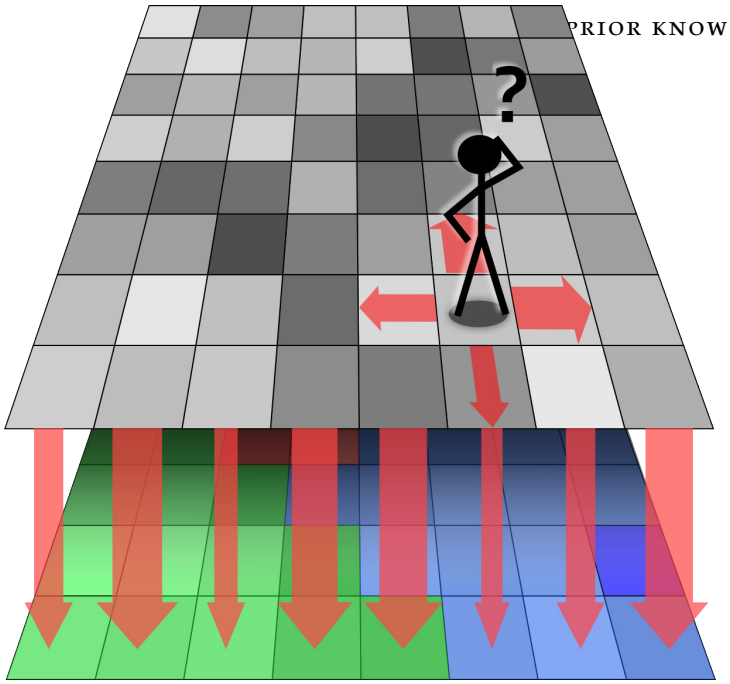

Figure 4.2.2 – Prior information is introduced to the Random Walks framework by adding a second layer of nodes (the prior nodes); each of these prior nodes has a known assignment probability to each label (here, each color on the second layer represents the most probable prior label). The transition probabilities (figured as vertical arrows) from the first layer to the second layer can vary along the pixels.

probability to label s of prior node v_{0i} as x^s_{0i}; no edge is set between the prior nodes. Then, the Random Walks node-wise probability (for primary nodes) becomes:

$$\forall v_i \in \mathcal{V},\ x^s_i = (1-\lambda_i) \sum_{v_j \in \mathcal{N}_i} p_{ij} x^s_j + \lambda_i x^s_{0i}, \tag{4.2.6}$$

where the probability p_{ij} of transition from primary node v_i to primary node v_j in the original RW equation 4.1.3 is multiplied by the probability to *not* walk onto the prior node v_{0i}: $(1-\lambda_i)$.

Denoting the diagonal transition matrix from primary nodes to prior nodes as$\Lambda = [\lambda_1, \ldots, \lambda_N]^T$ and the transition matrix between primary nodes as Π, we reformulate the Random Walks equation 4.2.6 in matrix form

$$\mathbf{x}^s = (I-\Lambda)\,\Pi\mathbf{x}^s + \Lambda\mathbf{x}^s_0 \tag{4.2.7}$$

$$[(I-\Lambda)\,(I-\Pi_{\mathcal{V}}) + \Lambda]\,\mathbf{x}^s = \Lambda\mathbf{x}^s_0. \tag{4.2.8}$$

We recall that D is the diagonal matrix such that $D_{ii} = \sum_j w_{ij}$, and that the combinatorial Laplacian of the primary graph is given by $L = D(I - \Pi)$. Then the new Random Walks equation is equivalent to

$$[(I - \Lambda) L + \Lambda D] \mathbf{x}^s = \Lambda D \mathbf{x}_0^s, \tag{4.2.9}$$

which can be solved if the transition probabilities to prior nodes are non-zero, ensuring that $(I - \Lambda) L + \Lambda D$ is positive definite.We can rewrite the previous as

$$(L + \Delta) \mathbf{x}^s = \Delta \mathbf{x}_0^s, \tag{4.2.10}$$

with the diagonal matrix

$$\Omega = (I - \Lambda)^{-1} \Lambda D, \tag{4.2.11}$$

which excludes setting $\lambda_i = 1$ without loss of generality, since such setting would make the problem trivial by considering only the prior model and ignoring the Laplacian.

The solution to the last equation 4.2.10 provides the minimum of the following objective:

$$\begin{aligned} E^s_{\text{RW+prior}}(\mathbf{x}^s) &= \mathbf{x}^{s\top} L \mathbf{x}^s + (\mathbf{x}^s - \mathbf{x}_0^s)^\top \Omega (\mathbf{x}^s - \mathbf{x}_0^s), & (4.2.12)\\ &= \mathbf{x}^{s\top} L \mathbf{x}^s + \|\mathbf{x}^s - \mathbf{x}_0^s\|^2_\Omega, & (4.2.13) \end{aligned}$$

where $\|.\|_\Omega$ is a weighted norm by Ω. Intuitively, the optimal segmentation minimizing this functional is influenced by both the Random Walk principle – segmentation along boundaries – and by the prior model – privileging label assignments consistent with the prior probabilities. Any solution to this problem is a probability distribution if $\{x^s_{0i}\}_s$ is a probability distribution (please refer to appendix B.3):

$$\sum_s x_0^s = \mathbf{1} \Rightarrow \sum_s x^s = \mathbf{1}. \tag{4.2.14}$$

Going back to the appearance prior model of section 4.2.1, we set the prior probabilities as:

$$\mathbf{x}_0^s = \left(\sum_t A^t \right)^{-1} \mathbf{a}^s, \tag{4.2.15}$$

and Ω as:

$$\Omega = w_{\text{app}} \left(\sum_t A^t \right), \tag{4.2.16}$$

retrieving the system for appearance prior 4.2.5.

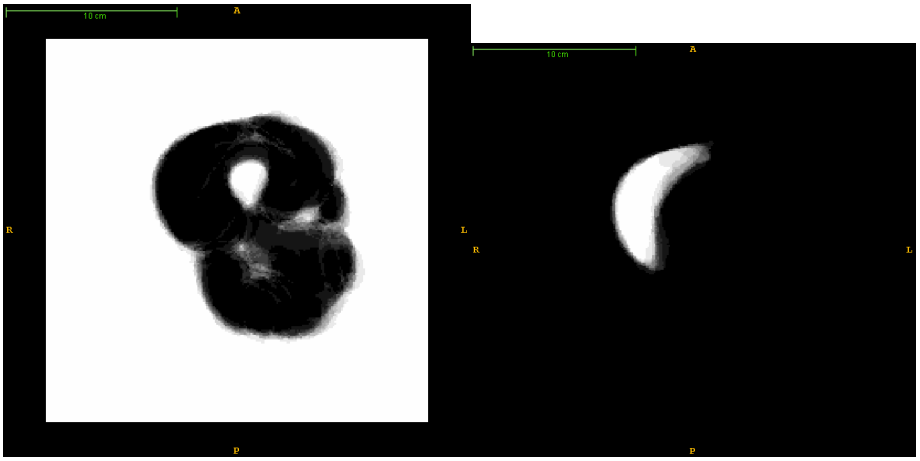

Figure 4.3.1 – Label assignment probability of prior model $\mathbf{x}_0^s$, for two different labels s : (left) background; (right) *vastus lateralis*.

4.3 RANDOM WALKS SEGMENTATION WITH SHAPE PRIOR

4.3.1 *Constant prior weight*

In this section we propose a simple shape prior model for the Random Walks based on the general formulation 4.2.12. Suppose that we have an estimate for x_0^s, based on learning a pixel-wise shape over previous segmentations in a training set $\mathcal{T}$ (see figure 4.3.1). We set the transition probabilities from the primary nodes to the prior nodes in Λ such that Ω is the constant diagonal matrix:

$$\Omega \quad = \quad w_{\text{shape}} I, \tag{4.3.1}$$

with $w_{\text{shape}} > 0$. Since $\Omega = (I - \Lambda)^{-1} \Lambda D$, this is achieved by setting the transition probability λ_i to the prior nodes in the following manner:

$$\forall v_i \in \mathcal{V},\ \lambda_i = \frac{1}{\sum_j \omega_{ij} / w_{\text{shape}} + 1}. \tag{4.3.2}$$

With this choice of Ω, each pixel is biased towards its prior assignment probabilities depending on the relative values of the w_{ij} and w_{shape}: if $\omega_{ij} \ll w_{\text{shape}}$, node v_i is influenced more by the prior than by its neighborhood; conversely, if $\omega_{ij} \gg w_{\text{shape}}$, then node v_i is more strongly influenced by the label assignment of its neighbors than by the prior probabilities. The corresponding functional is:

$$E^s_{RW\text{const}}(\mathbf{x}^s) = \mathbf{x}^{s\top} L \mathbf{x}^s + w_{\text{shape}} \left\| \mathbf{x}^s - \mathbf{x}_0^s \right\|^2. \tag{4.3.3}$$

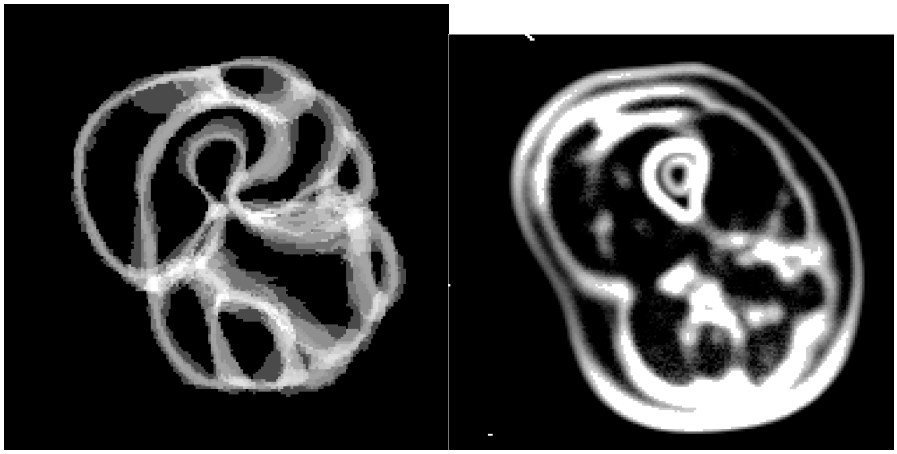

Figure 4.3.2 – Weighting schemes, displaying the diagonal of Ω as an image: (left) entropy weighting; (right) confidence map.

This model simply penalizes the deviation of vector x^s from x_0^s with a uniform weighting scheme.

The prior assignment probabilities can be obtained by computing the empirical mean of the assignment probabilities over the training set $\mathcal{T}$. Since the training $\mathcal{T}$ is composed of manually segmented images, the assignment of each pixel is known and the corresponding assignment probability is either 0 or 1. Referring to $l_k(i)$ as the assigned label of pixel i in training image k, the binary assignment probability is defined by:

$$z_{k\,i}^s = \begin{cases} 1 & \text{if } l_k(i) = s, \\ 0 & \text{if } l_k(i) \neq s, \end{cases} \tag{4.3.4}$$

and the corresponding vector is denoted as $\mathbf{z}_k^s$. Then, $\mathbf{x}_0^s$ is computed as:

$$\mathbf{x}_0^s = \frac{1}{|\mathcal{T}|} \sum_k \mathbf{z}_k^s. \tag{4.3.5}$$

4.3.2 *Prior-weighted models*

ENTROPY MODEL The previous shape model weights each pixel by the same amount w_{shape} towards the average probability. However, if we own a confidence measure of the quality of this prior model, it could be useful to vary the weights in order to constrain the pixels according to the degree of confidence. For instance, the entropy

$$e(i) = -\sum_s x_{0\,i}^s \log\left(x_{0\,i}^s\right) \tag{4.3.6}$$

is a measure of how certain the prior model is of the pixel assignment. Indeed, if pixel i was consistently assigned to label s in the training set, then $e(i) \approx 0$. Conversely, if pixel i has similar assignment probabilities for all the labels, $e(i) \approx \log(|\mathcal{S}|)$ (see figure 4.3.2). Hence, a possible weighting scheme is given by:

$$\Omega_{ii} = w_{\text{shape}} \left(1 - \frac{e(i)}{\log |\mathcal{S}|}\right). \tag{4.3.7}$$

GAUSSIAN MODEL We observe that the term $\|\mathbf{x}^s - \mathbf{x}_0^s\|^2$ in the constant shape prior functional can be interpreted as the typical log-likelihood functional for independent Gaussian variables, with mean $\mathbf{x}_0^s$ and variance 1:

$$\|\mathbf{x}^s - \mathbf{x}_0^s\|_2^2 = -\log \exp\left(-\|\mathbf{x}^s - \mathbf{x}_0^s\|^2\right). \tag{4.3.8}$$

Pursuing this Gaussian analogy, we may use the empirical variance as a measure of confidence:

$$\Omega_{ii}^s = w_{\text{shape}} \frac{1}{\sigma(i)}. \tag{4.3.9}$$

In the previous, we may use the empirical estimate of the variance over the training set:

$$\hat{\sigma}^2(i) = \frac{1}{|\mathcal{T}|\,|\mathcal{S}|} \sum_s \sum_k \left(z_{k\,i}^s - x_{0\,i}^s\right)^2. \tag{4.3.10}$$

Since $x(i,s)$ is a probability and since the variables in x^s are necessarily correlated, such Gaussian modeling can only be a rough approximation but it has the advantage of simplicity. In practice, due to the small number of training examples, it is necessary to regularize the variance term. Indeed, some pixels have a variance that is equal to zero when all training examples agree on this pixel's label. An empirical regularization of the variance can be obtained with the following formula:

$$\Omega_{ii}^s = w_{\text{shape}} \frac{1}{\alpha\sigma^2(i) + (1-\alpha)}, \tag{4.3.11}$$

with $\alpha \in]0,1[$.

4.3.3 *Confidence map*

All previous shape models balance the influence of the prior model according to information extracted from the training data. We may also find interesting to reduce the influence of the model according to the strength of the contours *in the test image*, rather than from the training set: the stronger the contours, the least we should rely on the model (see figure 4.3.2). Assume we possess such a "confidence map" c, with values

close to 0 on strong contours, and values close to 1 in homogeneous regions, we can set:

$$\Omega^s = w_{\text{shape}} \text{diag}\,(c)\,. \tag{4.3.12}$$

In the following, we propose a simple formula to compute the confidence map, but many other methods would likely be as effective. Denoting the local variance at pixel i computed on a patch with radius r as $\sigma_r^2\,(i)$, we propose the following formula:

$$c_i = \exp\left(-k_v \sigma_r^2\,(i)\right), \tag{4.3.13}$$

where k_v is a free parameter .

4.3.4 *Experimental validation*

DATASETS All our experiments were performed on datasets composed of 3D volumes of the right thigh of healthy subjects, covering a wide range of morphologies (males and females, ages range: 26 to 60), acquired with a 3T Siemens scanner and using 3pt Dixon sequence (TR=10ms, TE1=2.75 ms TE2=3.95 ms TE3=5.15 ms, rf. flip angle =3°; we used the out-of-phase image), with the an average image size of: $220 \times 220 \times 64$px. Four clinically relevant muscles (the quadriceps: vastus lateralis, vastus intermedius, vastus medialis and rectus femoris) were segmented in 30 volumes; in addition, among these 30 volumes, 13 of which had all 13 thigh muscles segmented.

REGISTRATION For a given test volume, all other volumes served as a training set and were non-rigidly registered onto the test volume. We chose non-rigid registration in order to have the least possible variations of the outer-boundaries of the muscle tissues (the muscle/fat boundaries). The objective is to rely on the RW algorithm only to segment the more difficult inter-muscle boundaries.

As mentionned in the conclusion of the previous chapter, registration of the outer-boundaries of the muscle tissues is much easier than registration of the inter-muscle boudaries. Indeed, the latter are often weak or confused with spurious ones, such that trying to register them may lead to larger errors if the registration process is not-rigid enough. For this reason, we generated binary masks of the muscle tissues from the water-map image (see appendix A) and used them to compute the deformation fields. With binary masks, the inter-muscle boundaries are completely ignored by the registration software, and thus only deformed accordingly to the outer-boundary matching. The deformation fields were then applied to the corresponding segmentation atlases in order to generate the average segmentation $\mathbf{x}_0$.

Registration was achieved using the Drop software (Glocker et al., 2008). This methods uses a multi-scale discrete modeling approach relying on a Markov random field objective function minimized through efficient linear programming. This method can

accept any similarity measure and we chose the default one (sum of absolute differences) to register the binary masks. The software provided a smooth deformation field which we then used to deform the reference atlas.

VALIDATION To evaluate the segmentation generated by our methods, we computed Dice coefficients. The expression of the Dice coefficient is: $D = 2\,|T \cap R| \,/\, (|T| + |R|)$, where T and R are the pixel sets of a given muscle in the inferred segmentation and the reference manual segmentation respectively. Let us denote the binary vector containing the manual segmentation for label s as $\mathbf{z}^s$, and the binarized vector containing the inferred segmentation for label s as $\overline{\mathbf{x}}^s$, that is,

$$\overline{\mathbf{x}}^s(i) = \begin{cases} 1 & \text{if } s = \arg\max_t \mathbf{x}^t(i), \\ 0 & \text{otherwise.} \end{cases} \tag{4.3.14}$$

Then, the Dice coefficient for label s of the inferred segmentation $\mathbf{x}$ is computed as

$$D_s = \frac{2\mathbf{z}^{s\top}\overline{\mathbf{x}}^s}{\mathbf{1}^\top(\mathbf{z}^s + \overline{\mathbf{x}}^s)}. \tag{4.3.15}$$

Thus, if the inferred segmentation is identical to the manual segmentation for a given label, it will have Dice coefficients equal to 1 for all muscles; conversely, if it is entirely different from the manual segmentation, its Dice coefficients will be 0.

Results are presented as boxplots[4], with the Dice coefficients for all the segmented muscles combined into a single box – *excluding* the background. Figs. 4.3.3 and 4.3.4show some examples of segmentations obtained with our method (constant weighting scheme, 13-muscle dataset). Large muscles tend to be better segmented that small ones. This is related to the average segmentation $\mathbf{x}_0$: since the average segmentation and the true segmentation of the target image do not correspond exactly, a larger proportion of the small muscles are incorrectly segmented by $\mathbf{x}_0$, as compared to the larger muscles. As a result, the assignment probabilities computed by the RW algorithm are generally lower than with large muscles, thus leading to more segmentation errors. This is also true when a muscle has a rarely seen location or shape : large segmentation errors with $\mathbf{x}_0$ may lead to large errors in the resulting segmentation. We conclude that this algorithm only works well for average anatomies. Improvements of this method will necessitate a more flexible model, capable of adapting to a larger range of anatomies.

4 Box-plot presentation: the boxes contain the middle 50% of the data and the median value, and the extremities of the lines indicate the min and max values, excluding the outliers (for more details, see the documentation of Matplotlib).

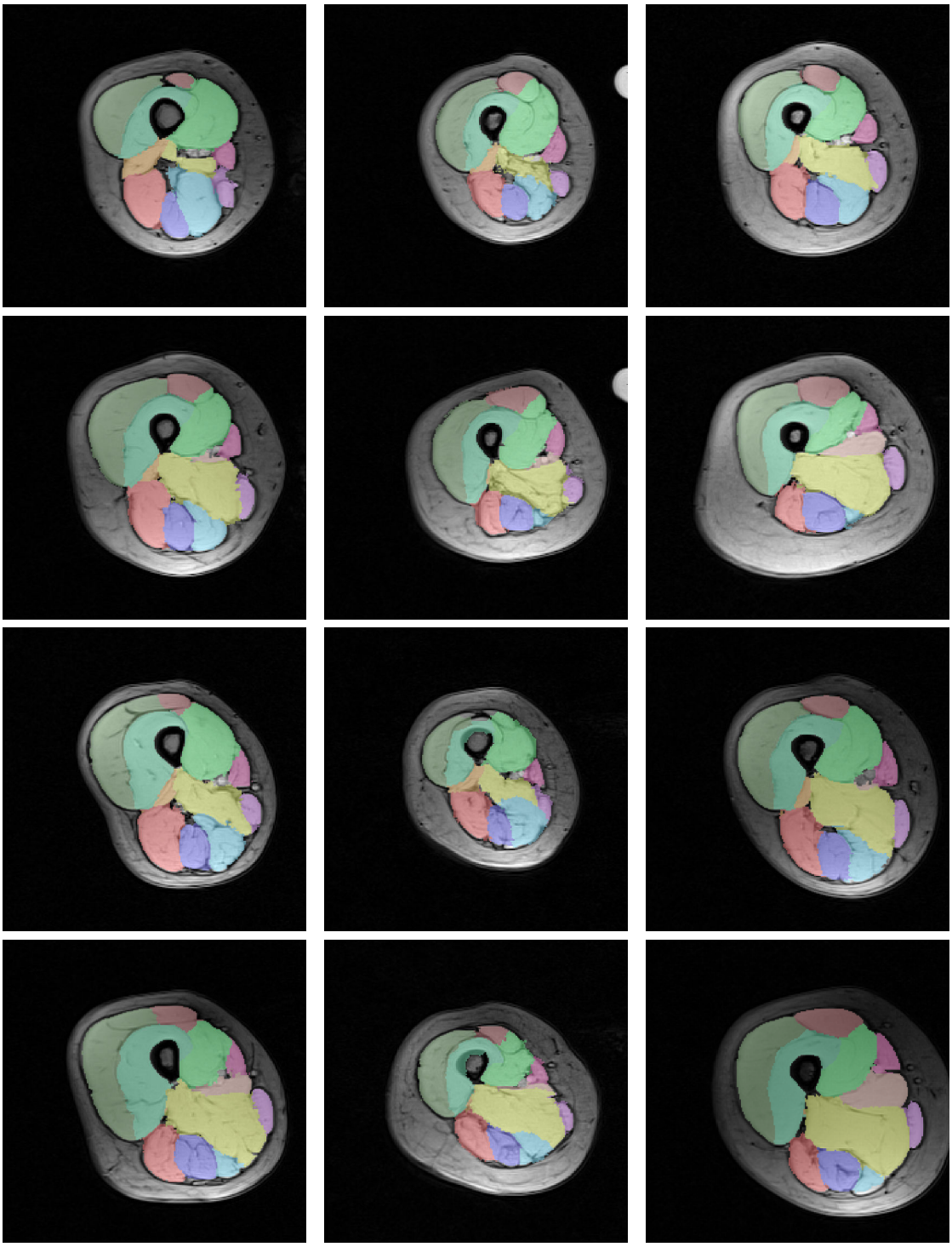

Figure 4.3.3 – Examples of segmentations obtained with the RW algorithm with shape priors (constant weighting scheme).

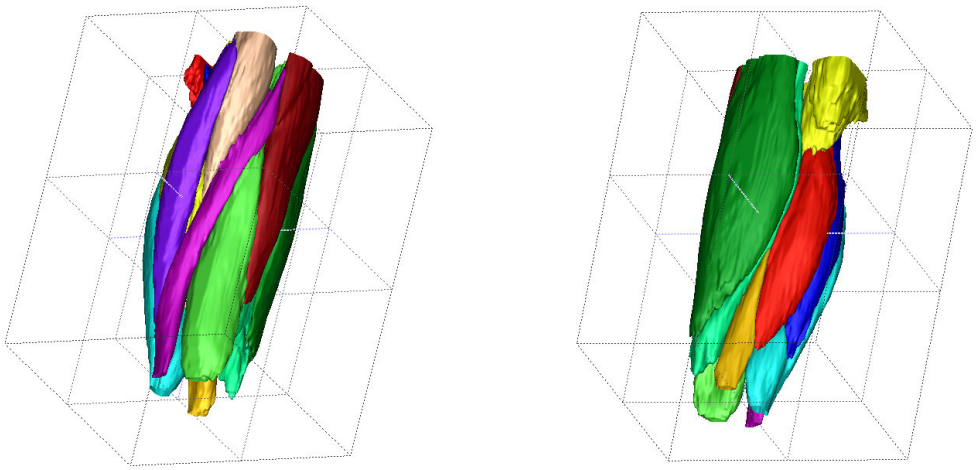

Figure 4.3.4 – Two 3D views of a segmentation (right thigh) obtained with our method.

VARIOUS WEIGHTING SCHEMES We tested different weighting schemes based on statistics on the training database: entropy weighting and Gaussian weighting and compared them to the constant weighting scheme. In addition, we also tested the confidence map weighting scheme. Dice coefficient for the different approaches were reported in figure 4.3.5 and show that for all tested parameters w_{shape}, neither weighting scheme improved on the constant weighting; the variance weighting even seems to degrade the performance. The reason for the lack of improvement is possibly due to the limited validity of the statistics on such a small number of training example (29 volumes, the 30th is the one being segmented). However, although no statistically significant differences in terms of Dice coefficients were found, one can observe on the segmented data some visual differences depending on the weighting scheme used. For instance, figure 4.3.6 shows some of the effects of the confidence map scheme on the segmentation results; here, the result is less influenced by the prior shape than with the constant prior, which improves the segmentation quality around the bone; however, the gain in accuracy in one place is lost in another place where the contours are missing, due to the increased freedom of the model. As often, it appears that the compromise between segmenting the contours accurately and keeping a valid shape is hard to find.

Since all hyper-parameters were selected by hand – in particular w_{shape} – we expect that an automatized parameter selection would yield more optimal values and maybe find significant differences between the weighting schemes. In the next chapter of this report, we propose such a method for selecting parameters.

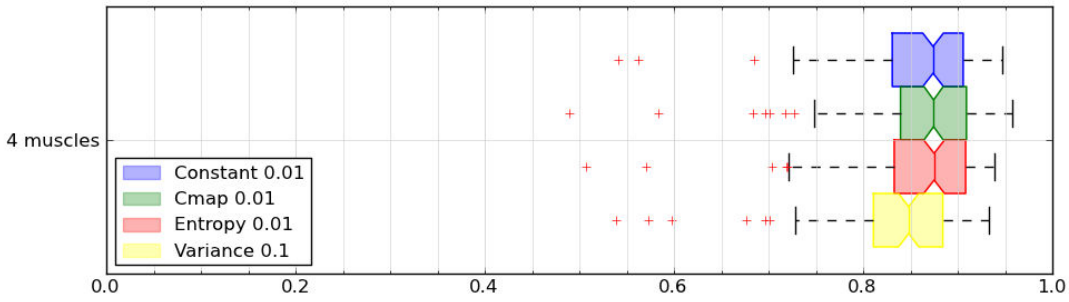

Figure 4.3.5 – Comparing performances for various weighting schemes; the boxplots are generated from Dice coefficients for all labels (4 muscle database). "Cmap" designates the confidence map weighting scheme.

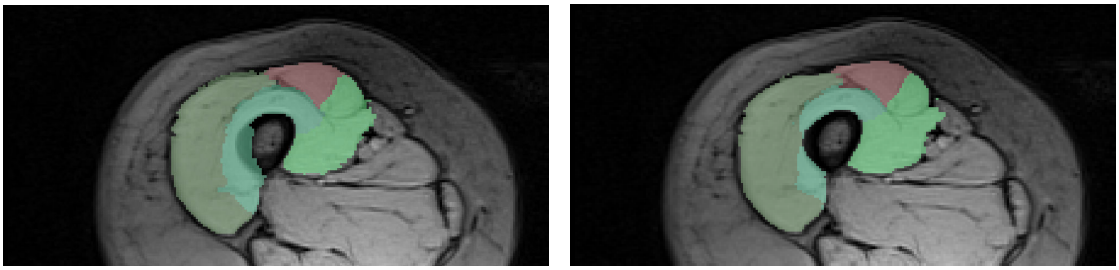

Figure 4.3.6 – Comparison of segmentations using the constant weighting scheme (left) and the confidence map weighting scheme (right).

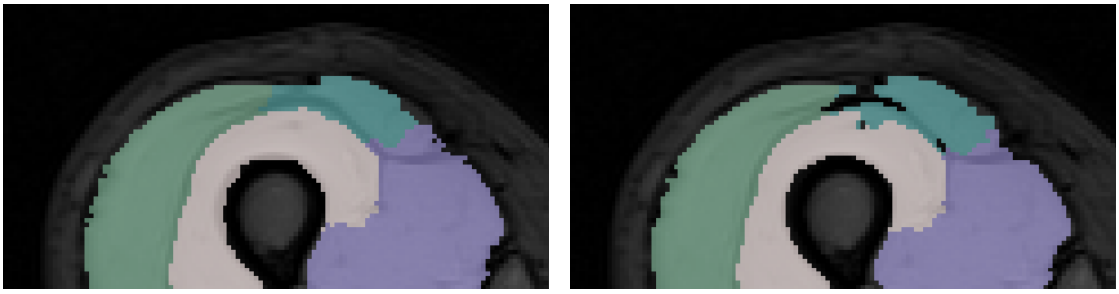

Figure 4.3.7 – Local segmentation differences due to adding an appearance prior term

USING THE APPEARANCE PRIOR As mentioned previously, it is possible to combine several prior terms to benefit from complementary knowledge sources. For instance, let us consider the appearance prior seen in section 4.2.4; this prior term is not usable in itself in the context of muscle separation, since all muscles have the same appearance properties; however, when used in combination with a shape prior term, it could help improving the separation between muscles and fat tissues.

In figure 4.3.7, we show identical cross-sections where the left one has been segmented only with the constant shape prior (that is: equation 4.3.3, using $w_{\text{shape}} = 1.10^{-2}$), whereas the right one has been segmented with the same prior term plus an appearance prior (equation 4.2.5, using $w_{\text{app}} = 1.10^{-2}$). One can see the appearance prior has an obvious influence over the muscle/background separation in several places. However, quantitative results – shown in figure 4.3.8 – show that the appearance prior does not improve the segmentation on our dataset. This is due to the fact that, in our test images (see appendix section A.2.2), the average intensity of the fat tissues is close to the average intensity of muscle tissues; moreover, there are notable local variations of the average intensity due to magnetic field inhomogeneities – a notorious artifact in MRI. Although it appears that such an appearance prior is inefficient on our database, the evidence of a discriminative effect shown in figure 4.3.7 leads us to believe it could improve the segmentation results on a different dataset where fat and muscle tissues are more clearly separated (for instance when using T_1-weighted images).

COMPARISON WITH PREVIOUS WORK We compared the presented method with related methods (see figures 4.3.11 and 4.3.9). For reasons explained in the following, the tests could not be performed on the exact same datasets, but only on overlapping datasets with different number of samples and segmented muscles. Thus, some of these comparisons should be taken cautiously. All tests were performed by the authors of the corresponding methods.

- The method of Gilles and Pai (2008), briefly presented in section 2.1.2. We recall this method belongs to the deformable model family, using a single reference

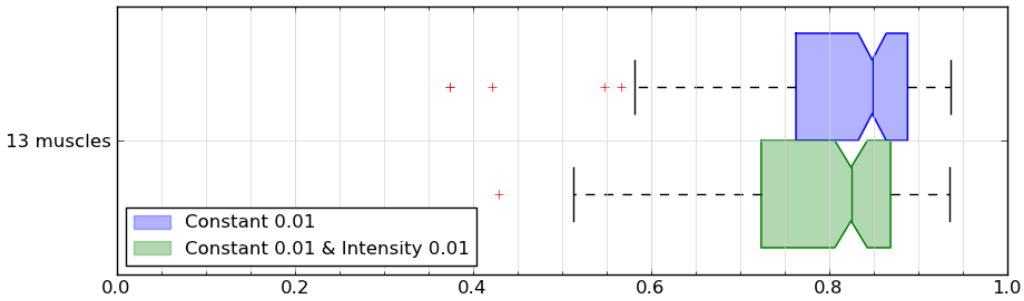

Figure 4.3.8 – Comparing performances with and without the appearance prior; the boxplots are generated from Dice coefficients for all labels (13 muscle database).

shape model as initial state and prior term. *Dataset*: 16 volumes extracted from our 30-muscle dataset, among which one was fully segmented for providing the prior shape; the 15 other volumes were then segmented automatically[5]; we computed the Dice coefficients only on the 4 muscles for which we had the manual segmentations.

- The method of Andrews et al. (2011a), briefly presented in section 2.2.1. This method is a region-based continuous method, with a shape prior derived from PCA performed on training samples projected into a Isometric Log-Ratio space. *Dataset used*: our database of 13 volumes with 13 segmented muscles. The images and corresponding segmentations have been cropped and aligned based on the location of the bone, as per the method described in the original paper[6]. On this dataset, due to not having a clear difference between the intensity of the muscles and that of the fat tissues, the method did not perform as well as in the published article where the authors present an average global Dice coefficient of 0.92.

- A method called "Registration", which consists of segmenting the volumes using the average segmentation $\mathbf{x}_0$. This amounts to considering only the performance of the non-rigid registration stage that is applied before constructing the prior model and segmenting the volumes with the RW method. This method can be seen as a multi-atlas segmentation method with majority label voting.

- Our own method for automatic generation of seeds (see chapter 3). *Dataset used*: our 15-muscle database. Since this method requires only one reference segmentation for providing a shape prior, many different tests could be performed in a cross-validation approach (with 13 volumes, this makes $13 \times 12 = 156$ tests).

- Our own Random Walks with shape prior method. *Dataset used*: either our 13-muscle or our 4-muscle database.

5 I would like to thank B. Gilles for performing these experiments
6 I would like to thank S. Andrews for performing these experiments

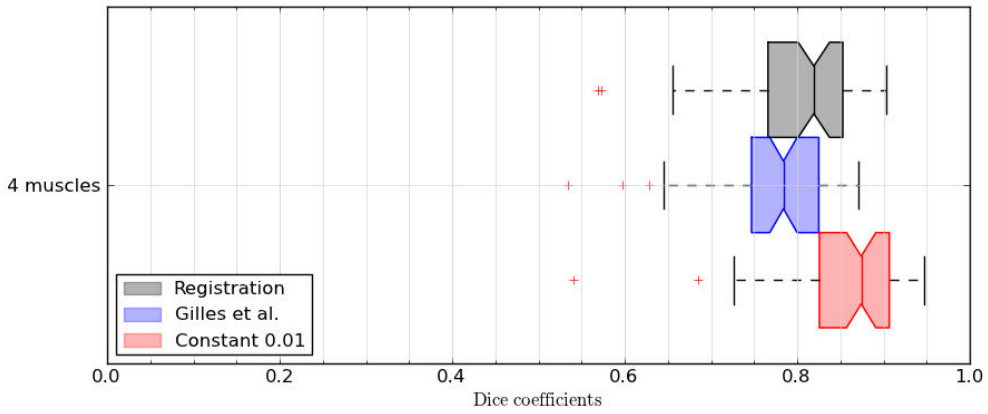

Figure 4.3.9 – Comparison of our method (bottom box) with the "registration" method (top box) and the method from (Gilles and Pai, 2008) (middle box). *Dataset*: subset of the 4-muscle dataset with only 15 test volumes.

Our first comparison concerns the method of Gilles and Pai (2008). We underline the fact that very few volumes of our dataset were segmented using this method, and thus the quantitative comparison with our own method has only limited validity. However some qualitative comments can be made about these results and the cross sections of segmentations obtained with this method (see figure 4.3.10). First, we notice that this method always produces valid shapes, that is, shapes that are realistic and could be actual segmentations. However, due to depending too much on one reference segmentation and also to the local minimization procedure, this method sometimes fails to segment large parts of the volumes even though the contours are well defined, as showed in the figure 4.3.10. These large errors explain the lower figures obtained here.

Second, we compare our method with the method of Andrews et al. (2011a). Here, the obtained Dices on our dataset with the latter method are abnormally low – as compared to the figures produced in the original paper. This can be explained by the fact that the type of contrast of our volumes is ill-adapted to this method, for it relies heavily on a the separation of muscle and fat tissues in a preliminary stage; however, our volumes do not show a clear visual difference between the two types of tissues, as the contrast parameters were rather selected for a good visualization of the contours.

Finally, we compare our method with the automatic seed placement method of chapter 3. As seen in figure 4.3.11, it appears that the autoseed method slightly outperforms[7] our current method on the tested dataset (all 13 muscle, 13 samples). It seems that this shape prior using the average segmentation is too constraining and does not

7 unlike what we published in (Baudin et al., 2012a). This is due to using a slightly different method here, where the labels are *not* weighted differenly, for remaining true to the probabilistic framework presented in section 4.2.2.

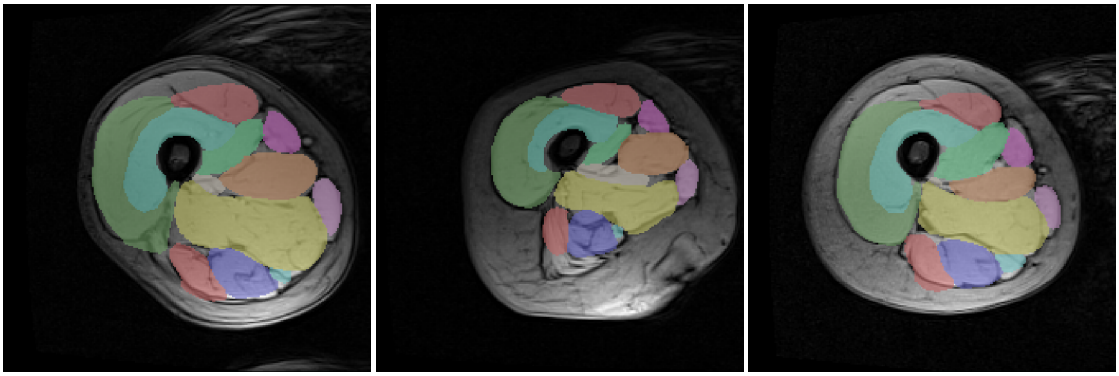

Figure 4.3.10 – Cross-sections of segmentations obtained with the method from (Gilles and Pai, 2008). This method always produces realistic segmentations, but sometimes fails to detect the contours.

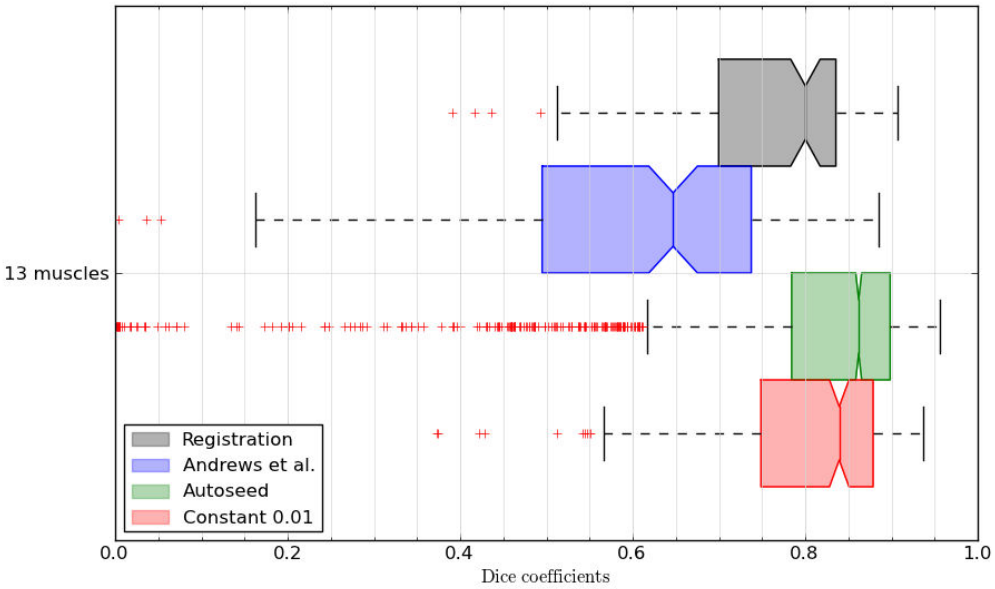

Figure 4.3.11 – Comparison of our method (bottom box) with the "registration" method (top box), the method from (Andrews et al., 2011a) (second box) and our automatic seeding method (third box). *Dataset*: 13-muscle dataset with 13 samples.

allow enough variability to fit the most different shapes. On the other hand, automatic seed placement is less constraining due to having no seeds close to the boundary of the muscles. However, when this latter method fails to label the seed correctly, errors can be extremely large as can be seen from the very low Dice coefficients in figure 4.3.11

PER-SLICE RESULTS In practice, certain analyzes of the muscles properties do not require to segment a muscle entirely. In figure 4.3.12 we show that the median Dice coefficient obtained with our method (using constant weighting, as it is the simplest method and provides results that are equivalent to or better than the other weighting schemes) is much greater in the middle slices than in the upper and lower slices. This is due to greater variability of the muscles shapes at their extremities, and to the fact that there are many more missing contours in those places (especially in the upper part of the thigh, where it is extremely difficult to determine the boundaries). These results show that the obtained segmentations are the most accurate in the places most useful for studying the muscles.

4.4 PCA MODEL

In this section, we propose a segmentation method based on the Random Walks algorithm, in which shape deformation is constrained to remain close to a Principal Component Analysis (PCA) shape space constructed over training examples. Using the PCA allows us to model complex non-rigid shape variations relying on a few eigen-modes. Such formulation does not fit in the probabilistic framework presented in section 4.2.2, and only yields an approximate solution.

4.4.1 *Additional notations*

Since minimizing 4.1.13 is an independent process for each label, the whole RW process can be equivalently synthesized in one equation. We denote the stacked up label-wise segmentations as $\mathbf{x}^\top = \left[\mathbf{x}^{1\top}|\mathbf{x}^{2\top}|\dots|\mathbf{x}^{K\top}\right]$, such that $\mathbf{x} \in \mathbb{R}^{KN\times 1}$ and the extended Laplacian matrix as:

$$\widetilde{L} = \begin{bmatrix} L & \cdots & 0 \\ \vdots & L & \vdots \\ 0 & \cdots & L \end{bmatrix} \in \mathbb{R}^{KN\times KN}. \tag{4.4.1}$$

Matrix $\tilde{L}$ is block-diagonal with K identical blocks, where$K$ is the number of labels. In this notations, the RW objective becomes:

$$E_{\text{RW}}(\mathbf{x}) = \mathbf{x}^T\widetilde{L}\mathbf{x}, \tag{4.4.2}$$

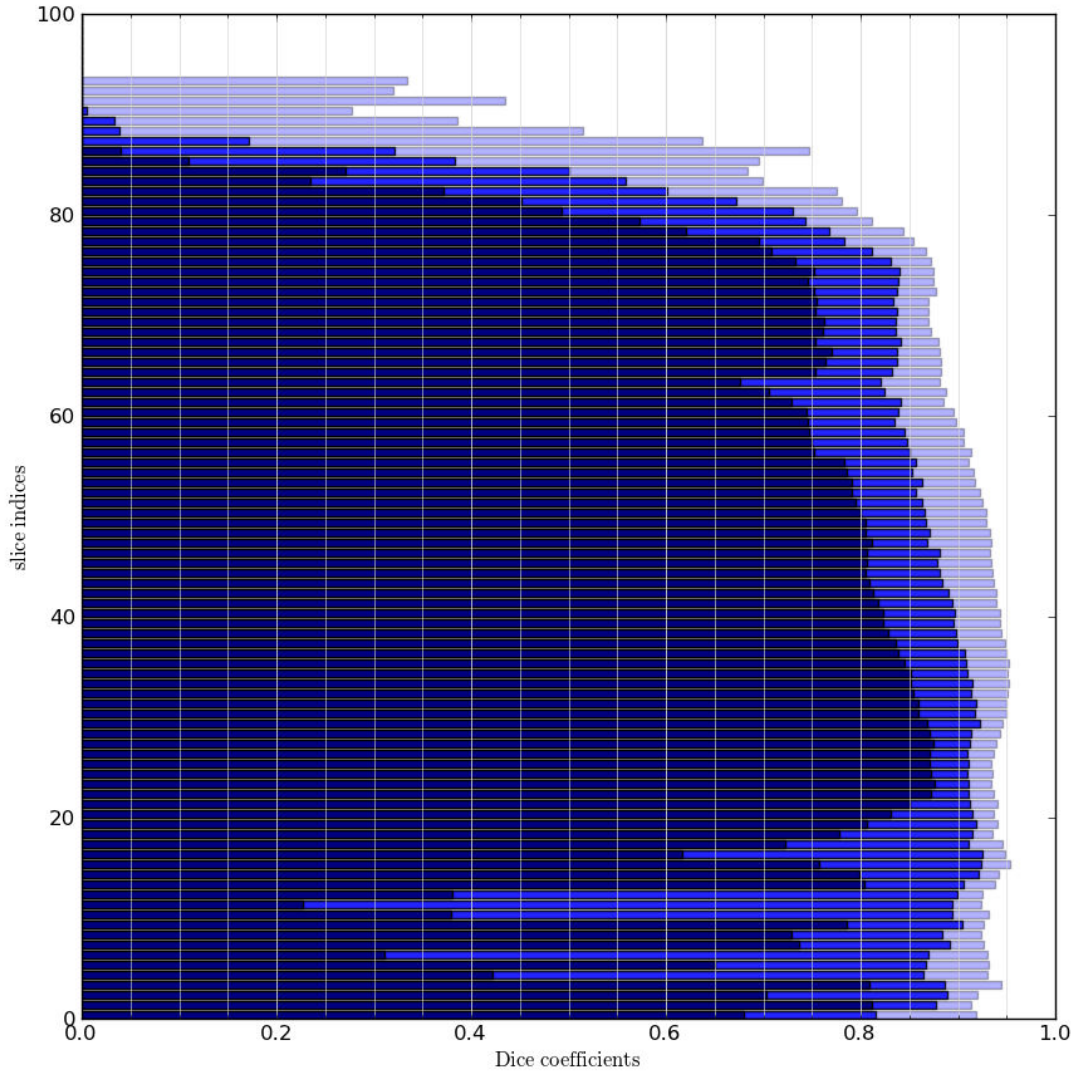

Figure 4.3.12 – Per-slice plot of the Dice coefficients for the segmentation method with constant weighting. The darker bars represent the average of the lower quartile, the intermediary bars the median and the lighter bars the average of the upper quartile.

4.4.2 *Shape space*

It is obvious that the entries of x are not independent from each other, but instead are highly tied to each other, at least for sets of close-by pixels. This implies that there exists an implicit lower-dimensional space in which any true segmentations reside. The principle of a shape space is to design a low-dimensional affine space approximating this implicit space. If we succeed, we expect a valid segmentation to lie "not too far" from the shape space; we also expect the projection of that segmentation onto the shape space to be a good approximation of the segmentation itself. PCA notoriously provides a simple mean to compute such a shape space.

Assume we possess a set $\mathcal{T}$ of co-registered segmented training volumes. We model a segmentation vector x as a random vector X with normal distribution $\mathcal{N}(\mathbf{x}_0, \Sigma)$, for which we possess a number of samples $\{\mathbf{z}_k\}_{k=1\ldots|\mathcal{T}|}$. Let us denote the centered segmentation vectors as $\mathbf{x}_k^c = \mathbf{z}_k - \mathbf{x}_0$. The expression of the empirical covariance Σ is given by:

$$\Sigma_{ij} = \frac{1}{|\mathcal{T}|} \sum_k \left(z_{ki} - x_{0i}\right)\left(z_{kj} - x_{0j}\right). \tag{4.4.3}$$

The previous can be reformulated in matrix form as

$$\Sigma = \frac{1}{|\mathcal{T}|} X_c X_c^\top \tag{4.4.4}$$

with $X_c = \left[\mathbf{x}_1^c \cdots \mathbf{x}_{|\mathcal{T}|}^c\right]$.

Then we compute the eigen-decomposition of Σ:

$$\Sigma = U \Delta U^\top, \tag{4.4.5}$$

where Δ is a diagonal matrix with the eigenvalues of Σ on the diagonal, and the columns of square matrix U are the eigenvectors of Σ, in the same order as the eigenvalues. Since Σ is symmetric and positive definite, its eigenvectors are orthogonal and its eigenvalues are all positive or zero. Moreover, since the number of training examples is smaller than the size of X: $|\mathcal{T}| \ll N$, we know we have at most $|\mathcal{T}|$ strictly positive eigenvalues, all other eigenvalues being zero.

We retain the $n < |\mathcal{T}|$ largest eigenvalues and associated eigenvectors, yielding an approximate covariance matrix:

$$\tilde{\Sigma} \approx U_n \Delta_n U_n^\top, \tag{4.4.6}$$

where Δ_n is the $n \times n$ diagonal matrix with the n largest eigenvalues of Σ and U_n is the non-square matrix whose columns are the n retained eigenvectors.

In practice, due to the large number of variables N, it is not possible to compute Σ. Thus, we use a different approach to perform the PCA. We define the $|\mathcal{T}| \times |\mathcal{T}|$ matrix:

$$\Sigma' = \frac{1}{|\mathcal{T}|} X_c^\top X_c. \tag{4.4.7}$$

This matrix being much smaller that Σ, we can perform an eigen-decomposition:

$$\Sigma' = V \Delta' V^\top \tag{4.4.8}$$

with obvious notations. Since

$$X_c \Sigma' = \Sigma X_c, \tag{4.4.9}$$

the eigenvalues of Σ' are all eigenvalues of Σ:

$$X_c \left(\Sigma' V\right) = X_c \left(V \Delta'\right), \tag{4.4.10}$$

$$\Sigma \left(X_c V\right) = \left(X_c V\right) \Delta'. \tag{4.4.11}$$

The eigenvectors of Σ are now obtained by computing

$$U_n = X_c V_n, \tag{4.4.12}$$

which will be normalized afterward to form an orthonormal basis.

The matrix U_n determines a *shape space*, in which the projection of any segmentation $\mathbf{x}$ in the shape space is given by

$$\tilde{\mathbf{x}} = \mathbf{x}_0 + U_n \gamma, \tag{4.4.13}$$

where γ is the coordinate vector of x in the shape space. We expect that $\tilde{x}$ will a good approximation of a valid segmentation $\mathbf{x}$. For any segmentation, we can write:

$$\mathbf{x} = \mathbf{x}_0 + U_n \gamma + \mathbf{dx}, \tag{4.4.14}$$

where $\mathbf{dx}$ is the deviation of $\mathbf{x}$ from the shape space. If the shape space models the space of valid segmentation well, then the norm of $\mathbf{dx}$ will be small when $\mathbf{x}$ is a valid segmentation.

4.4.3 *PCA model*

In order to obtain a segmentation which remains close to the shape space, we want to minimize the objective function 4.4.2 with respect to both $\mathbf{dx}$ and γ, while keeping $\mathbf{dx}$ small. This leads to the following functional:

$$E_{\text{RWpca}}\left(\mathbf{dx}, \gamma\right) = \left(\mathbf{dx} + U\gamma + \bar{\mathbf{x}}\right)^T \widetilde{L} \left(\mathbf{dx} + U\gamma + \bar{x}\right) + w_{\mathbf{dx}} \left\|\mathbf{dx}\right\|^2, \tag{4.4.15}$$

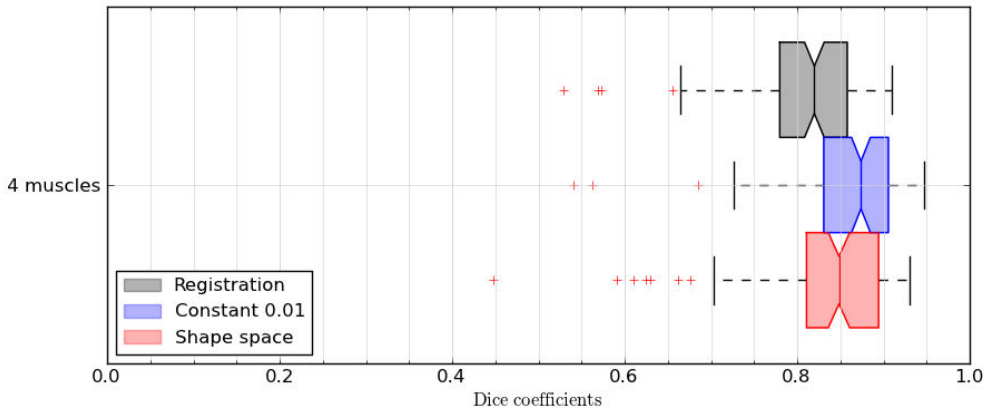

Figure 4.4.1 – Comparison of our RW with shape space method (bottom box) with the simpler RW with prior method with constant weighting (middle box, see section 4.3) and the registration-based baseline method (top box, see section 4.3.4).

where $w_{\mathbf{dx}}$ is a hyper parameter setting a constraint on the norm of $\mathbf{dx}$. We reformulate 4.4.15 as:

$$E_{\text{RWpca}}(\mathbf{y}) = (A\mathbf{y} + \mathbf{x}_0)^T \widetilde{L} (A\mathbf{y} + \mathbf{x}_0) + w_{\mathbf{dx}} \mathbf{y}^T B \mathbf{y}, \tag{4.4.16}$$

with

$$\mathbf{y} = \begin{bmatrix} \mathbf{dx} \\ \gamma \end{bmatrix}, \; A = [I_{KN} \; U], \; B = \begin{bmatrix} I_{KN} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{4.4.17}$$

where I_{KN} is the identity matrix of size $KN \times KN$.

The minimum of 4.4.16 verifies:

$$\left(A^T \widetilde{L} A + w_{\mathbf{dx}} B\right) \mathbf{y} = -A^T \widetilde{L} \mathbf{x}_0. \tag{4.4.18}$$

Obviously, the entries of $\mathbf{x}$ being probability distributions, assuming a Gaussian distribution can only lead to a rough approximation. For these reason, we expect the PCA shape space not to be a very good space of realistic solutions. In particular, it is unlikely that any projection into this shape space will yield a probability distributions.

4.4.4 *Experimental validation*

Our experiments were conducted on the same dataset as with the previous method (see section 4.3.4). This dataset consists of 30 volumes, in which four clinically relevant muscles (the quadriceps: vastus lateralis, vastus intermedius, vastus medialis and rectus femoris) were segmented in 30 volumes. Comparisons are made using Dice coefficients as previously.

In figure 4.4.1, we compare the Dice coefficients obtained with our shape space method to the previously presented simpler method of section 4.3 (using a constant scheme), and to the baseline registration method. It appears that the presented method does not improve over the previous, simpler method. This can be explained by the poor quality of the PCA generative model for probabilistic segmentations.

On figure 4.4.2, we show the effect of the PCA shape prior. We see that the PCA model is deformed from its initial shape (left column) to fit the boundaries of the test image (middle column). Due to the rough approximation of modeling probabilistic atlases with a linear subspace, the boundaries of the muscles are rather fuzzy. However, when allowing a small deviation (called $\mathbf{dx}$) from the model, as does our method, the contours are more accurately defined (right column). Notice there remains segmentation errors as the PCA model is not capable of deforming enough to fit unusual shapes while retaining a realistic topology. In the presented method, the main segmentation errors are due to the muscles with unusual shape – more elongated, smaller, etc. Modeling probabilistic segmentations with PCA model does not allow representing shapes which differ too much from standard shapes.

4.5 CONCLUSION

To conclude this chapter, we will say that directly incorporating prior knowledge of shape into the Random Walks framework is valid and compares favorably to state-of-the-art methods, but also shows some limitations. In the case of the simplest prior (sec. 4.3), the simplicity and the validity of the formulation allow to propose an easy-to-use method, quite efficient (less than 30 seconds for 13 muscles in a $220 \times 220 \times 64$ volume) and which does not require a large training dataset (we used between 12 and 29 samples during our experiments). However, errors inevitably appear when the average model $\mathbf{x}_0$ is too different from the segmented image. In particular, smaller muscles with too large inter-subject variations are likely to be incorrectly registered. In a very recent publication, Eslami et al. (2012) have proposed a very similar shape prior formulation for the Random Walks framework, called "guided" Random Walks. However, instead of using the empirical mean over the training samples, the author propose to use directly each training segmentation as a reference segmentation $\mathbf{x}_0$ and retain the infered segmentation $\mathbf{x}$ that most overlaps with $\mathbf{x}_0$. This promising approach is aimed at retrieving shapes which are quite different from the average model. For lack of time, we were not able to test this method on our data.

Ideally, we would like to handle large scale deformations by allowing the model to evolve in a low-dimensional shape space of valid segmentations. Small-scale deformations or subject-specific fine details would be segmented like in the simple model through the RW process. The PCA model presented in section 4.4 was an attempt at such flexible shape prior. Indeed, PCA is a well-know way to generate a low-

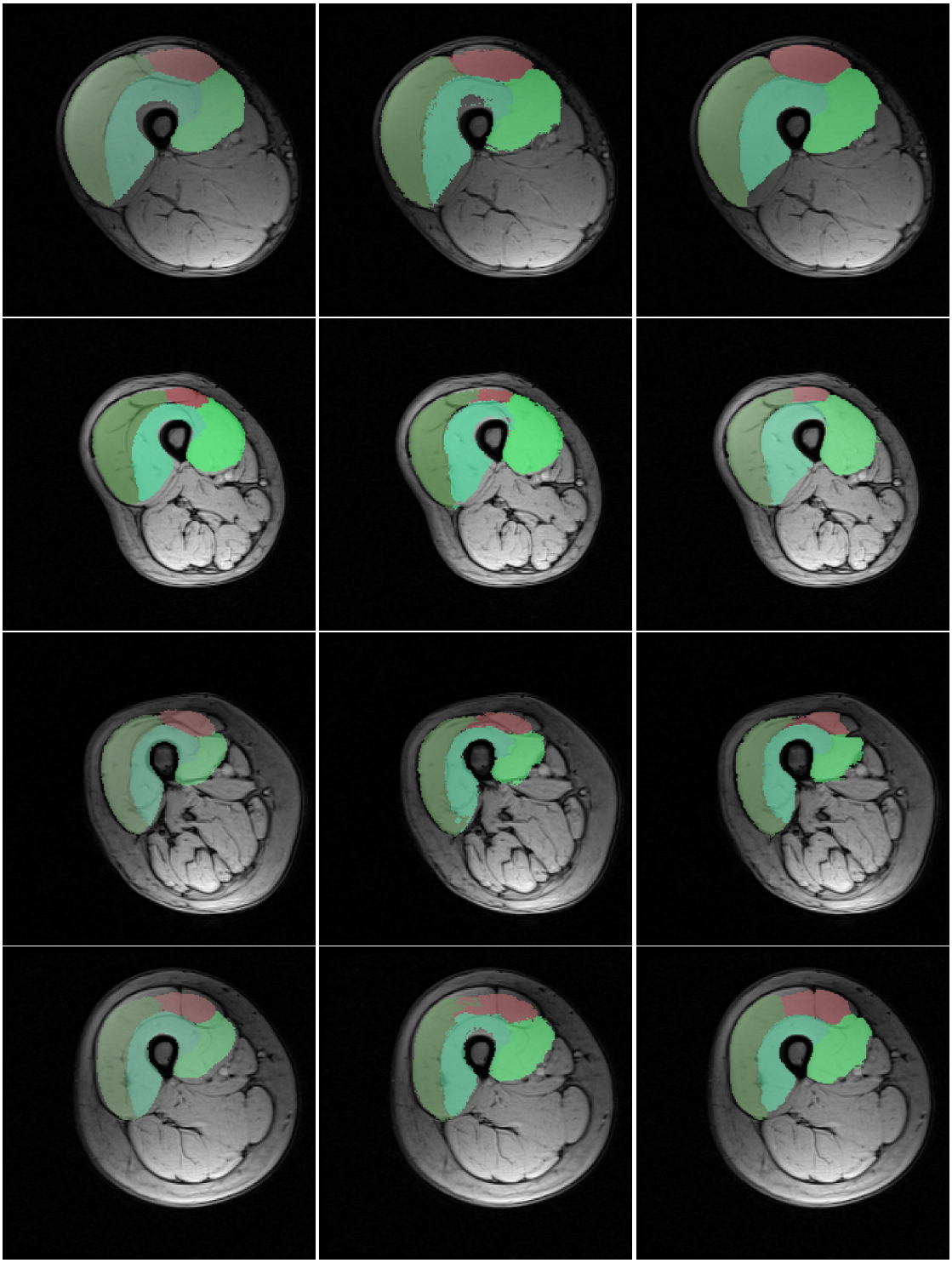

Figure 4.4.2 – Segmentation examples. Cross-sections of segmented volumes with: (left column) the registration method $\bar{\mathbf{x}} = \mathbf{x}_0$; (middle column) the shape-space component of the inferred segmentation $\bar{\mathbf{x}} = \mathbf{x}_0 + U\gamma$; (right column) the inferred segmentation $\bar{\mathbf{x}} = \mathbf{x}_0 + U\gamma + \mathbf{dx}$.

dimensional shape space. The mitigated performance we achieved is likely to be due to the weak modeling of probabilities with a PCA model.

Future direction would include finding a different shape space more compatible with probabilities, such as a barycenter model – indeed, a barycenter of a probability distribution remains a probability distribution. In the conclusion of this report, we also propose a different possible direction for dealing with large scale deformations, relying on additional connections between remote voxels.

5

LEARNING PARAMETERS FOR RANDOM WALKS-BASED SEGMENTATION

In the previous chapter, we empirically observed that the accuracy of the RW algorithm relies heavily on the relative weighting between the various contrast and prior terms. For instance, the contrast parameter used to construct the Laplacian matrix was chosen so that it gives the best results on average. In a more general approach, a different edge weighting function could be designed to extract different features of the image, more complex and specific appearance information. Then, instead of selecting the best Laplacian, it would be interesting to see if a particular linear combination of these Laplacians would yield better results than a single one of them. Similarly, we could also combine the different prior models.

At present, the proposed methods rely on a user to hand-tune the parameters or on exhaustive cross-validation. However, both these approaches quickly become infeasible as the number of terms in the RW objective function increase: a human would be unable to hand-tune a large number of parameters, and a thorough cross-validation would require vast and expensive computational resources.

In contrast to the RW literature, the problem of parameter estimation has received considerable attention in the case of discrete models such as conditional random fields (Lafferty et al., 2001). Recent years have witnessed the emergence of structured-output support vector machine (Structured SVM) as one of the most effective discriminative frameworks for supervised parameter estimation (Taskar et al., 2003; Tsochantaridis et al., 2004). Given a training dataset that consists of pairs of input and their ground-truth output, structured SVM minimizes the empirical risk of the inferred output with respect to the ground-truth output. The risk is determined by a user-specified loss function that measures the difference in quality between two given outputs. The parameters are estimated by minimizing a convex quadratic program, for which many efficient algorithms have been proposed (Joachims et al., 2009; Shalev-Shwartz et al., 2007; Taskar et al., 2003; Tsochantaridis et al., 2004). The structured SVM formulation has been successfully employed for many applications including graph-cuts based segmentation (Szummer et al., 2008), object detection (Blaschko and Lampert, 2008) and natural language parsing (Tsochantaridis et al., 2004).

Inspired by the efficacy of structured SVM in discrete models, we would like to discriminatively learn the parameters of the RW formulation. Such an approach would allow us to fully exploit the benefits of RW using numerous contrast and prior terms. To this end, a straightforward application of structured SVM would require a training dataset that consists of pairs of inputs – here, medical acquisitions using MRI scans – as

well as their ground-truth outputs – in our case, the *optimal* probabilistic segmentation. In other words, we require a human to provide us with the output of the RW algorithm for the best set of parameters. This is an unreasonable demand since the knowledge of the optimal probabilistic segmentation is as difficult to acquire as it is to hand-tune the parameters itself. Thus we are faced with a scenario where we cannot directly use structured SVM to estimate the desired parameters.

In order to handle the above difficulty, we propose a novel formulation for discriminative parameter estimation in the RW framework. Specifically, we learn the parameters using a weakly supervised dataset that consists of pairs of medical acquisitions and their hard segmentations. Unlike probabilistic segmentations, hard segmentations can be obtained easily from human annotators. We treat the optimal probabilistic segmentation that is compatible with the hard segmentation as a latent variable. Here, compatibility refers to the fact that the probability of the ground-truth label (as specified by the hard segmentation) should be greater than the probability of all other labels for each pixel/voxel. Clearly, the number of compatible probabilistic, or soft, segmentations for a given hard segmentation (that is, the size of the space for the latent variables) is uncountably infinite. Nonetheless, the resulting representation allows us to learn the parameters using the latent SVM formulation (Yu and Joachims, 2009).

While latent SVM does not result in a convex optimization problem, its local optimum solution can be obtained using the iterative concave-convex procedure (CCCP) (Yuille et al., 2002). At each iteration, CCCP performs two steps: (i) estimating a compatible probabilistic segmentation for each training sample using the current set of parameters – commonly referred to as annotation consistent inference (ACI); and (ii) updating the parameters by fixing the compatible probabilistic segmentations to those obtained during ACI. The second step of CCCP involves solving a structured SVM problem, which lends itself to efficient optimization. In order to make the overall algorithm computationally feasible, we propose a novel efficient approach for ACI based on dual decomposition (Bertsekas, 1999; Komodakis et al., 2007). We demonstrate the benefit of our learning framework over a baseline structured SVM using a challenging dataset of real 3D MRI volumes[1].

5.1 RELATED WORK

As mentioned earlier, the methods that employ the RW algorithm rely on parameters that have been hand-tuned or exhaustively cross-validated (see chapter 4). In contrast, the parameters of discrete models such as conditional random fields (CRFs) are often estimated using large supervised training datasets. One of the most effective supervised learning formulations is structured SVM (Taskar et al., 2003; Tsochantaridis et al., 2004), which generalizes the well-known binary SVM classifier to cases where the de-

1 This part was developed with the help of Danny Goodman and Pawan Kumar.

sired output is a vector of inter-dependent elements. Structured SVM minimizes a regularized upper bound on the empirical risk as defined by a user-specified loss function. The loss function provides a measure of the difference between two different outputs. Using the structured SVM formulation, the problem of parameter estimation reduces to that of solving a convex quadratic problem, for which several efficient algorithms have been proposed in the literature (Joachims et al., 2009; Shalev-Shwartz et al., 2007; Taskar et al., 2003; Tsochantaridis et al., 2004). The strong theoretical foundation and practical feasibility of structured SVM has led to its use in several real-world applications (Blaschko and Lampert, 2008; Szummer et al., 2008; Taskar et al., 2003; Tsochantaridis et al., 2004). One of the main deficiencies of structured SVM is that it requires a fully supervised training dataset, which consists of pairs of inputs and their desired outputs. Often, it is extremely challenging, or even impossible, to obtain full supervision (for example, in our case where a human annotator cannot specify the optimal probabilistic segmentation). However, it is relatively easy to obtain weakly supervised training samples.

In order to exploit the information present in such datasets, several researchers have independently proposed an extension to structured SVM known as latent SVM (Felzenszwalb et al., 2008; Smola et al., 2005; Yu and Joachims, 2009). Latent SVM treats any missing information in the human annotation as latent variables, which allows it to upper bound the empirical risk as a difference-of-convex optimization problem. Although its globally optimal solution cannot in general be computed in polynomial time, its local minimum or saddle point solution can be obtained efficiently using the CCCP algorithm (Yu and Joachims, 2009; Yuille et al., 2002). Since its first appearance, latent SVM has been steadily gaining popularity due to its accurate empirical performance (Felzenszwalb et al., 2008; Kumar et al., 2010, 2011; Yu and Joachims, 2009). Most of the previous instantiations of structured SVM and latent SVM have focused on discrete CRFs, that is, where the desired output is deterministic. In contrast, we propose a novel approach to learn the parameters of the probabilistic RW algorithm, where the latent variables model the unknown optimal soft segmentation that is compatible with a given hard segmentation.

5.2 PRELIMINARIES

5.2.1 *Notations*

In this part, the notations will differ slightly from chapter 4, for consistency with the SVM notations. We will refer to a 3D volume as $\mathbf{x}$. We use i to denote the index of a voxel in this volume, and denote the set of all indices as $\mathcal{V}$. In a hard segmentation, each voxel is assigned a label $s \in \mathcal{S}$ (for example, the index of a muscle). We will use z

to represent the human annotation (that is, the class labels of the voxels in x) in binary form:

$$\mathbf{z}(i,s) = \begin{cases} 1 & \text{if voxel } i \text{ is of class } s, \\ 0 & \text{otherwise.} \end{cases} \tag{5.2.1}$$

In other words, the binary form $\mathbf{z}$ of the annotation specifies delta distribution over the putative labels for each voxel. Our training dataset is a collection of training images $\mathbf{x}$ and hard segmentations z: $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{z}_k)\}_k$. Note that we use subscript k to denote the input index within a dataset, and parenthetical i to denote a voxel within a particular input.

We recall that the RW algorithm provides a probabilistic – or soft – segmentation of an input $\mathbf{x}$, which we denote by $\mathbf{y}$, that is,

$$\mathbf{y}(i,s) = \Pr[\text{voxel } i \text{ is of class } s], \ \forall i \in \mathcal{V}, s \in \mathcal{S}. \tag{5.2.2}$$

When using one contrast term and one prior model, the RW algorithm amounts to minimizing the following convex quadratic objective functional:

$$\begin{aligned} E(\mathbf{y}, \mathbf{x}) &= \mathbf{y}^\top L(\mathbf{x})\, \mathbf{y} + w^{\text{prior}} \|\mathbf{y} - \mathbf{y}_0\|^2_{\Omega_0(\mathbf{x})}, & (5.2.3) \\ &= \mathbf{y}^\top L(\mathbf{x})\, \mathbf{y} + E_{\text{prior}}(\mathbf{y}, \mathbf{x}). & (5.2.4) \end{aligned}$$

Here, $\mathbf{y}_0$ is a reference prior probabilistic segmentation dependent on appearance (Grady, 2005) or shape (Baudin et al., 2012a), and $\Omega_0(\mathbf{x})$ is a diagonal matrix that specifies a voxel-wise weighting scheme for $\mathbf{x}$. It can be shown that the energy functional 5.2.4 is convex and can be minimized over all possible probabilistic segmentations $\mathbf{y}$ in polynomial time. In fact, Grady (2006) showed that the RW algorithm amounts to solving a sparse linear system of equations, which lends itself to efficient optimization. We refer the reader to section 4.2.2 and to the work of Grady (2006) for further details.

5.2.2 *Parameters and Feature Vectors*

In the above description of the RW algorithm, we restricted ourselves to a single Laplacian and a single prior. However, our goal is to enable the use of numerous Laplacians and priors. To this end, let $\{L_\alpha\}_\alpha$ denote a known family of Laplacian matrices and $\{E_\beta(\cdot)\}_\beta$ denote a known family of prior energy functionals. In section 5.4, we will

specify the family of Laplacians and priors used in our experiments. We denote the general form of a linear combination of Laplacians and prior terms as:

$$L(\mathbf{x};\mathbf{w}) = \sum_{\alpha} w_\alpha L_\alpha(\mathbf{x}),\ w_\alpha \geq 0,\ \forall\alpha, \tag{5.2.5}$$

$$E_{\text{prior}}(\cdot,\mathbf{x};\mathbf{w}) = \sum_{\beta} w_\beta E_\beta(\cdot,\mathbf{x}),\ w_\beta \geq 0,\ \forall\beta. \tag{5.2.6}$$

Each term $E_\beta(\cdot,\mathbf{x})$ is of the form:

$$E_\beta(\mathbf{y},\mathbf{x}) = \left\|\mathbf{y}-\mathbf{y}_\beta\right\|^2_{\Omega_\beta(\mathbf{x})}, \tag{5.2.7}$$

where $\mathbf{y}_\beta$ is the β-th reference segmentation and $\Omega_\beta(\mathbf{x})$ is the corresponding voxel-wised weighting matrix (which are both known). We denote the set of all parameters as $\mathbf{w} = \{w_\alpha, w_\beta\}$. Clearly the RW energy 5.2.4 is linear in $\mathbf{w}$. and can therefore be formulated as:

$$\begin{aligned} E(\mathbf{y},\mathbf{x};\mathbf{w}) &= \mathbf{y}^\top L(\mathbf{x};\mathbf{w})\,\mathbf{y} + E_{\text{prior}}, &(5.2.8)\\ &= \mathbf{w}^\top \psi(\mathbf{x},\mathbf{y}), &(5.2.9)\end{aligned}$$

where $\psi(\mathbf{x},\mathbf{y})$ is known as the joint feature vector of $\mathbf{x}$ and $\mathbf{y}$. Note that by restricting the parameters to be non-negative (that is, $\mathbf{w} \geq 0$), we ensure that the energy functional $E(\cdot,\mathbf{x};\mathbf{w})$ remains convex.

5.2.3 *Loss Function*

As mentioned earlier, we would like to estimate the parameters $\mathbf{w}$ by minimizing the empirical risk over the training samples. The risk is specified using a loss function that measures the difference between two segmentations. In this work, we define the loss function as the number of incorrectly labeled voxels. Formally, let $\text{bin}(\mathbf{y})$ denote the underlying hard segmentation of the soft segmentation $\mathbf{y}$, that is $\text{bin}(\mathbf{y})(i,s) = \delta(s = \arg\max_{s\in\mathcal{S}} \mathbf{y}(i,s))$, where δ is the Kronecker function. The loss function is defined as

$$\Delta(\mathbf{z},\mathbf{y}) = 1 - \frac{1}{|\mathcal{V}|}\text{bin}(\mathbf{y})^\top \mathbf{z}, \tag{5.2.10}$$

where $\mathcal{V}$ is the set of all voxels and $|\cdot|$ denotes the cardinality of a set. If both segmentations are equal, then $\Delta(\mathbf{z},\mathbf{y}) = 0$; if they are disjoint, $\Delta(\mathbf{z},\mathbf{y}) = 1$.

Note for later that this function in *not concave in* $\mathbf{y}$, and that infinitely many soft segmentations $\mathbf{y}$ are compatible with the hard segmentation $\mathbf{z}$, that is, there is an infinite number of solutions $\mathbf{y}$ to the equation $\Delta(\mathbf{z},\mathbf{y}) = 0$.

5.3 PARAMETER ESTIMATION USING LATENT SVM

5.3.1 *Upper Bound on the Risk*

Given a dataset $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{z}_k), k = 1, \ldots, N\}$, which consists of inputs $\mathbf{x}_k$ and their hard segmentation $\mathbf{z}_k$, we would like to estimate parameters $\mathbf{w}$ such that the resulting inferred segmentations are accurate. Here, the accuracy is measured using the loss function $\Delta(\cdot, \cdot)$. Formally, let $\tilde{\mathbf{y}}_k(\mathbf{w})$ denote the soft segmentation obtained by minimizing the energy functional $E(\cdot, \mathbf{x}_k; \mathbf{w})$ for the k-th training sample, that is,

$$\tilde{\mathbf{y}}_k(\mathbf{w}) = \arg\min_{\mathbf{y}} \mathbf{w}^\top \psi(\mathbf{x}_k, \mathbf{y}). \tag{5.3.1}$$

We would like to learn the parameters $\mathbf{w}$ such that the empirical risk is minimized over all samples in the dataset. In other words, we would like to estimate the parameters $\mathbf{w}^\star$ such that

$$\mathbf{w}^\star = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_k \Delta(\mathbf{z}_k, \tilde{\mathbf{y}}_k(\mathbf{w})). \tag{5.3.2}$$

The above objective function is highly non-convex in $\mathbf{w}$, which makes it prone to bad local minimum solutions. To alleviate this deficiency, the latent SVM formulation upper bounds the risk for a sample $(\mathbf{x}, \mathbf{z})$ as follows:

$$\begin{aligned}
\Delta(\mathbf{z}_k, \tilde{\mathbf{y}}_k(\mathbf{w})) &= \Delta(\mathbf{z}_k, \tilde{\mathbf{y}}_k(\mathbf{w})) + \mathbf{w}^\top [\psi(\mathbf{x}_k, \tilde{\mathbf{y}}_k(\mathbf{w})) - \psi(\mathbf{x}_k, \tilde{\mathbf{y}}_k(\mathbf{w}))], && (5.3.3)\\
&\leq \min_{\Delta(\mathbf{z}_k, \hat{\mathbf{y}})=0} \mathbf{w}^\top \psi(\mathbf{x}_k, \hat{\mathbf{y}}) && (5.3.4)\\
&\quad - \left[\mathbf{w}^\top \psi(\mathbf{x}_k, \tilde{\mathbf{y}}_k(\mathbf{w})) - \Delta(\mathbf{z}_k, \tilde{\mathbf{y}}_k(\mathbf{w}))\right],\\
&\leq \min_{\Delta(\mathbf{z}_k, \hat{\mathbf{y}})=0} \mathbf{w}^\top \psi(\mathbf{x}_k, \hat{\mathbf{y}}) && (5.3.5)\\
&\quad - \min_{\overline{\mathbf{y}}} \left[\mathbf{w}^\top \psi(\mathbf{x}_k, \overline{\mathbf{y}}) - \Delta(\mathbf{z}_k, \overline{\mathbf{y}})\right].
\end{aligned}$$

The first inequality follows from the fact that the prediction $\tilde{\mathbf{y}}_k(\mathbf{w})$ has the *minimum possible energy* (see equation 5.3.1). Thus, its energy has to be less than or equal to the energy of any compatible segmentation $\hat{\mathbf{y}}$ with $\Delta(\mathbf{z}_k, \hat{\mathbf{y}}) = 0$. The second inequality is true since it replaces the *loss augmented* energy of the prediction $\tilde{\mathbf{y}}_k(\mathbf{w})$ with the minimum loss augmented energy.

RELATION TO THE CRF CASE In some cases – for instance, when learning conditional random fields (Szummer et al., 2008; Komodakis, 2011) – the loss function is *concave*, so that the minimum loss is only reached when the prediction $\hat{\mathbf{y}}$ is equal to the annotation $\mathbf{z}_k$:

$$\min_{\Delta(\mathbf{z}_k, \hat{\mathbf{y}})=0} \mathbf{w}^\top \psi(\mathbf{x}_k, \hat{\mathbf{y}}) = \mathbf{w}^\top \psi(\mathbf{x}_k, \mathbf{z}_k). \tag{5.3.6}$$

As a result, the upper bound on the empirical risk becomes

$$\Delta\left(\mathbf{z}_k, \tilde{\mathbf{y}}_k\left(\mathbf{w}\right)\right) \quad \leq \quad \mathbf{w}^\top \psi\left(\mathbf{x}_k, \mathbf{z}_k\right) - \min_{\bar{\mathbf{y}}} \left[\mathbf{w}^\top \psi\left(\mathbf{x}_k, \bar{\mathbf{y}}\right) - \Delta\left(\mathbf{z}_k, \bar{\mathbf{y}}\right)\right], \tag{5.3.7}$$

This upper bound is *convex* in $\mathbf{w}$, since the minimum of a collection of affine functions is concave (which becomes convex with the $-$ sign). As a result, one obtains a *learning algorithm* by minimizing the upper bound for each sample in the training set:

$$\begin{aligned} &\min_{\mathbf{w}, \xi_k \geq 0} \lambda ||\mathbf{w}||^2 + \frac{1}{N} \sum_{k=1}^{N} \xi_k, \\ &\text{s.t. } \mathbf{w}^\top \left[\psi\left(\mathbf{x}_k, \mathbf{z}_k\right) - \psi\left(\mathbf{x}_k, \bar{\mathbf{y}}\right)\right] + \Delta\left(\mathbf{z}_k, \bar{\mathbf{y}}\right) \leq \xi_k, \ \forall \bar{\mathbf{y}}, \forall k, \end{aligned} \tag{5.3.8}$$

where the slack variable ξ_k represents the upper bound of the risk for the k-th training sample. The regularizing term $||\mathbf{w}||^2$, weighted by hyper-parameter λ, ensures that we do not over-fit to the training samples. The previous algorithm is a *convex quadratic program* and corresponds to the formulation of Structural SVM with rescaled margin (Taskar et al., 2003; Tsochantaridis et al., 2004), for which may efficient algorithms have been proposed (Joachims et al., 2009; Shalev-Shwartz et al., 2007; Taskar et al., 2003; Tsochantaridis et al., 2004).

However, as mentioned earlier, the loss function $\Delta\left(\cdot, \cdot\right)$ is *not* concave in $\mathbf{w}$, which in turns leads to a non-convex upper bound on the empirical risk. Equation 5.3.6 is not verified, and solving the baseline Struct-SVM problem 5.3.8 is bound to fail in our case.

5.3.2 *Formulation with Latent SVM*

Let us recall the upper bound of the empirical risk derived above:

$$\Delta\left(\mathbf{z}_k, \tilde{\mathbf{y}}_k\left(\mathbf{w}\right)\right) \leq \min_{\Delta(\mathbf{z}_k, \hat{\mathbf{y}})=0} \mathbf{w}^\top \psi\left(\mathbf{x}_k, \hat{\mathbf{y}}\right) - \min_{\bar{\mathbf{y}}} \left[\mathbf{w}^\top \psi\left(\mathbf{x}_k, \bar{\mathbf{y}}\right) - \Delta\left(\mathbf{z}_k, \bar{\mathbf{y}}\right)\right].$$

While this upper bound is not convex, it is a difference of two convex functions in (Yu and Joachims, 2009). This observation allows us to obtain a local minimum or saddle point solution using the CCCP algorithm (Yu and Joachims, 2009; Yuille et al., 2002), which iteratively improves the parameters starting with an initial estimate $\mathbf{w}_0$ (the detailed optimization algorithm will be described shortly).

Using the upper bound on the empirical risk, the latent SVM optimization problem is specified as follows:

$$\min_{\mathbf{w},\,\xi_k \geq 0} \lambda \left\|\mathbf{w}\right\|^2 + \lambda' \left\|\mathbf{w} - \mathbf{w}_0\right\|^2 + \frac{1}{N}\sum_k \xi_k, \tag{5.3.9}$$
$$\text{s.t.} \min_{\Delta(\mathbf{x}_k,\hat{\mathbf{y}})=0} \mathbf{w}^\top \psi\left(\mathbf{x}_k, \hat{\mathbf{y}}\right) \leq \mathbf{w}^\top \psi\left(\mathbf{x}_k, \bar{\mathbf{y}}\right) - \Delta\left(\mathbf{z}_k, \bar{\mathbf{y}}\right) + \xi_k,\ \forall \bar{\mathbf{y}},\ \forall k,$$

where the slack variable ξ_k represents the upper bound of the risk for the k-th training sample. Note that we have added two regularization terms for the parameters $\mathbf{w}$. The first term $\left\|\mathbf{w}\right\|^2$, weighted by hyper-parameter λ, ensures that we do not over-fit to the training samples. The second term $\left\|\mathbf{w} - \mathbf{w}_0\right\|^2$, weighted by hyper-parameter λ', ensures that we do not obtain a solution that is very far away from our initial estimate $\mathbf{w}_0$. The reason for including this term is that our upper bound to the empirical risk may not be sufficiently tight. Thus, if we do not encourage our solution to lie close to the initial estimate, it may drift towards an inaccurate set of parameters. In section 5.4, we show the empirical effect of the hyper-parameters λ and λ' on the accuracy of the parameters.

A local minimum or saddle point solution for problem 5.3.9 can be found using the iterative CCCP method, which is outlined in Algorithm 5.1. It consists of two main steps at each iteration: (i) step 3, which involves estimating a compatible soft segmentation for each training sample – known as annotation consistent inference (ACI); and (ii) step 4, which involves updating the parameters by solving problem 5.3.10. In the following subsections, we provide efficient algorithms for both the steps.

Algorithm 5.1 The CCCP method for parameter estimation using latent SVM.

Input: Dataset $\mathcal{D}$, λ, λ', $\mathbf{w}_0$,ε

1: Set $t = 0$. Initialize $\mathbf{w}_t = \mathbf{w}_0$.
2: **repeat**
3: Compute $\mathbf{y}_k^\star = \arg\min_{\hat{\mathbf{y}}_k,\,\Delta(\mathbf{z}_k,\hat{\mathbf{y}}_k)=0} \mathbf{w}_t^\top \psi\left(\mathbf{x}_k, \hat{\mathbf{y}}_k\right),\ \forall k$.
4: Update the parameters by solving the following problem

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w},\,\xi_k \geq 0} \lambda \left\|\mathbf{w}\right\|^2 + \lambda' \left\|\mathbf{w} - \mathbf{w}_0\right\|^2 + \frac{1}{N}\sum_k \xi_k, \tag{5.3.10}$$
$$\text{s.t. } \mathbf{w}^\top \psi\left(\mathbf{x}_k, \mathbf{y}_k^\star\right) \leq \mathbf{w}^\top \psi\left(\mathbf{x}_k, \bar{\mathbf{y}}\right) - \Delta\left(\mathbf{z}_k, \bar{\mathbf{y}}\right) + \xi_k,\ \forall \bar{\mathbf{y}},\ \forall k,$$

5: $t = t + 1$
6: **until** The objective function of problem 5.3.9 does not decrease below tolerance ε.

5.3.3 *Annotation Consistent Inference*[2]

Given an input $\mathbf{x}$ and its hard segmentation $\mathbf{z}$, ACI requires us to find the soft segmentation $\mathbf{y}^\star$ with the minimum energy, under the constraint that it should be *compatible* with $\mathbf{z}$ (see step 3 of Algorithm 5.1). We denote the ground truth label of a voxel i by s_i, that is, $s_i = \arg\max_s \mathbf{z}(i,s)$, and the set of all voxel indices by $\mathcal{V}$. Using our notation, ACI can be formally specified as

$$\mathbf{y}^\star = \arg\min_{\mathbf{y}\in\mathcal{C}(\mathcal{V})} \mathbf{y}^\top L(\mathbf{x};\mathbf{w})\,\mathbf{y} + E_{\text{prior}}(\mathbf{y},\mathbf{x};\mathbf{w}). \tag{5.3.11}$$

Here, $\mathcal{C}(\mathcal{V})$ is the set of all compatible probabilistic segmentations, that is,

$$\begin{aligned}
&\mathcal{C}(\mathcal{V}):\\
&\qquad \mathbf{y}(i,s) \geq 0,\ \forall s \in \mathcal{S}, && (5.3.12)\\
&\qquad \sum_{s\in\mathcal{S}} \mathbf{y}(i,s) = 1,\ \forall i \in \mathcal{V}, && (5.3.13)\\
&\qquad \mathbf{y}(i,s_i) \geq \mathbf{y}(i,s),\ \forall i \in \mathcal{V}, \forall s \in \mathcal{S}. && (5.3.14)
\end{aligned}$$

Constraints 5.3.12 and 5.3.13 ensure that $\mathbf{y}^\star$ is a valid probabilistic segmentation. The last set of constraints 5.3.14 ensure that $\mathbf{y}^\star$ is compatible with the hard ground truth $\mathbf{z}$. Note that in the absence of constraint 5.3.14, the other constraints are not necessary (see section 4.2.2) and the above problem can be solved efficiently using the RW algorithm. However, since the ACI problem requires the additional set of compatibility constraints, we need to develop a novel efficient algorithm to solve the above convex optimization problem. To this end, we exploit the powerful dual decomposition framework (Bertsekas, 1999; Komodakis et al., 2007).

DUAL DECOMPOSITION FOR THE ACI Briefly, dual decomposition (DD) allows us to iteratively solve a convex optimization problem of the form

$$\mathbf{y}^\star = \arg\min_{\mathbf{y}\in\mathcal{F}} \sum_{m=1}^{M} g_m(\mathbf{y}), \tag{5.3.15}$$

that is, any problem which can be exactly decomposed into a sum of (preferably simpler) sub-problems with functionals $g_m(\mathbf{y})$. At each iteration t, DD solves the set of slaves problems

$$\mathbf{y}_m^\star = \arg\min_{\mathbf{y}\in\mathcal{F}} \left(g_m(\mathbf{y}) + \rho_m^t \mathbf{y}\right), \tag{5.3.16}$$

2 This section was developed with the help of Puneet Kumar and Pawan Kumar

where ρ_m^t are the dual variables satisfying $\sum_m \rho_m^t = 0$. The dual variables are initialized as $\rho_m^0 = 0$, $\forall m$, and updated at iteration t as follows:

$$\rho_m^{t+1} \leftarrow \eta^t \left(\mathbf{y}_m^\star - \frac{1}{M} \sum_{n=1}^{M} \mathbf{y}_n^\star \right),$$

where η^t is the learning rate at iteration t. Under fairly general conditions, this iterative strategy converges to the globally optimal solution of the original problem, that is, $\mathbf{y}^\star = \mathbf{y}_m^\star$, $\forall m$. We refer the reader to the works of Bertsekas (1999); Komodakis et al. (2007) for details.

In order to specify our slave problems, we divide the set of voxels $\mathcal{V}$ into subsets $\mathcal{V}_m$, $m = 1, \ldots, M$, such that each pair of neighboring voxels $(i, j) \in \mathcal{N}$ appear together in *exactly one* subset $\mathcal{V}_m$. Given such a division of voxels, our slave problems correspond to the following:

$$\min_{\mathbf{y}_m \in \mathcal{C}(\mathcal{V}_m)} \mathbf{y}_m^\top L_m(\mathbf{x}; \mathbf{w})\, \mathbf{y}_m + E_m^{\text{prior}}(\mathbf{y}_m, \mathbf{x}; \mathbf{w}) + \rho_m^t \mathbf{y}_m, \tag{5.3.17}$$

where $L_m(\mathbf{x}; \mathbf{w})$ is the Laplacian matrix whose non-zero entries correspond to the voxels in $\mathcal{V}_m$, which leads to

$$L = \sum_m L_m. \tag{5.3.18}$$

The prior prior energy functions E_m^{prior} modifies the original prior E_{prior} by weighting each voxel $i \in \mathcal{V}_m$ by the reciprocal number of subsets that contain i. This weighting scheme allows us to decompose the prior term exactly, that is

$$E_{\text{prior}}(\mathbf{y}, \mathbf{x}; \mathbf{w}) = \sum_m E_m^{\text{prior}}(\mathbf{y}, \mathbf{x}; \mathbf{w}), \tag{5.3.19}$$

which is the consequence of the following trivial equality:

$$\|\mathbf{y} - \mathbf{y}_\beta\|_{\Omega_\beta}^2 = \sum_m \left\| \mathbf{y} - \mathbf{y}_\beta \right\|_{R\Omega_\beta}^2, \tag{5.3.20}$$

where R is the diagonal matrix such that $R_{ii} = 1/\left|\{\mathcal{V}_n, \text{ s.t. } i \in \mathcal{V}_n\}\right|$. In other words, the prior term for each voxel $i \in \mathcal{V}$ is multiplied by the number of times voxel i appears in a slave problem $\mathcal{V}_n$.

Thus, the slave problems defined above provide a valid reparametrization of the original problem 5.3.11, that is $g(\mathbf{y}) = \sum_m g_m(\mathbf{y})$. Decomposing the set of constraints $\mathcal{C}(\mathcal{V}_m)$ for each subset of variables is straightforward. By using small subsets $\mathcal{V}_m$ we can optimize each slave problem in every iteration *independently* using a standard quadratic programming solver. The sub-problems – which are quadratic programs – can be made sufficiently small to be solved in parallel with a dedicated solver. In

our experiments, we used the Mosek solver. To the best of our knowledge, this is the first application of dual decomposition to solve a probabilistic segmentation problem under linear constraints.

5.3.4 *Parameter Update*

We now describe how the parameters can be efficiently updated by solving problem 5.3.10 for a fixed set of soft segmentations $\mathbf{y}_k^\star$. Note that while problem 5.3.10 is convex, the number of constraints is of the order of the number of possible soft segmentations of an input. In other words, it consists of an infinite number of constraints. Nonetheless, the problem can be solved efficiently using the cutting plane method outlined in Algorithm 5.2. The method starts by specifying no constraints for any of the training samples. At each iteration, it finds the most violated constraint for each sample (step 3), and updates the parameters (step 5) until the increase in the objective function is less than the tolerance.

Algorithm 5.2 The cutting plane method for updating the parameters.

Input: Dataset $\mathcal{D}$, $\mathbf{y}_k^\star$, λ, λ', $\mathbf{w}_t$, ε

1: Set $\mathcal{W}_k = \emptyset$ for all k. Initialize $\mathbf{w} = \mathbf{w}_t$.
2: **repeat**
3: Compute $\bar{\mathbf{y}}_k = \arg\min_{\mathbf{y}} \; \mathrm{w}^\top \psi\left(\mathbf{x}_k, \mathbf{y}\right) - \Delta\left(\mathbf{z}_k, \mathbf{y}\right), \; \forall k.$
4: Update $\mathcal{W}_k = \mathcal{W}_k \bigcup \overline{\mathbf{y}}_k$.
5: Update the parameters by solving the following problem:

$$\begin{aligned} \min_{\mathbf{w}, \xi_k \geq 0} \; & \lambda ||\mathbf{w}||^2 + \lambda' ||\mathbf{w} - \mathbf{w}_0||^2 + \frac{1}{N} \sum_k \xi_k, \\ \text{s.t. } \mathbf{w}^\top & \left[\psi\left(\mathbf{x}_k, \mathbf{y}_k^\star\right) - \psi\left(\mathbf{x}_k, \overline{\mathbf{y}}\right)\right] \leq \xi_k - \Delta\left(\mathbf{z}_k, \overline{\mathbf{y}}\right), \; \forall \overline{\mathbf{y}} \in \mathcal{W}_k, \; \forall k. \end{aligned} \tag{5.3.21}$$

6: **until** The objective function of problem 5.3.21 does not increase above tolerance ε.

The main ingredient of the cutting plane method is the computation of the most violated constraint, that is,

$$\overline{\mathbf{y}} = \arg\min_{\mathbf{y}} \; \mathrm{w}^\top \psi\left(\mathbf{x}, \mathbf{y}\right) - \Delta\left(\mathbf{z}, \mathbf{y}\right) \tag{5.3.22}$$

for a given sample $(\mathbf{x}, \mathbf{y})$. While in most of the previous instantiations of structured SVM and latent SVM, the above problem can be solved optimality, in our case it presents a computational challenge. The reason for this is that the loss function, defined in equation 5.2.10, is not concave in $\mathbf{y}$, which makes the above problem non-

convex. One obvious solution to this challenge would be to replace the loss function by its concave approximation, that is,

$$\Delta'(\mathbf{z}, \mathbf{y}) = 1 - \frac{1}{|\mathcal{V}|}\mathbf{y}^\top \mathbf{z}. \tag{5.3.23}$$

However, the resulting convex approximation of problem 5.3.10 would not lend itself to the RW algorithm, as it does not guaranty that the output is a probability distribution as required by the RW probabilistic framework. In other words, we would need to resort to a dual decomposition strategy to obtain a constraint at each iteration of the cutting plane method. Such an approach would render the overall CCCP algorithm computationally infeasible. Instead, in this work, we obtain the approximate most violated constraint as the predicted segmentation, that is,

$$\overline{\mathbf{y}} = \arg\min_{\mathbf{y}} \mathbf{w}^\top \psi(\mathbf{x}, \mathbf{y}). \tag{5.3.24}$$

The above problem only requires the minimization of the energy (without any loss terms), which can be performed efficiently using the RW algorithm. As our results indicate, coupled with the regularization of the parameters, this simple cutting plane strategy provides an accurate set of parameters.

5.4 EXPERIMENTAL VALIDATION

We demonstrate the efficacy of our approach on a challenging dataset. Specifically, we test the hypothesis that it is important to model the unknown optimal soft segmentations of the training samples using latent variables in order to learn an accurate set of parameters. In what follows, we describe the experimental setup and our results in detail.

Dataset

We use the 30 volume dataset described in section 5.4. This datasets consists of volumes of the thigh region of dimensions $224 \times 224 \times 100$. The various segments correspond to 4 different muscle groups together with the background class. We randomly split the dataset into 80% for training and 20% for testing. In order to reduce the training time for both our method and the baselines, we divide each volume into 100/2 volumes of dimension $224 \times 224 \times 2$. During dual decomposition, we further divide the volumes into dimensions $10 \times 10 \times 2$ in order to specify the slave problems.

Laplacians and Prior Terms

We used 4 different Laplacians based on the standard weighting function given in section 4.1, 3 of them corresponding to $\beta \in \{10, 50, 100\}$ – to be used with volumes which have been normalized with respect to their standard deviation. The last one to the alternate expression given in footnote 2 with $\beta = 100$. Furthermore, we use two shape priors and one appearance prior: the first one as the shape prior with constant weighting as described in section 4.3, the second one using the entropy weighting also described in section 4.3, and the third one being the contrast based prior of Grady (2005). This results in a total of 7 parameters to be estimated.

Methods

The main hypothesis of our work is that it is important to represent the unknown optimal soft segmentation using latent variables. In order to test this hypothesis, we compare our method with a baseline structured SVM that replaces the latent variables with the given hard segmentations. In other words, our baseline estimates the parameters by solving the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{w},\, \xi_k \geq 0} \lambda ||\mathbf{w}||^2 + \lambda' \left\|\mathbf{w} - \mathbf{w}_0\right\|^2 + \frac{1}{N} \sum_{k=1}^{N} \xi_k, \\ &\text{s.t. } \mathbf{w}^\top \left[\psi\left(\mathbf{x}, \mathbf{z}_k\right) - \psi\left(\mathbf{x}, \bar{\mathbf{y}}\right)\right] \leq \xi_k - \Delta\left(\mathbf{z}, \bar{\mathbf{y}}\right), \ \forall \bar{\mathbf{y}}, \forall k, \end{aligned} \tag{5.4.1}$$

which is the same as 5.3.8 with an added proximal regularization term. The above problem can be solved using Algorithm 5.2, where the imputed soft segmentations $\mathbf{y}_k^\star$ are replaced by the hard segmentations $\mathbf{z}_k$.

During our experiments, we found that replacing the hard segmentation with a pseudo soft segmentation based on the distance transform systematically decreased the loss of the output. Thus, the method referred to as "Baseline" uses a structured SVM with distance-transform "softened" segmentations. The transformed segmentation, denoted as $\tilde{\mathbf{z}}_k$, is computed in the following manner:

$$\tilde{\mathbf{z}}_k\left(i, s\right) = \frac{\exp\left(-\gamma \mathrm{dist}_k\left(i; s\right)^2\right)}{\sum_t \exp\left(-\gamma \mathrm{dist}_k\left(i; t\right)^2\right)}, \tag{5.4.2}$$

where $\mathrm{dist}_k\left(i, s\right)$ is the Euclidean distance from pixel i to the closest pixel with label s in the hard segmentation $\mathbf{z}_k$; if pixel i already belongs to class s, then $\mathrm{dist}_k\left(i; s\right) = 0$. This term is normalized by a sum over all labels such that $\tilde{\mathbf{z}}_k$ is a probability distribution. Parameter $\gamma > 0$ determines the amount of spatial smoothing of the distribution, that is, the smaller γ is, the more progressive are the transitions between labels. Fig. 5.4.1 illustrates the softening process of the segmentation.

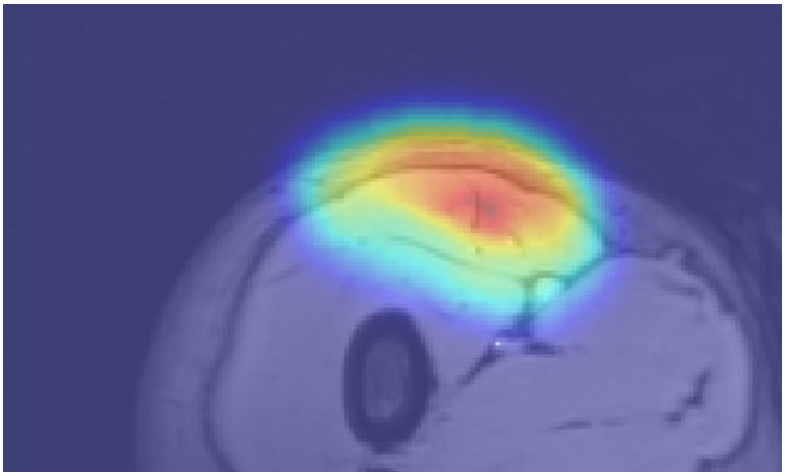

Figure 5.4.1 – Probability map of one muscle using the distance transform to "soften" the probabilistic segmentation spatially.

Results

Figure 5.4.2 shows the test loss for three different methods: (i) the initial hand-tuned parameters $\mathbf{w}_0$; (ii) the baseline structured SVM with distance transforms; and (iii) our proposed approach using latent SVM. As can be seen from Fig. 5.4.2, latent SVM provides significantly better results than the baselines – even when using the distance transform (which is the baseline method displayed in Fig. 5.4.2).

For the 4 x 5 hyper-parameter settings that we report (that is, four different values of λ and 5 different values of λ'), latent SVM is significantly better than SVM in 15 cases, and significantly worse in only 2 cases. Note that latent SVM provides the best results for very small values of λ', which indicates that the upper bound on the empirical risk in tight. As expected, for sufficiently large values of λ', all the methods provide similar results. For the best settings of the corresponding hyper-parameters, the percentage of incorrectly labeled voxels as follows: (i) for w_0, 13.5%; (ii) for structured SVM, 10.0%; and (iii) for latent SVM, 9.2%. Fig. 5.4.3 shows some example segmentations for the various methods.

Conclusion

Results show that our approach allows to estimate good parameters for the Random Walks segmentation method using shape priors. At the moment, since tests were performed on a small dataset (6 test volumes, 24 training volumes, 4 muscles), we can only claim it to be a proof of concept. Our results provide empirical evidence for our main hypothesis, namely the importance of treating the underlying soft segmentations as latent variables. This can be seen not only in the overall improvement achieved

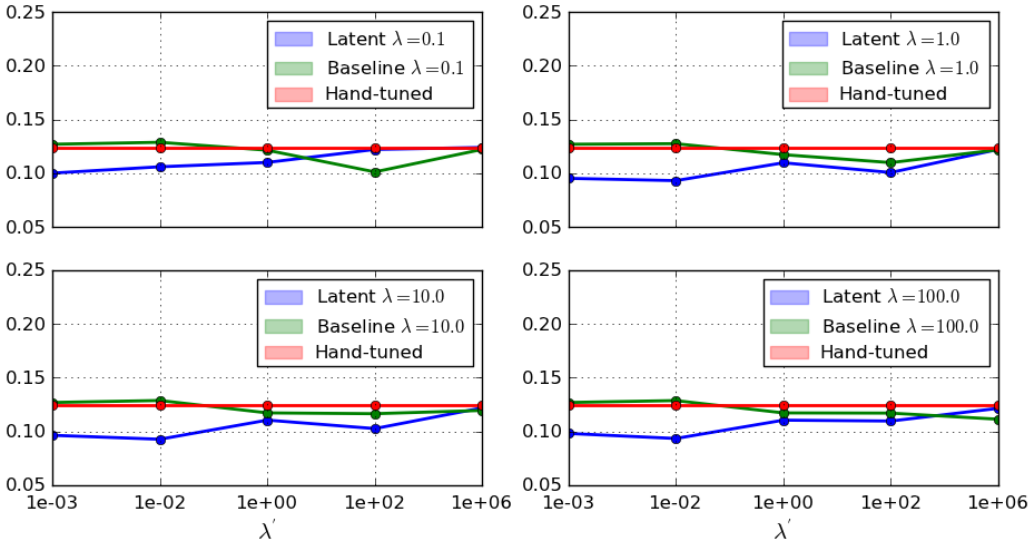

Figure 5.4.2 – Test loss for three different methods. Each figure corresponds to a different λ – that is, the SVM regularization parameter. Curves are plotted with respect of λ' – that is, the proximal regularization parameter, which penalizes deviations from the hand-tuned parameters $\mathbf{w}_0$. "Baseline" corresponds to the standard struct-SVM method using distance-transformed pseudo-soft segmentations $\tilde{\mathbf{z}}_k$ as annotation vectors.

by our approach, but also by the relative robustness of latent SVM to the choice of hyperparameters. This allows us to envision more extensive estimation processes. By carefully choosing more Laplacians and shape priors and learning their relative weighting, our framework may be able to provide bigger improvements. In particular, we are interested in learning separate parameters for different image regions. Indeed, some regions have edges which are consistently visible, while others do not. Thus it would be useful to change the balance between the model and the contrast term based on this knowledge, a task that would be impossible to fulfill manually. Moreover the latent variable formulation opens the door for more semi-supervised learning: authorizing the use of partially or coarsely segmented ones would allow us to increase the size of the learning set by orders of magnitude. An improvement of the method could be tested in replacing the implied L_2 norm of the regularization terms – $\|\mathbf{w}\|^2$ and $\|\mathbf{w}-\mathbf{w}_0\|^2$ – by a L_1 norm. Such change could promote sparsity in the parameter values, such that only the most relevent terms are selected, rather that searching for a combination of all terms.

However, our experiments revealed an important drawback: computing time. Indeed SVM learning is an iterative process and in each iteration, all the training samples have to be segmented (during the LAI process). Each segmentation task is taking several dozen of seconds for the full image on a powerful desktop computer. Even

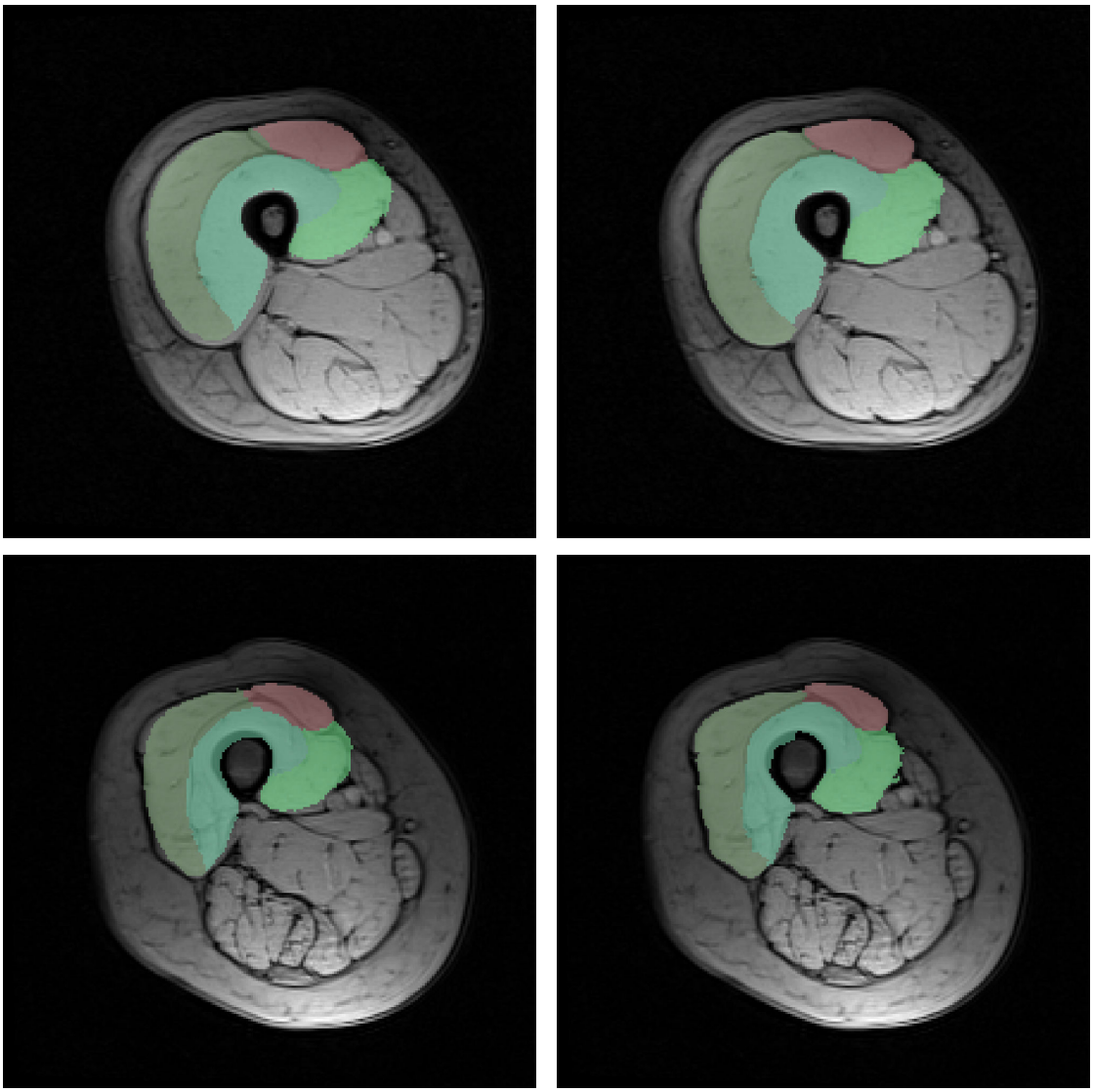

Figure 5.4.3 – (left column) segmentation using $\mathbf{w}_0$ (hand-tuned parameters); (right column) segmentation using learning $\mathbf{w}$ using latent structured SVM. Segmentations with learned parameters are more accurate.

more problematic is the ACI stage, where we had to solve a large quadratic program using dual decomposition. Even though it allowed us to estimate the latent soft segmentation, this process takes several dozens of minutes for a single volume. All our experiments were conducted on the computer cluster of the École Centrale Paris. Even there, a single experiment; for a given set of parameters λ, λ', etc., took more than 2 hours for estimating seven parameters. However, we point out that the code written in Python and not optimized for efficient computation. Besides, all presented algorithms (cutting plance, CCCP, ACI) are highly parallelizable.

A second limitation is the apparent weakness of the improvements in the segmentation results (as measured by loss decrease). The limited improvements may be due to the small number of parameters we estimated and by the fact that there were little room for improvement with the segmentation method considered. As always, parameter learning can only improve the results of a method within the limits of the method. Here, the single reference shape model may not be able to provide results which are much better than what was obtained by hand-tuning the parameters. To conclude, we say that this learning method appears to be effectively learning and could be re-used and extended to more interesting cases, but we also believe that the segmentation model is even more crucial and should be improved as well.

6 SUMMARY AND CONCLUSION

6.1 MATERIAL

Most of the presented work was coded using the open source and highly portable Python programming language, in combination with the scientific libraries Numpy and Scipy. The author would like to contribute to publicize to the scientific community the great versatility and power of this free software. All development and computations were performed on standard desktop machines, except for the learning process presented in chapter 5 which had to be performed on the computer cluster of École Centrale Paris.

All MRI volumes used for testing and training the segmentation methods were acquired on the 3 Tesla Siemens scanner of the Institute of Myology.

Moreover, all manual segmentations where performed using the free and open source ITK Snap software (Yushkevich et al., 2006).

6.2 SUMMARY OF THE CONTRIBUTIONS

In this thesis report, we have presented several contributions towards solving the difficult problem of muscle MRI segmentation. All presented methods require a require a registration stage – either rigid or non-rigid – of a training set of atlases onto the target image.

Our first contribution takes advantage of the good performance of interactive Random Walks segmentation algorithm (Grady, 2006) by automatically computing the seeds, which are normally drawn manually. This approach consists in generating unlabeled seeds over the target image, and labeling them automatically by detecting the contours between the muscles and based on prior knowledge of the relative position of the muscles extracted from one reference atlas. Labeling is achieved through a graph-based discrete energy approximate minimization procedure.

In a different approach, we proposed to integrate prior-knowledge shape directly to the Random Walks formulation. In a first contribution, similar to an intensity prior formulation (Grady, 2005), we added a term to the Random Walks objective that prevents the segmentation to deviate too much from a known average segmentation. We show this shape prior term does not break the probabilistic framework of the RW formulation with an easy-to-minimize objective functional – which amounts to solving a large sparse linear system. We propose several weighting schemes for the shape term based on reasonable intuitions. Obtained results showed that our method is favorably

comparable to the current state-of-the-art. In a second approach, we try to expand the range of accessible shapes in order to improve the segmentation performance on morphologies far from the average. We build a low-dimensional shape space through Principal Component Analysis on the training set. Then we propose an objective in which the segmentation is constrained to remain in the vicinity of the shape space while it minimizes the RW objective. However, the results indicate that this approach suffers from the poor modelization of probabilistic segmentations using a PCA as well as the small number of samples in the training set.

In a last part, we proposed a novel discriminative learning framework to estimate the parameters for the probabilistic Random Walks segmentation algorithm. Indeed, the information in the training dataset had not been exploited to finely tune the proposed models up to this point. In order to deal with the lack of full supervision, we represented the optimal soft segmentation that is compatible with the hard segmentation of each training sample as a latent variable. This allowed us to formulate the problem of parameter estimation using latent SVM, which upper bounds the empirical risk of prediction with a difference of convex optimization program. Using a challenging clinical dataset of MRI volumes, we demonstrated the efficacy of our approach over the baseline method that replaces the latent variables with the given hard segmentations. Our formulation allows for a large number of parameters to be estimated using a training dataset.

6.3 FUTURE DIRECTIONS

Although it appears that the automatic seed method presented in chapter 3 slightly outperforms the Random Walks-based model of chapter 4, we believe that the latter method has more potential than the former for several reasons: (i) the automatic seeding method requires one more computing stage than the RW-based method (registration, seeding, segmentation for the former, registration and segmentation for the latter) and is more likely to propagate errors from one stage to the next. Integrating the shape prior within the segmentation method as does our RW-based method in a very simple way is likely to be more robust – which the smaller number of outliers for this method seems to confirm (in Fig. 4.3.11). (ii) Many possible extensions of the RW-based method can be explored, for which we propose two:

A. Incorporate long-range connections between pixels instead of using only adjacent edges – this would help in situations where the prior shape model is too different from the target image. This could be achieved in the following manner: generate random edges between voxels, targeting an optimal edge length based on a small world perspective (Kleinberg, 2000). On the training set, learn typical distributions of the voxel intensity along the edges, in order to build a contour detector. The geodesic distance or the commute-time distance can be used as features to the classifier. Run the classifier on the tested image to gener-

This idea was suggested by Danny Goodman.

ate the weights for the long-range edges: voxels with no contours between them would have a transition probability close to one. The added edges do not change the formulation of the Random Walks algorithm. In addition, if one wishes to increase the probability of label change when a contour is found between the connected voxels, it can also be done by introducing non-zero inter-label weights in the global Laplacian (that is, the Laplacian which gather all RW equations, as seen in section 4.4). It can be shown that this formulation retains a probabilistic meaning, and therefore also resolves in a large sparse linear system.

B. Instead of a PCA (see sec. 4.4), build a shape space from a barycenter of training segmentations – a formulation which may better fit within the probabilistic framework since a barycenter of probability distributions is still a probability distribution.

(iii) Apart form shape priors, different types of priors can be easily integrated to the RW-based method, in particular appearance-based priors involving intensity (Grady, 2005).

Regarding parameter estimation, as mentioned in chapter 5, future work could involve exploring extensions where the parameters specify spatially varying terms in the RW objective function. This would allow us to encourage the presence of edges in between different segments, while at the same time not penalize the absence of edge in the background. The latent SVM framework can also be used to estimate parameters from partially segmented data. Such an approach would allow us to scale the size of the training dataset by orders of magnitude.

Although their is plenty of room for improvement in automatic techniques, one should not loose sight that segmentation methods are to be used by researchers and clinicians in practical settings. In particular, considering no fully automatic method will provide perfect results in a near future, one should still consider user interaction as an acceptable mean. Thus, it is important that segmentation methods remain simple to use, and that learning methods remain practical, the latter of which the presented work did not fully achieve. More importantly, it is crucial to develop new interactive methods for correcting the imperfect output of automatic methods. Otherwise, they may remain useless if it takes as much time to correct an automatic segmentation than to segment the image manually from the start.

A

MRI BASICS

A.1 BASICS IN NUCLEAR MAGNETIC RESONANCE

The aim of this section is to give a brief insight into the origin of the contrast in MRI. It is based on the basics of Magnetic Resonance Imaging by Fleckenstein et al. (1996). Nuclear Magnetic Resonance is a quantum phenomenon affecting certain atomic nuclei having a non-zero nuclear spin $\overrightarrow{S}$. The spin is an inherent property of protons and neutrons, and of nuclei with an odd number of these particles. Some of the nuclei which have a non-zero spin can be found in the human body, such as: ^1H, ^{13}C,^{23}Na, ^{31}P. Among them, Hydrogen is the most prevalent and therefore is the most adequate for MRI.

The nucleus of a ^1H hydrogen atom consists of a single proton, which has a positive electric charge. The spin of the proton generates an electrical current loop and hence a magnetic field. The proton has a magnetic moment parallel to the direction of its spin: $\overrightarrow{\mu} = \gamma \overrightarrow{S}$, where γ is the gyromagnetic ratio of the nucleus of interest ($\gamma_H = 267.513 \times 10^6 \text{rad.s}^{-1}.\text{T}^{-1}$). The gyromagnetic ratio γ is specific for each nuclear species.

Without a magnetic field, the orientation of the magnetic moment is random. When placed in a magnetic field $\overrightarrow{B_0}$, $\overrightarrow{\mu}$ tends to align with the direction of B_0 and, due to a resonance phenomenon, begins a precessional movement of the spinning axis around the direction of B_0. The angular velocity ω_0 – called *Larmor angular velocity* – of the precession movement is proportional to the amplitude of B_0 and the gyromagnetic ratio γ:

$$\omega_0 = \gamma B_0. \tag{A.1.1}$$

When a sample containing many protons is placed in a magnetic field $\overrightarrow{B_0}$, the individual protons tend to align with the direction of the field which results in a net magnetization $\overrightarrow{M}$ of the sample. However, due to thermal movements which prevent a complete alignment, the resulting magnetization vector is very weak. A NMR experiment aims at measuring $\overrightarrow{M}$.

Let us consider the rotating frame of reference (x, y, z) and assume $\overrightarrow{B_0} = B_0 \overrightarrow{e_z}$. We can decompose the net magnetization vector as: $\overrightarrow{M} = M_x \overrightarrow{e_x} + M_y \overrightarrow{e_y} + M_z \overrightarrow{e_z}$. The measured signal S is proportional to the transverse component: $S \propto \sqrt{M_x^2 + M_y^2}$. During an experiment in NMR, $\overrightarrow{M}$ is tilted away from its equilibrium position, using short pulses of radio-frequency (RF) radiation $\overrightarrow{B_1}$, perpendicular to $\overrightarrow{B_0}$, centered at the middle of the NMR spectrum. Applying $\overrightarrow{B_1}$ on $\overrightarrow{M}$ produces a magnetic moment

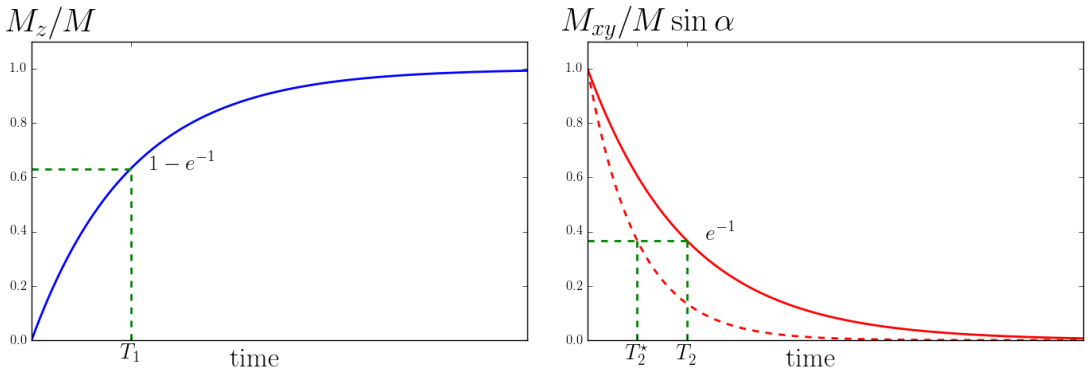

Figure A.1.1 – Longitudinal (left) and transverse (right) relaxations. The transverse component decreases with a time constant T_2 in a homogeneous field $\overrightarrow{B_0}$, and $T_2^\star$ in an inhomogeneous field.

which causes $\overrightarrow{M}$ to precess about $\overrightarrow{B_1}$ as well. This additional moment is: $\overrightarrow{m} = \overrightarrow{M} \times \overrightarrow{B_1}$. This results in a magnetization vector $\overrightarrow{M}$ which deviates from the z axis by a *flip angle* α. When the pulse ceases, the out-of-equilibrium magnetization vector returns to equilibrium state after a certain delay. This phenomenon is exponential and is characterized by two relaxation constants (cf. figure A.1.1):

- the *longitudinal magnetic* – or *spin-lattice – relaxation time* T_1 is relaxation constant of the longitudinal component M_z of the net magnetization vector returning to its initial state:
$$M_z(t) = M\left(1 - e^{-t/T_1}\right); \tag{A.1.2}$$
- the *transverse* – or *spin-spin – relaxation time* T_2 is relaxation constant of the transverse component M_{xy} of the net magnetization vector when returning to zero. It is the average time for the spins of precessing nuclei to dephase, due to mutual interactions:
$$M_{xy}(t) = M \sin(\alpha) e^{-t/T_2}. \tag{A.1.3}$$

The relaxation constants T_1 and T_2 depend the amplitude of the magnetic field $\overrightarrow{B_0}$ and on the physicochemical environment: the biological composition of the tissue of interest, as well as controlled parameters such as the temperature and the pressure. However in practice, the local field seen by the nuclei is not spatially homogeneous, due to $\overrightarrow{B_0}$ being non-homogeneous itself, but also to the chemical environment created by the biological tissue. This makes the resonant frequency of the spins vary continuously across space. The phase shift between the spins increasing with time, this results in an different exponential decrease in transverse magnetization, modeled by different relaxation time $T_2^\star < T_2$ (cf. figure A.1.1). The values of T_1, T_2 and $T_2^\star$ change signifi-

cantly depending on the tissue in which the protons are located, and are the source of the contrast in MRI.

The visibility of protons in MRI depend mainly on their relaxation time. Therefore, proton density reflects water and fat concentration in tissues. For example, a healthy muscle contains about 80% water. By choosing a sequence of RF pulses and adequate parameters and $\overrightarrow{B_0}$ gradients, one can have the different biological structures – e.g. muscle tissues and fat tissues – to take different intensities, allowing to separate them visually. For instance, a T_2-weighted sequence provides images in which the intensity is a increasing function of the T_2 value of the tissue in this location. There exist many sequences for acquiring T_1-weighted, T_2-weighted or $T_2^\star$-weighted images, in which the biological tissues are shown with different intensity distributions.

A.2 NMR IMAGING

In a volume, it is possible to make localized measurements of the amplitude of the components of $\overrightarrow{M}$, using magnetic field gradients and strategies involving certain Fourier space encoding, resulting in an multidimensional image (2D or 3D).

For the task of segmenting images of muscles, it is convenient to use. a choice of sequence and sequence parameters which reveals as much as possible the boundary of the muscles. The muscles are surrounded by a thin bio-membrane of connective tissue – the *fasciae* (singular: *fascia*) – and generally separated by a layer of fat tissues. Fasciae are made of collagen fibers, with the property of having an ultra-short T_2 which, in most typical modalities yields very little signal and show in black in images: if the transverse component of $\overrightarrow{M}$ is fully relaxed, the output signal is zero whatsoever.

Muscle and fat tissues have different and well know relaxation constants, and some methods have been developed to give them different intensities in images, for easy visual discrimination. For instance, some T_1-weighted images can be used for separating muscle and fat tissues, even thought the fasciae are difficult to see (cf. figure 1.1.1).

Some modalities reveal the muscle/fat transition more strongly than others. For example, *gradient-echo* sequences produce T_2^*-weighted images using the fact that the spin of fat protons (i.e. protons in fat-tissues) precess in a different rate than muscle (water) protons. In regions containing both kinds of protons, this causes a phase difference which results in a decreased amplitude of the transverse component at the acquisition time. In these images, muscle boundaries show a deep contrast (cf. figure 1.1.1). Among $T_2^\star$-weighted modalities, the Dixon method combines several acquisitions in order to provide separate images for fat-composed tissues and a water-composed tissues (muscle tissues and vessels). This method provides convenient material for segmenting muscles (cf. figure A.2.2):

- a map of muscle tissues only (vessels are negligible volume). We may use this *water-map* for easy separation of the muscles from fat tissues;

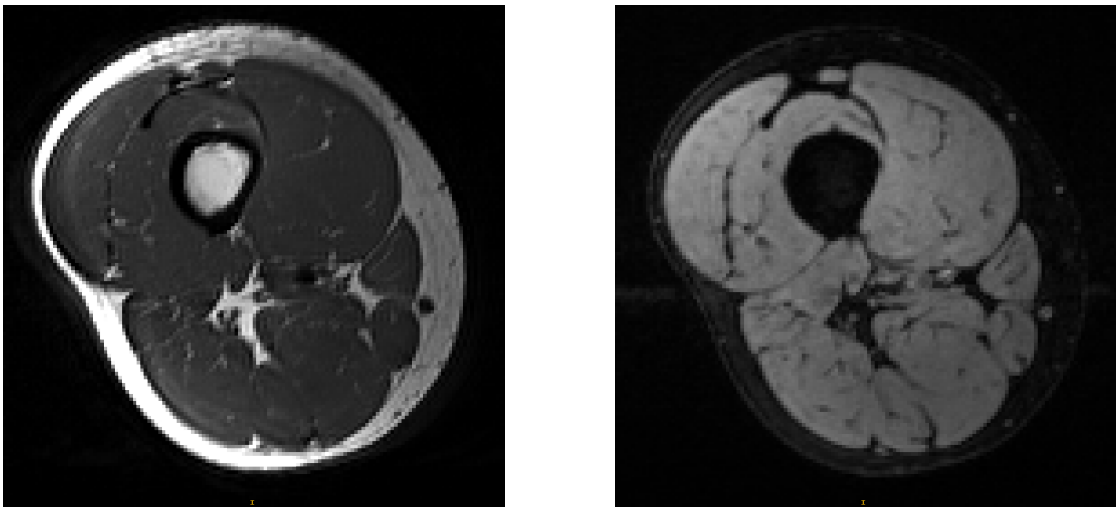

Figure A.2.1 – Axial-cross sections of thigh: (left) T_1-weighted; (right)$T_2^\star$-weighted with fat-suppression (a special method was applied during the acquisition of this volume to ignore signal from fat tissues).

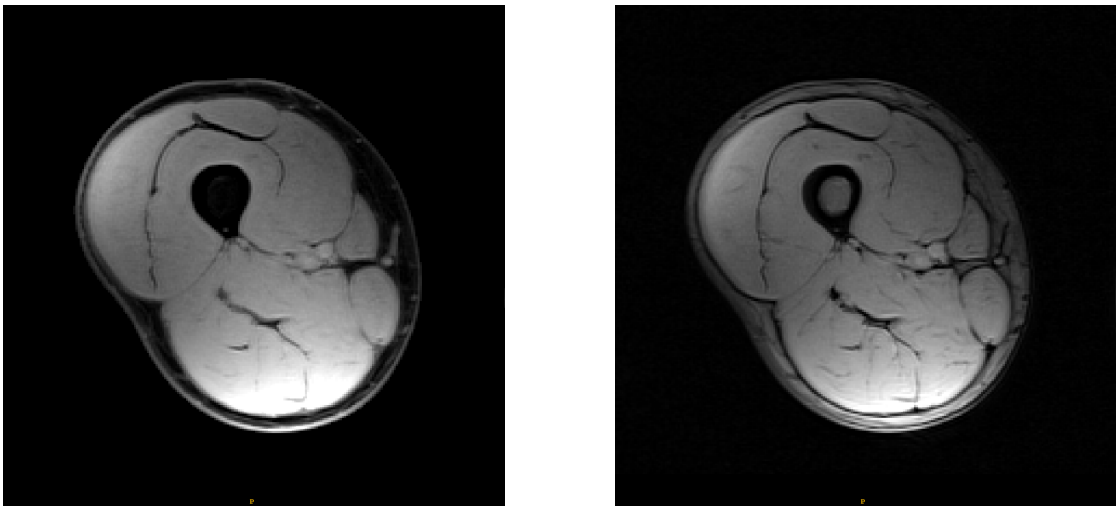

Figure A.2.2 – Dixon method, axial-cross sections of thigh: (left) water-map; (rig) opposed-phase image.

- an opposed-phase image, acquired are the instant when fat protons and water protons have opposed-phase spins, which enhances the boundaries. We will use this image in segmentation techniques relying on having contrasted boundaries for the objects of interest.

One should note that healthy muscles tissues cannot be discriminated from their appearance, since all muscles are made of the same biological tissue, which results in identical intensities and textures. The differences of intensities observable in the presented muscles are due to spatial inhomogeneities of the magnetic fields.

B

MATHEMATICAL APPENDICES

B.1 POSITIVE DEFINITENESS OF L_U

Proposition 1. *Assume there exists at least one path composed of nonzero-weight edges between each unknown node and one seed. Then L_U is positive definite:*

$$\forall \mathbf{x} \in \mathbb{R}^{|U|},\ \mathbf{x}^\top L_U \mathbf{x} > 0.$$

Proof. We recall the definition of L_U:

$$L_U(i,j) = \begin{cases} d_i & \text{if } i = j, \\ -w_{i,j} & \text{if } e_{ij} \in \mathcal{E}, \\ 0 & \text{otherwise,} \end{cases}$$

where $d_i = \sum_{j \in U \cup M} w_{ij}$. Consider the following:

$$\begin{aligned}
\forall x \in \mathbb{R}^{|U|},\ \mathbf{x}^\top L_U \mathbf{x} &= \sum_{i \in U} d_i x_i^2 - 2 \sum_{i,j \in U} w_{ij} x_i x_j, && \text{(B.1.1)} \\
&= \sum_{i \in U} \left(\sum_{j \in U \cup M} w_{ij} \right) x_i^2 - 2 \sum_{i,j \in U} w_{ij} x_i x_j, && \text{(B.1.2)} \\
&= \sum_{i \in U,\ j \in M} w_{ij} x_i^2 + && \text{(B.1.3)} \\
&\quad \left(\sum_{i,j \in U} w_{ij} x_i^2 - 2 \sum_{i,j \in U} w_{ij} x_i x_j \right), \\
&= \sum_{i \in U,\ j \in M} w_{ij} x_i^2 + \sum_{i,j \in U} w_{ij} \left(x_i - x_j \right)^2. && \text{(B.1.4)}
\end{aligned}$$

Using *Reductio ad absurdum*, suppose:

$$\exists x \in \mathbb{R}^{|U|} / \{\mathbf{0}\} \ \ s.t.\ \mathbf{x}^\top L_U \mathbf{x} = 0. \qquad \text{(B.1.5)}$$

For this hypothesis to stand, we must have:

$$\begin{aligned}
\forall i,j \in U,\ w_{ij} \left(x_i - x_j \right)^2 &= 0, && \text{(B.1.6)} \\
\forall (i,j) \in U \times M,\ w_{ij} x_i^2 &= 0. && \text{(B.1.7)}
\end{aligned}$$

Let us consider the index k of any nonzero entry of $\mathbf{x}$: $\forall k \in \{i \in U / x_i \neq 0\}$. There exists a path $\mathcal{P}_{kl} \subset \mathcal{E}$ of edges with a nonzero weight between node v_k and a marked node $v_l \in \mathcal{V}_M$. Then (a)$\forall e_{ij} \in \mathcal{P}_{kl} \cap (\mathcal{V}_U \times \mathcal{V}_U)$ equation B.1.6 gives: $x_i = x_j$; (b) $\forall e_{ij} \in \mathcal{P}_{kl} \cap (\mathcal{V}_U \times \mathcal{V}_M)$ equation B.1.7 gives: $x_i = 0$. Both results lead to $\forall k \in U,\ x_k = 0$, which contradicts the first hypothesis B.1.5. □

B.2 RANDOM WALKS AND PROBABILITY DISTRIBUTIONS

Proposition 2. *The Random Walks system 4.1.18 can only yield a solution* $\{\mathbf{x}^s\}_s$ *that is a probability distribution:*

$$\forall i, \sum_s x_i^s = 1, \tag{B.2.1}$$

$$\forall i, \forall s,\ x_i^s \geq 0. \tag{B.2.2}$$

Proof. We use the fact that L_U is positive definite and thus invertible, and that $\mathbf{1} = [1, \ldots, 1]^\top$ is a solution to $L\mathbf{x} = 0$ and thus

$$(L_U \mathbf{1}_U + B\mathbf{1}_M) = 0.$$

Consider:

$$L_U \mathbf{x}_U^s = -B\mathbf{x}_M^s, \tag{B.2.3}$$

$$\sum_s L_U \mathbf{x}_U^s = -\sum_s B\mathbf{x}_M^s, \tag{B.2.4}$$

$$L_U \left(\sum_s \mathbf{x}_U^s \right) = -B\mathbf{1}_M, \tag{B.2.5}$$

$$L_U \left(\sum_s \mathbf{x}_U^s \right) = L_U \mathbf{1}_U, \tag{B.2.6}$$

Since L_U is invertible, we have

$$\sum_s \mathbf{x}_U^s = \mathbf{1}_U.$$

In order to prove the non-negativity of $\mathbf{x}$, consider the RW objective and its gradient at location $\mathbf{0} = [0, \ldots, 0]^\top$:

$$E_{\mathrm{RW}}(\mathbf{x}_U^s) = \frac{1}{2} \mathbf{x}_U^{s\top} L \mathbf{x}_U^s + \mathbf{x}_U^{s\top} B \mathbf{x}_M^s + \mathrm{cst}, \tag{B.2.7}$$

$$\nabla E_{\mathrm{RW}}(\mathbf{0}) = B\mathbf{x}_M^s. \tag{B.2.8}$$

Since all entries of the gradient of $E_{\mathrm{RW}}(\cdot)$ at location $\mathbf{0}$ are non-positive:

$$(B\mathbf{x}_M^s)[i] = -\sum_{j\in M} w_{ij}x_j^s \leq 0, \tag{B.2.9}$$

we know that the direction of the solution is towards the positive numbers:

$$\forall i \in U, \forall s,\ x_i^s \geq 0.$$

□

B.3 RANDOM WALKS, SHAPE PRIOR AND PROBABILITY DISTRIBUTIONS

Proposition 3. *Assume the entries of $\{\mathbf{x}_0^s\}_s$ are probabilities. Then, the Random Walks system incorporating a prior knowledge term 4.2.10 can only yield a solution that is a probability:*

$$\forall i, \sum_s x_i^s = 1, \tag{B.3.1}$$

$$\forall i, \forall s,\ x_i^s \geq 0. \tag{B.3.2}$$

Proof. We use the fact that $(L+\Delta)$ is positive definite – and thus the solution to $(L+\Delta)\mathbf{x} = \mathbf{b}$ is unique – and that $\mathbf{1} = [1,\ldots,1]^T$ is a solution to $L\mathbf{x} = 0$. Consider:

$$(L+\Delta)\mathbf{x}^s = \Delta\mathbf{x}_0^s, \tag{B.3.3}$$

$$\sum_s (L+\Delta)\mathbf{x}^s = \Delta\left(\sum_s \mathbf{x}_0^s\right), \tag{B.3.4}$$

$$(L+\Delta)\left(\sum_s \mathbf{x}^s\right) = \Delta\mathbf{1}. \tag{B.3.5}$$

Since the solution is unique,

$$\sum_s \mathbf{x}^s = \mathbf{1}.$$

Considering the RW objective and its gradient at location $\mathbf{0} = [0,\ldots,0]^T$:

$$E_{\mathrm{RW+prior}}(\mathbf{x}^s) = \mathbf{x}^{sT}L\mathbf{x}^s + \|\mathbf{x}^s - \mathbf{x}_0^s\|_\Omega^2, \tag{B.3.6}$$

$$\nabla E_{\mathrm{RW+prior}}(\mathbf{0}) = -2\Omega\mathbf{x}_0^s. \tag{B.3.7}$$

Thus the entries of $\nabla E_{\mathrm{RW+prior}}(\mathbf{0})$ are non-positive and we know the direction of the solution is towards the positive numbers:

$$\forall i, \forall s,\ x_i^s \geq 0.$$

□

BIBLIOGRAPHY

Rolf Adams and Leanne Bischof. Seeded region growing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(6):641–647, 1994. (Cited on page 40.)

Salam A Al-Attar, Rebecca L Pollex, John F Robinson, Brooke A Miskie, Rhonda Walcarius, Brian K Rutt, and Robert A Hegele. Semi-automated segmentation and quantification of adipose tissue in calf and thigh by mri: a preliminary study in patients with monogenic metabolic syndrome. *BMC medical imaging*, 6(1):11, 2006. (Cited on page 5.)

Shawn Andrews, Ghassan Hamarneh, Azadeh Yazdanpanah, Bahareh HajGhanbari, and W. Darlene Reid. Probabilistic multi-shape segmentation of knee extensor and flexor muscles. In *MICCAI*, volume 14, pages 651–8. January 2011a. (Cited on pages 31, 37, 76, 77, and 78.)

Shawn Andrews, Chris McIntosh, and Ghassan Hamarneh. Convex Multi-Region Probabilistic Segmentation with Shape Prior in the Isometric Log-Ratio Transformation Space. *Medical Image Analysis*, pages 1–17, 2011b. (Cited on page 31.)

Eric Bardinet, Laurent D. Cohen, and Nicholas Ayache. Tracking and motion analysis of the left ventricle with deformable superquadrics. *Medical image analysis*, 1(2):129–49, June 1996. (Cited on page 20.)

Vincent Barra and Jean-Yves Boire. Segmentation of fat and muscle from MR images of the thigh by a possibilistic clustering algorithm. *Computer methods and programs in biomedicine*, 2002. (Cited on page 5.)

Pierre-Yves Baudin, N Azzabou, P G Carlier, and Nikos Paragios. Prior Knowledge, Random Walks and Human Skeletal Muscle Segmentation. In Nicholas Ayache, Hervé Delingette, Polina Golland, and Kensaku Mori, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012*, volume 7510 of *Lecture Notes in Computer Science*, pages 569–576. Springer, 2012a. ISBN 978-3-642-33414-6. (Cited on pages 8, 53, 77, and 90.)

Pierre-Yves Baudin, Noura Azzabou, Pierre Carlier, and Nikos Paragios. Manifold-enhanced Segmentation through Random Walks on Linear Subspace Priors. In *Proceedings of the British Machine Vision Conference*, pages 52.1–52.10. BMVA Press, 2012b. ISBN 1-901725-46-4. doi: http://dx.doi.org/10.5244/C.26.52. (Cited on pages 8 and 53.)

Pierre-Yves Baudin, Noura Azzabou, Pierre G Carlier, and Nikos Paragios. Automatic skeletal muscle segmentation through random walks and graph-based seed placement. pages 1036–1039, 2012c. (Cited on page 7.)

M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems*, 14:585–591, 2001. (Cited on page 62.)

Martin Bergtholdt, Daniel Cremers, and Christoph Schnörr. Variational Segmentation With Shape Priors. In *Handbook of Mathematical Models in Computer Vision*, pages 131–143. 2006. (Cited on page 16.)

D.P. Bertsekas. Nonlinear programming. 1999. (Cited on pages 88, 95, and 96.)

Ahmed Besbes. *Image Segmentation using MRFs and Statistical Shape Modeling*. PhD thesis, 2010. (Cited on page 28.)

Ahmed Besbes, Nikos Komodakis, Georg Langs, and Nikos Paragios. Shape priors and discrete mrfs for knowledge-based segmentation. pages 1295–1302, 2009. (Cited on page 28.)

Andrew Blake and Andrew Zisserman. *Visual reconstruction*. MIT Press, 1987. (Cited on page 15.)

M. Blaschko and C. Lampert. Learning to localize objects with structured output regression. *Computer Vision–ECCV 2008*, pages 2–15, 2008. (Cited on pages 87 and 89.)

Yuri Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *ICCV*, volume 1, pages 105–112. IEEE Comput. Soc, 2001. ISBN 0-7695-1143-0. doi: 10.1109/ICCV.2001.937505. (Cited on pages 8, 32, 35, and 40.)

Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE transactions on pattern analysis and machine intelligence*, 26(9):1124–37, September 2004. (Cited on page 33.)

Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 377–384 vol.1, 1999. (Cited on page 34.)

Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *International journal of computer vision*, 1997. (Cited on page 26.)

Tony F. Chan and Luminita A. Vese. Active contours without edges. *Image Processing, IEEE transactions on*, 10(2):266–77, January 2001. (Cited on pages 26 and 31.)

Guillaume Charpiat, Pierre Maurel, Jean-Philippe Pons, and Renaud Keriven. Generalized gradients: Priors on minimization flows. Technical Report March, 2007. (Cited on pages 19 and 20.)

Yunmei Chen, Hemant D. Tagare, and Sheshadri Thiruvenkadam. Using prior shapes in geometric active contours in a variational framework. *International Journal of Computer Vision*, 50(3):315–328, 2002. (Cited on page 27.)

Deepak Roy Chittajallu, Gerd Brunner, UUday Kurkure, Raja P. Yalamanchili, and Ioannis A. Kakadiaris. Fuzzy-Cuts: A knowledge-driven graph-based method for medical image segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (2009)*, pages 715–722, 2009. (Cited on page 36.)

Isaac Cohen and Laurent D. Cohen. A hybrid hyperquadric model for 2-d and 3-d data fitting. *Computer Vision and Image Understanding*, 1996. (Cited on page 21.)

Laurent D. Cohen. On active contour models and balloons. *CVGIP: Image understanding*, 53(2):211–218, 1991. (Cited on page 14.)

Laurent D. Cohen and Isaac Cohen. Finite Element Methods for Active Contour Models and Balloons for 2D and 3D Images. *Pattern analysis and machine intelligence, IEEE transactions on*, 15(11):1131–1147, 1993. (Cited on page 14.)

Timothy F. Cootes, Christopher J. Taylor, David H. Cooper, and Jim Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995. (Cited on pages 21 and 22.)

Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active Appearance Models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):681–685, 2001. (Cited on page 21.)

Daniel Cremers, Florian Tischh Auser, and Joachim Weickert. Diffusion Snakes: Introducing Statistical Shape Knowledge into the Mumford-Shah Functional. *International Journal of Computer Vision*, 50(3):295–313, 2002. (Cited on pages 16 and 22.)

Daniel Cremers, Timo Kohlberger, and Christoph Schnörr. Shape statistics in kernel space for variational image segmentation. *Pattern Recognition*, 36(9):1929–1943, September 2003. (Cited on page 23.)

Daniel Cremers, Stanley J. Osher, and Stefano Soatto. Kernel Density Estimation and Intrinsic Alignment for Shape Priors in Level Set Segmentation. *International Journal of Computer Vision*, 69(3):335–351, May 2006a. (Cited on pages 23 and 27.)

Daniel Cremers, Mikael Rousson, and Rachid Deriche. A Review of Statistical Approaches to Level Set Segmentation: Integrating Color, Texture, Motion and Shape.

International Journal of Computer Vision, 72(2):195–215, August 2006b. (Cited on pages 13, 25, and 26.)

Daniel Cremers, Frank R. Schmidt, and Frank Barthel. Shape priors in variational image segmentation: Convexity, Lipschitz continuity and globally optimal solutions. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, June 2008. (Cited on page 30.)

H. Delingette. *Modélisation, Déformation et Reconnaissance d'objets tridimensionnels a l'aide de maillages simplexes*. Thèse de sciences, Ecole Centrale de Paris, July 1994. (Cited on page 17.)

H. Delingette. General object reconstruction based on simplex meshes. *International Journal of Computer Vision*, 32(2):111–146, September 1999. (Cited on pages 15 and 17.)

Andrew Delong and Yuri Boykov. Globally optimal segmentation of multi-region objects. *Computer Vision, 2009 IEEE 12th . . .*, (October):1–8, 2009. (Cited on page 36.)

Alain Dervieux and François Thomasset. A finite element method for the simulation of a rayleigh-taylor instability. In *Approximation methods for Navier-Stokes problems*, pages 145–158. Springer, 1980. (Cited on page 25.)

Abouzar Eslami, Athanasios Karamalis, Amin Katouzian, and Nassir Navab. Segmentation by retrieval with guided random walks: Application to left ventricle segmentation in mri. *Medical image analysis*, 2012. (Cited on page 84.)

Salma Essafi, Georg Langs, and Nikos Paragios. Hierarchical 3D diffusion wavelet shape priors. In *ICCV*, pages 1717–1724. IEEE, September 2009. ISBN 978-1-4244-4420-5. doi: 10.1109/ICCV.2009.5459385. (Cited on pages 23 and 37.)

Jianping Fan, David KY Yau, Ahmed K Elmagarmid, and Walid G Aref. Automatic image segmentation by integrating color-edge extraction and seeded region growing. *Image Processing, IEEE Transactions on*, 10(10):1454–1466, 2001. (Cited on page 40.)

P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. (Cited on page 89.)

J.L. Fleckenstein, J.V. Crues, and C.D. Reimers. *Muscle imaging in health and disease*, volume 208. Springer, 1996. (Cited on pages 1 and 109.)

Daniel Freedman and Tao Zhang. Interactive Graph Cut Based Segmentation with Shape Priors. In *CVPR*, volume 1, pages 755–762. Ieee, 2005. ISBN 0-7695-2372-2. doi: 10.1109/CVPR.2005.191. (Cited on page 36.)

Benjamin Gilles and Nadia Magnenat-Thalmann. Musculoskeletal MRI segmentation using multi-resolution simplex meshes with medial representations. *Medical image analysis*, 14(3):291–302, June 2010. (Cited on pages 17 and 37.)

Benjamin Gilles and Dinesh K. Pai. Fast musculoskeletal registration based on shape matching. In *MICCAI*, volume 5242 of *Lecture Notes in Computer Science*, pages 822–829. January 2008. (Cited on pages 18, 19, 37, 75, 77, and 78.)

Bram Van Ginneken, Mikkel B. Stegmann, and Marco Loog. Segmentation of anatomical structures in chest radiographs using supervised methods: a comparative study on a public database. *Medical Image Analysis*, 2006. (Cited on page 21.)

Ben Glocker, Nikos Komodakis, Georgios Tziritas, Nassir Navab, and Nikos Paragios. Dense image registration through MRFs and efficient linear programming. *Medical image analysis*, 12(6):731–41, December 2008. (Cited on page 70.)

Leo Grady. Multilabel Random Walker Image Segmentation Using Prior Models. In *CVPR*, volume 1, pages 763–770. Ieee, 2005. ISBN 0-7695-2372-2. doi: 10.1109/CVPR.2005.239. (Cited on pages 7, 63, 64, 90, 99, 105, and 107.)

Leo Grady. Random walks for image segmentation. *Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, 2006. (Cited on pages 6, 39, 53, 90, and 105.)

L. Hagen and A.B. Kahng. New spectral methods for ratio cut partitioning and clustering. *Computer-aided design of integrated circuits and systems, ieee transactions on*, 11 (9):1074–1085, 1992. (Cited on pages 60 and 61.)

J. Ham, D.D. Lee, S. Mika, and B. Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the twenty-first international conference on Machine learning*, page 47. ACM, 2004. (Cited on page 62.)

Andrew J. Hanson. Hyperquadrics: Smoothly deformable shapes with convex polyhedral bounds. *Computer Vision, Graphics, and Image Processing*, 1988. (Cited on page 21.)

J.A. Hartigan and M.A. Wong. Algorithm as 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108, 1979. (Cited on page 60.)

Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27–59, May 2009. (Cited on pages 87, 89, and 93.)

Michael Kass, Andrew Witkin, and Dimetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988. (Cited on pages 13 and 14.)

Satyanad Kichenassamy, Arun Kumar, Peter Olver, Allen Tannenbaum, and Anthony Jr. Yezzi. Gradient flows and geometric active contour models. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 810–815. IEEE Comput. Soc. Press, 1995. ISBN 0-8186-7042-8. doi: 10.1109/ICCV.1995.466855. (Cited on page 26.)

Junmo Kim, John W Fisher III, Anthony Jr Yezzi, Mujdat Cetin, and Alan S Willsky. Non parametric methods for image segmentation using information theory and curve evolution. *Image Processing, IEEE Transactions on*, 14(10):9–12, 2002. (Cited on page 27.)

Jon Kleinberg. The small-world phenomenon: an algorithm perspective. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 163–170. ACM, 2000. (Cited on page 106.)

Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1568–83, October 2006. (Cited on pages 7, 40, and 47.)

Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE transactions on pattern analysis and machine intelligence*, 26(2):147–59, February 2004. (Cited on page 34.)

Nikos Komodakis. Efficient training for pairwise or higher order crfs via dual decomposition. pages 1841–1848, 2011. (Cited on pages 29 and 92.)

Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. MRF Optimization via Dual Decomposition: Message-Passing Revisited. *2007 IEEE 11th International Conference on Computer Vision*, (2):1–8, 2007. (Cited on pages 28, 34, 88, 95, and 96.)

M.P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. *Advances in Neural Information Processing Systems*, 23:1189–1197, 2010. (Cited on page 89.)

M.P. Kumar, H. Turki, D. Preston, and D. Koller. Learning specific-class segmentation from diverse data. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1800–1807. IEEE, 2011. (Cited on page 89.)

J. Lafferty, A. McCallum, and F.C.N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001. (Cited on page 87.)

Georg Langs, Nikos Paragios, Laboratoire Mas, Ecole Centrale De Paris, Equipe Galen, and Inria Saclay. Modeling the structure of multivariate manifolds: Shape Maps. 2008. (Cited on page 24.)

Bertrand Leroy, Isabelle L. Herlin, and Laurent D. Cohen. Multi-resolution algorithms for active contour models. In Marie-Odile Berger, Rachid Deriche, Isabelle Herlin, Jérome Jaffré, and Jean-Michel Morel, editors, *ICAOS'96*, volume 219 of *Lecture Notes in Control and Information Sciences*, pages 58–65. Springer Berlin / Heidelberg, 1996. ISBN 978-3-540-76076-4. (Cited on page 14.)

Michael E. Leventon, W. Eric Grimson, and Olivier Faugeras. Statistical shape influence in geodesic active contours. In *Computer Vision and Pattern Recognition*, volume 1, pages 316–323, 2000. (Cited on page 27.)

Jean-Christophe Lombardo, Claude Puech, et al. Oriented particles: A tool for shape memory objects modelling. In *Graphics Interface'95*, pages 255–262, 1995. (Cited on page 14.)

F Maier, A Wimmer, G Soza, JN Kaftan, D Fritz, and R Dillmann. Automatic liver segmentation using the random walker algorithm. In *Bildverarbeitung für die Medizin 2008*, pages 56–61. Springer, 2008. (Cited on page 40.)

JP Mattei, Y Le Fur, N Cuge, S Guis, PJ Cozzone, and D Bendahan. Segmentation of fascias, fat and muscle from magnetic resonance images in humans: the dispimag software. *Magnetic Resonance Materials in Physics, Biology and Medicine*, 19(5):275–279, 2006. (Cited on page 5.)

Tim McInerney and Demetri Terzopoulos. Deformable models in medical image analysis: a survey. *Medical image analysis*, 1(2):91–108, June 1996. (Cited on pages 13 and 14.)

M. Meila and J. Shi. A random walks view of spectral segmentation. 2001. (Cited on pages 60 and 61.)

Branislav Mičušík and Allan Hanbury. Automatic image segmentation by positioning a seed. In *Computer Vision–ECCV 2006*, pages 468–480. Springer, 2006. (Cited on page 40.)

Johan Montagnat, Hervé Delingette, and Nicolas Ayache. A review of deformable surfaces: topology, geometry and deformation. *Image and Vision Computing*, 19(2001): 1023–1040, 2004. (Cited on page 13.)

David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989. (Cited on page 15.)

Chahab Nastar and Nicholas Ayache. Fast segmentation, tracking, and analysis of deformable objects. In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 275–279. IEEE, 1993. (Cited on pages 14 and 16.)

A.Y. Ng, M.I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002. (Cited on page 60.)

Stanley J. Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of computational physics*, 79(1988):12–49, 1988. (Cited on page 25.)

Sébastien Ourselin, Alexis Roche, Sylvain Prima, and Nicholas Ayache. Block matching: A general framework to improve robustness of rigid registration of medical images. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2000*, pages 557–566. Springer, 2000. (Cited on page 41.)

Nikos Paragios. Curve propagation, level set methods and grouping. In Nikos Paragios, Yunmei Chen, and Olivier Faugeras, editors, *Handbook of Mathematical Models in Computer Vision*, pages 145–159. Springer, 2005. (Cited on page 25.)

Nikos Paragios and Rachid Deriche. A PDE-based level-set approach for detection and tracking of moving objects. In *Computer Vision, 1998. Sixth International Conference on*, pages 1139–1145, 1998. (Cited on page 26.)

Nikos Paragios and Rachid Deriche. Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects. *Pattern analysis and machine intelligence, IEEE transactions on*, 22(3):266–280, 2004. (Cited on page 27.)

Nikos Paragios, Yunmei Chen, and Olivier Faugeras, editors. *Handbook of mathematical models in computer vision*. Springer, 2005. (Cited on page 13.)

Stephen M Pizer, P Thomas Fletcher, Sarang Joshi, Andrew Thall, James Z Chen, Yonatan Fridman, Daniel S Fritsch, A Graham Gash, John M Glotzer, Michael R Jiroutek, et al. Deformable m-reps for 3d medical image segmentation. *International Journal of Computer Vision*, 55(2-3):85–106, 2003. (Cited on page 18.)

J-P Pons and J-D Boissonnat. Delaunay deformable models: Topology-adaptive meshes based on the restricted delaunay triangulation. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. (Cited on page 15.)

Sami Romdhani, Shaogang Gong, and Alexandra Psarrou. A Multi-View Nonlinear Active Shape Model Using Kernel PCA. In *Proceedings of BMVC*, number i, pages 483–492, 1999. (Cited on pages 21 and 23.)

Mikael Rousson. A Variational Framework for Active and Adaptative Segmentation of Vector Valued Images. In *Motion and Video Computing, 2002. Proceedings. Workshop on*, pages 56–61, 2002. (Cited on page 27.)

Mikael Rousson and Daniel Cremers. Efficient kernel density estimation of shape and intensity priors for level set segmentation. In Guido Duncan, James and Gerig,

editor, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2005*, volume 3750 of *Lecture Notes in Computer Science*, pages 757–764. Springer Berlin / Heidelberg, January 2005. (Cited on page 27.)

Mikael Rousson and Nikos Paragios. Shape priors for level set representations. In *Computer Vision - ECCV 2002*, volume 2351 of *Lecture Notes in Computer Science*, pages 78–92. Springer Berlin / Heidelberg, 2002. (Cited on page 27.)

M. Saerens, F. Fouss, L. Yen, and P. Dupont. The principal components analysis of a graph, and its relationships to spectral clustering. *Machine Learning: ECML 2004*, pages 371–383, 2004. (Cited on page 62.)

James A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences of the United States of America*, 93(4):1591–5, February 1996. (Cited on page 45.)

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th international conference on Machine learning*, pages 807–814. ACM, 2007. (Cited on pages 87, 89, and 93.)

J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000a. (Cited on pages 60, 61, and 62.)

Jianbo Shi and Jitendra Malik. Normalized Cuts and Image Segmentation Normalized Cuts and Image Segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000b. (Cited on pages 32, 34, 35, and 60.)

Ali Kemmal Sinop and Leo Grady. A Seeded Image Segmentation Framework Unifying Graph Cuts And Random Walker Which Yields A New Algorithm. *Computer Vision, 2007. ICCV 2007. IEEE . . .*, 2007. (Cited on page 35.)

A. Smola, SVN Vishwanathan, and T. Hoffman. Kernel methods for missing variables. In *AISTATS*, 2005. (Cited on page 89.)

Richard Szeliski and David Tonnesen. *Surface modeling with oriented particle systems*, volume 26. ACM, 1992. (Cited on page 14.)

Martin Szummer, Pushmeet Kohli, and Derek Hoiem. Learning CRFs using graph cuts. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Computer Vision ECCV 2008*, volume 5303 of *Lecture Notes in Computer Science*, pages 582–595. Springer Berlin / Heidelberg, 2008. ISBN 978-3-540-88685-3. doi: 10.1007/978-3-540-88688-4_43. (Cited on pages 8, 87, 89, and 92.)

Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin markov networks. *Neural Information Processing Systems*, 16(im):94–104, 2003. (Cited on pages 87, 88, 89, and 93.)

Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, volume 21, pages 205–214, 1987. ISBN 0897912276. (Cited on pages 16 and 19.)

Demetri Terzopoulos, Andrew Witkin, and Michael Kass. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36:91–123, 1988. (Cited on page 14.)

Dimetri Terzopoulos and Dimitris Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. *Computer Vision, 1990. ...*, 1990. (Cited on page 20.)

Andy Tsai, Anthony Jr Yezzi, William III Wells, Clare Tempany, Duwey Tucker, Ayres Fan, W Eric Grimson, and Alan Willsky. Model-based curve evolution technique for image segmentation. In *Computer Vision and Pattern Recognition. 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 00, page 463, 2001. ISBN 0769512720. (Cited on page 27.)

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. *Twenty-first international conference on Machine learning - ICML '04*, page 104, 2004. (Cited on pages 87, 88, 89, and 93.)

Carole J. Twining and Christopher J. Taylor. Kernel Principal Component Analysis and the Construction of Non-Linear Active Shape Models. In *Proceedings of BMVC*, pages 23–32, 2001. (Cited on page 23.)

M. Alex O. Vasilescu and D. Terzopoulos. Adaptive meshes and shells: irregular triangulation, discontinuities, and hierarchical subdivision. *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 829–832, 1992. (Cited on page 16.)

Olga Veksler. Star shape prior for graph-cut image segmentation. *Computer Vision–ECCV 2008*, pages 454–467, 2008. (Cited on page 36.)

U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007. (Cited on pages 58, 60, and 61.)

Chaohui Wang, Olivier Teboul, Fabrice Michel, Salma Essafi, and Nikos Paragios. 3D knowledge-based segmentation using pose-invariant higher-order graphs. In *MICCAI*, volume 6363 of *Lecture Notes in Computer Science*, pages 189–196. 2010. (Cited on pages 29 and 37.)

Paul Wighton, Maryam Sadeghi, Tim K Lee, and M Stella Atkins. A fully automatic random walker segmentation for skin lesions in a supervised setting. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009*, pages 1108–1115. Springer, 2009. (Cited on page 40.)

Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1993. (Cited on pages 32 and 34.)

CL Wyatt, Y Ge, and DJ Vining. Automatic segmentation of the colon for virtual colonoscopy. *Computerized medical imaging and graphics*, 24(1):1–9, 2000. (Cited on page 40.)

Chenyang Xu. Snakes, shapes, and gradient vector flow. *Image Processing, IEEE Transactions on*, 7(3):359–69, January 1998. (Cited on page 14.)

Anthony Jr. Yezzi, Andy Tsai, and Alan Willsky. A statistical approach to snakes for bimodal and trimodal imagery. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, pages 898–903, 1999. (Cited on page 27.)

Chun-Nam John Yu and Thorsten Joachims. Learning structural SVMs with latent variables. *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, pages 1–8, 2009. (Cited on pages 88, 89, and 93.)

A.L. Yuille, A. Rangarajan, and AL Yuille. The concave-convex procedure (cccp). *Advances in Neural Information Processing Systems*, 2:1033–1040, 2002. (Cited on pages 88, 89, and 93.)

Alan L Yuille. Deformable Templates for Face Recognition. 3(1), 1990. (Cited on page 20.)

Paul A. Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C. Gee, and Guido Gerig. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006. (Cited on page 105.)

Hong-Kai Zhao, Tony F. Chan, Barry Merriman, and Stanley J. Osher. A variational level set approach to multiphase motion. *Journal of computational*, 195:179–195, 1996. (Cited on page 27.)

Song C. Zhu and Alan Yuille. Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *Pattern analysis and machine intelligence, IEEE transactions on*, 18(9), 1996. (Cited on page 15.)