UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

WAGNER GONÇALVES RAMPON

Patient Specific Mesh Generation

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Prof. Dr. Anderson Maciel
Advisor

Prof. Dr. João L. D. Comba
Coadvisor

Porto Alegre, March 2016

CIP – CATALOGING-IN-PUBLICATION

Rampon, Wagner Gonçalves

Patient Specific Mesh Generation / Wagner Gonçalves Rampon. – Porto Alegre: PPGC da UFRGS, 2016.

60 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2016. Advisor: Anderson Maciel; Coadvisor: João L. D. Comba.

1. Mesh generation. 2. Volume segmentation. 3. Surgery planning. 4. Mesh deformation. 5. Laplacian method. I. Maciel, Anderson. II. Comba, João L. D.. III. Title.

”So much universe, and so little time""
— TERRY PRATCHETT

ACKNOWLEDGEMENTS

First of all, I would like to thank deeply to my advisor, Anderson Maciel, for giving me the chance of working with him and supporting me through these years. He offered guidance to me, a completely inexperienced researcher, with clever suggestions and true commitment, believing in what we were doing and motivating me. Because of that opportunity, I got to know world of research. I would also like to give a special thank my co-advisor João Comba, which always had insightful comments and a vast experience in works of excellent quality. I consider myself very lucky for the chance of working with such great persons.

I also would like to thank the Federal University of Rio Grande do Sul (UFRGS) and the Institute of Informatics (INF), for both the great infrastructure and funding, which made this research possible.

Through these years, I also received a lot of support from various different people, and close friends. I would like to give special thanks to Bernardo Henz for his almost infinite patience in explaining complex contents in such an easy and understandable way, and for his friendship and support through stressful days. A gigantic thanks to Victor Adriel, who helped me in all possible ways and put up with my complaining and bad mood. I am also very grateful to Lenna Maule for all the out of the box ideas, constant motivation, and even some codding lessons. I also acknowledge Gerson Groth for the best chimarrão I could wish for, and for his friendship through these years. I was very lucky indeed, to also have made so many good friends and work with them on a daily basis.

I would like to give the biggest of all thanks to my family for their support and love, specially my mother. To all my friends, thank you for putting up with me not being as available, for your words of support and your eagerness to see me finishing the research. And finally, I want to acknowledge all my colleagues in the Visualization, Interaction and Computer Graphics Lab. You are the best company I could have wished for, and I wish the best to all of you. Thank you.

ABSTRACT

This work presents a study about medical-volume segmentation and a solution to generate patient-specific meshes to use in patient-specific surgery simulations. Patient-specific meshes are useful assets for surgery planning and to allow better visualization of certain pathological conditions of a given patient, which are not obtainable by artistically designed meshes. We analyzed what are the complications to obtain a patient-specific mesh using only standard medical imagery exams. For that, we reviewed several medical volume segmentation techniques. It led us to define the problems within the existing techniques and to develop a method that does not suffer from these problems, with the least possible user interaction or relying on any other data other then the patient exam. Our target for obtaining specific meshes were soft tissue organs, which are a specially complicated case due to various issues related to the medical images and human anatomy. This is accomplished by geometrical operations over special meshes that deform until achieving the shape of the desired organ. Results show that our technique was able to obtain patient-specific meshes from medical images with superior quality than algorithms of the same class. Thanks to the simplicity of the developed approach, its also easy to implement and to reproduce our obtained results.

Keywords: Mesh generation, volume segmentation, surgery planning, mesh deformation, laplacian method.

Geração de Malhas para Pacientes Específicos

RESUMO

Este trabalho apresenta um estudo sobre segmentação de volumes médicos e uma solução para se obter malhas poligonais de pacientes específicos para uso em simulações de cirurgia. Malhas de pacientes específicos são importantes para planejamento de intervenções cirúrgicas e permitem uma melhor visualização de condições patológicas em um paciente, coisa não obtível em malhas geradas artisticamente. Nós analisamos quais são os fatores complicantes para se obter estas malhas de um paciente específico usando apenas imagens médicas obtidas em exames padrões. Para isso, nós revisamos diversos métodos existentes para segmentação de volumes médicos. Isso nos levou a definir os problemas com as técnicas existentes, e a desenvolver um método que não sofra destes problemas, utilizando pouca interação humana e não tendo dependências de mais dados que não o exame do paciente. Nosso alvo para obter malhas especificas foram órgãos de tecido mole, que são um caso especialmente complicado da área, graças a várias questões relacionadas às imagens médicas e à anatomia humana. Atacamos esse problema aplicando modificações geométricas em malhas especiais, que deformam até atingir a forma dos órgãos que se deseja segmentar. Os resultados mostram que nossa técnica conseguiu obter malhas específicas de pacientes a partir de volumes médicos com qualidade superior a de outros algoritmos de mesma classe. Graças a simplicidade do método desenvolvido, nossos resultados são facilmente implementáveis e reproduzidos.

Palavras-chave: segmentação de volumes, planejamento de cirurgia, malhas deformáveis, método laplaciano.

LIST OF ABBREVIATIONS AND ACRONYMS

PSVR	Patient-Specific Virtual Reality Simulation
CT	Computed Tomography
MRI	Magnetic Resonance Imaging
ASM	Active Shape Model
SSM	Statistical Shape Model
AAM	Active Appearance Model

LIST OF FIGURES

TABLE OF CONTENTS

1 INTRODUCTION

The report from the Institute of Medicine, "To err is Human" (KOHN et al., 1999), published more than a decade ago, raises awareness to the high prevalence of medical errors in modern healthcare and the impact on patient safety. Although these errors emerge from various different causes, a significant subset resulted from invasive diagnostic and therapeutic interventions. Among different efforts to address and improve patient safety by minimizing procedural complications (AMERICA, 2001), surgery simulators became essential for surgery training. Reports show that the use of general VR simulations allow the acquisition, improvement and sustenance of complex skills without the risk of causing any harm to a real patient. Simulation research has also shown that these competences can transfer to real operations in a variety of specialties such as laparoscopy or endoscopy (SEYMOUR et al., 2002).

More recently, patient-specific virtual reality simulation (PSVR) has become possible thanks to advances especially in the field of image processing, allowing the incorporation of patient-specific imagery (such as CT or MRI data) in these simulations (WILLAERT et al., 2010). However, most simulation environments still use default anatomic models to represent the patient's organs, especially with soft tissues. This simplification is justified by the difficulties involved in generating a triangle mesh from the medical imaging exams. Therefore, these simulators are not suitable for surgery planning, as those require patient-specific data.

1.1 Problem

The difficulties of obtaining patient-specific meshes emerge from the limitations of the standard information acquisition techniques, for instance, computed tomography (CT) or Magnetic Resonance Imaging (MRI). These techniques yield a series of 2D image slices that, when stacked, compose a volume. The smallest fragment of information of a medical volume is called a *voxel*. Voxels' intensities vary according to the type of tissue and acquisition modality. Generally, these are the only information available to obtain a patient-specific model.

Obtaining patient-specific meshes from 3D image exams has been seen as a two-fold problem: segmentation of these medical images and mesh generation. While efficient solutions exist for the latter (LORENSEN; CLINE, 1987), medical volume segmentation (especially when soft tissues are involved) is still a widely researched topic. There are different reasons for volume segmentation being a difficult problem. Some organs boundaries may be indistinct given the similarity in intensities of the voxels. There is also the problem that intensities inside the same organ are often inhomogeneous thanks to cysts, blood vessels, or different stages of diseases. These problems are visible in Fig. 1.1. In-

Figure 1.1: A CT scan of a region with soft tissues. The blue square indicates the boundary region between liver and muscle. Notice that intensities belonging to the liver are indistinguishable from those of the muscle. The red square indicates a region within the liver, and the different inhomogeneous intensities.

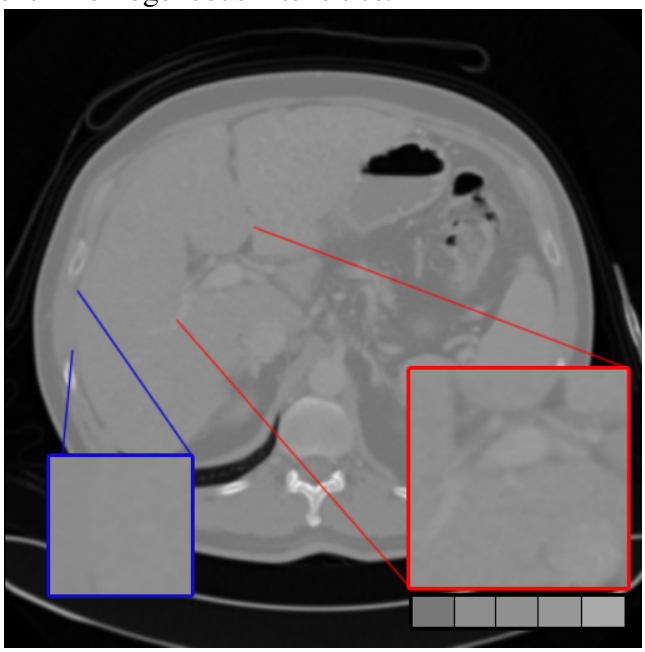

Figure 1.2: Source: original image

distinct boundaries and inhomogeneous intensities prevent the target organ from being easily captured without leakage problems (when another organ is partially selected as part of the target) or converging too far from the expected boundary of the organ (inhomogeneous intensities creating a convergence parameter inside the organ). Soft tissues also may present complex shapes, and vary greatly from one patient to another, which increases the difficulty of obtaining a patient-specific mesh.

There are different techniques for volume segmentation (watershed, graph cuts, level-set, among others). Each of these type of algorithms has specific properties and are prone to different problems. More details on volume-segmentation algorithms are available in Chapter. 2.

1.2 Thesis Objective

In this thesis, we present a unified solution for volume segmentation and mesh generation to obtain patient-specific meshes. Our technique overcomes problems of volume-segmentation algorithms and allows us to obtain plausible meshes that are very similar to the organ shape. We tackle the segmentation problem fitting multiple spherical meshes to the boundary region of the target organ in a medical volume.

This approach is based on the assumption that the initial spheres and the final organ meshes are homeomorphic. In such a way, the spheres can be placed somewhere inside the foreground object (i.e., the target organ) and dilated (or inflated) towards the target boundaries to eventually represent its shape.

The inflation process is iterative: vertices search for *feature voxels*, and are displaced by a fraction of the distance towards that target at each iteration. A *feature voxel* can

be defined by properties in the volume such as target intensity (DORNHEIM et al., 2006), gradient magnitude (KASS; WITKIN; TERZOPOULOS, 1988), gaussian mixture (HABAS et al., 2008), intensity distribution (KAINMÜLLER; LANGE; LAMECKER, 2007), among others. Our technique is generic and works with any combination of these.

Vertices without a correspondent boundary region, however, have their positions given by geometric constraints that use Laplacian preservation. We also use modified geometrical constraints which are easily integrated in the deformation framework to avoid artifacts, e.g. self-intersection of the mesh, while preserving sharp features of the volume. The multiple seed-points for the deformation process allow for a better coverage of the volume, which solves the problem of undersampled regions. The algorithm convergence can be either defined by a number of iterations or by the average displacement of a vertex during an iteration.

Our method is robust, finding plausible organ boundaries even when they are not visible to the eye or state of the art computer vision. It outputs meshes with improved overall sampling quality, enabling the extraction of irregular features in the anatomy while avoiding self-intersection of the mesh triangles.

1.3 Thesis Organization

In Chapter 2 we review the different volume segmentation techniques, with their strengths and weaknesses. Our technique, that overcomes these weaknesses is presented in detail in Chapter 3. In Chapter 4 we experimentally demonstrate the quality of our method in comparison to previous approaches and show that our technique outputs models closer to the real features of the patient's organs. Conclusions are drawn in Chapter 5.

2 RELATED WORK

Segmentation of medical images is a relevant topic in computer vision. Methods generally fall in one of these four categories: manual, interactive, semi-automatic or automatic. This discussion is presented on Chapter 2.1. Changing certain aspects of a method may cause it to switch categories. So, we also provide two other classifications, regarding how the problem is approached, independent of how interaction is performed: Local feature methods, presented in Chapter 2.2, and methods based on deformable models, presented in Chapter 2.3.

2.1 Interaction Modes for Segmentation Algorithms

Current medical image segmentation methods are in one of the following categories: manual, interactive, semi-automatic and fully automatic.

2.1.1 Manual Segmentation

Methods in this class are often accurate but very time-consuming. They require trained experts to draw contours by hand on the computer screen. Given that the technique relies on human understanding of organ boundaries and structures, manual methods are prone to wide variability within a same model. For instance, an inter-observer variability of 14-22% measured in disagreement ratio was reported (KAUS et al., 1999). For that reason, when aiming for the ground-truth for medical image segmentation, the average of various manual segmentations is used. These methods are very time consuming, being reported to take two to four hours for 1500-2000 images of 512x512 pixels to be manually segmented (STRAKA et al., 2003).

2.1.2 Fully-Automatic

These segmentation methods do not rely on any human labour. Instead, they use statistics and training sets, among other techniques, to initialize and perform segmentation (OKADA et al., 2008). This leads to repeatable results. Currently, they are not reliable enough for clinical usage. This is due to several factors, such as large intensity variation for a same target tissue across different patients. These may happen due to different acquisition methods for the medical images, or different tissues properties among patients or even different stages of diseases. Signal to noise ratio may also vary from image to image, requiring different pre-processing steps to normalize. Finally, large shape variance of the same target tissue (which may be due to natural causes or diseases) can also hinder the process of automatic segmentation. Usually, these techniques are applied to cases were a high contrast between background and foreground objects is obtainable,

such as brain and cranial segmentation, or bones in CT scan. For all these reasons, this class of methods are not as robust as necessary on clinical usage.

2.1.3 Interactive

Interactive methods require the user to give constant input, in a interactive fashion during the segmentation process. This can be seen as an manual method, but greatly aided by the computer. Typical scenarios include the user creating crude strokes around the organ as input for the algorithm to opperate. These strokes are not necessary on every image of the volume, only on certain key slices. If a segmentation result of one slice is not satisfactory, new strokes or parameter values can be used (or modifications of previous segmentation results) to improve the accuracy. Usually, these methods are time-efficient and are adopted clinically. The accuracy is generally proportional to how much interaction is involved. Commercial products like PathFinder and IntraSense use techniques that are interactive for segmentation. Usually, retouch stages are incorporated so the user can correct minor imperfections in the segmentation. These stages include expanding undersegmented regions or removing segmented areas which are not part of the target.

2.1.4 Semi-Automatic

Methods classified as semi-automatic incorporate minimal user input, usually as initialization parameters and parameter values, and are the methods which present the best trade-off as to produce clinically acceptable results, requiring less human labour than Interactive methods. The method presented in this thesis can be categorized as a Semi-Automatic technique.

2.2 Local-feature based segmentation

In this section we will present different techniques for medical image segmentation which rely solely on local features. These are methods that generally use local information to perform segmentation, not taking into account global information of the target organ. Usually, they are computationally cheaper and not as difficult to implement than deformable models.

2.2.1 Thresholding

This technique can be defined as categorizing pixels below a certain intensity value t into one group, and the rest of the pixels in another group. The result of thresholding a medical image can be seen on Fig. 2.1. Given that medical images generally do not have bimodal intensities distribution, and that thresholding does not consider spatial relationship between pixels, this technique is hardly used alone in segmentation, but can be useful when applied as a pre-processing step (GOSHTASBY; TURNER, 1995).

2.2.2 Edge detection

Detecting edges is a more refined approach to segmentation than thresholding. A *Sobel* operator is a discrete 3 x 3 convolution kernel which computes the gradient of the image along x and y direction. More refined edge detectors like *Canny* (CANNY, 1986) uses a double-thresholding technique that allows a better link of detected edges. Edge detection can be applied to segmentation of brain structures (BOMANS et al., 1990), thanks

Figure 2.1: Applying a Threshold filter to a CT slice. Intensities range from 0 to 255. Notice that the method is susceptible to noise, and join different structures as one single component. Region in red on the left image is the liver groundtruth segmentation

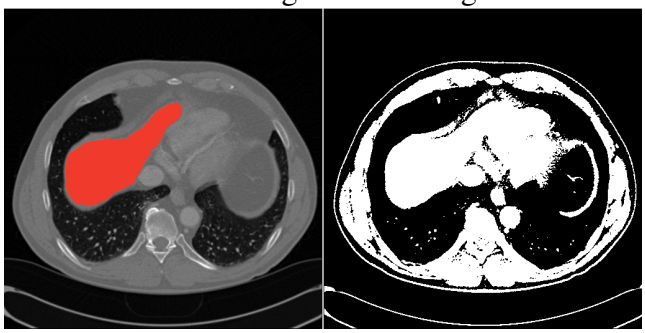

Figure 2.2: Source: original image

Figure 2.3: Edge detection of a CT slice. Despite identifying some organs contours, the liver was segmented as not-closed regions, with some leakage to the muscle tissue. Region in red on the left image is the liver groundtruth segmentation

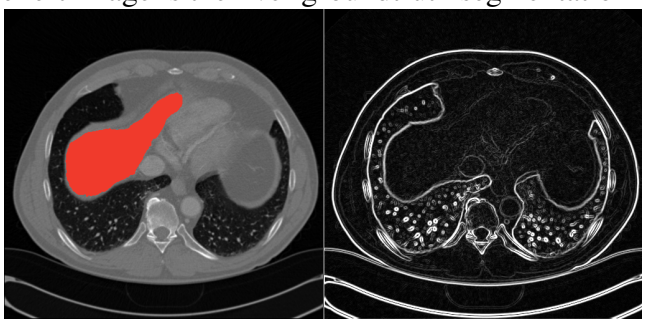

Figure 2.4: Source: original image

to the distinct intensities and definite boundaries of the brain and the skull. However, edge detectors are sensitive to noise, and edges extracted may be disjoint, needing further steps to form closed and connected object regions. For that reason, filtering is often used to smooth higher frequencies caused by noise, while preserving important features like fractures in bones (HACIHALILOGLU et al., 2008). In a soft-tissue segmentation scenario, using only edge detection is impractical, as seen in Fig. 2.3.

2.2.3 Region-based

Region-based is another subclass of local-feature algorithms. A good representative of this class is the **Watershed Algorithm** (BEUCHER; LANTUÉJOUL, 1979). An intuitive interpretation of the technique is to think about the image as a surface which is flooded by water. The height of each pixel is given by its intensity. Markers, either user-defined or by local minimum, are then used as seed points to flood regions, and form basins. Whenever two basins meet, a watershed line is formed. This process is presented on Fig. 2.5.

Problems with the watershed algorithm is possible over-segmentation, since each local minimum will form a basin regardless of the region size. To improve the robustness of the technique, prior information can be encoded in the algorithm (GRAU et al., 2004), so instead of using the gradient to form basins, a probability function is applied. However, this solution does not solve problems such as noise sensitivity and difficulty to detect thin

Figure 2.5: Example of the watershed algorithm. Two connected shapes with two markers inside and a red line crossing the entire image (a). A height representation of the pixels where the red line passes (b). The two green markers are the starting point for the water to fill the basins. Where two different flooded regions meet (orange and blue) a watershed is formed (purple line). In (c), is the resulting segmentation of the watershed algorithm.

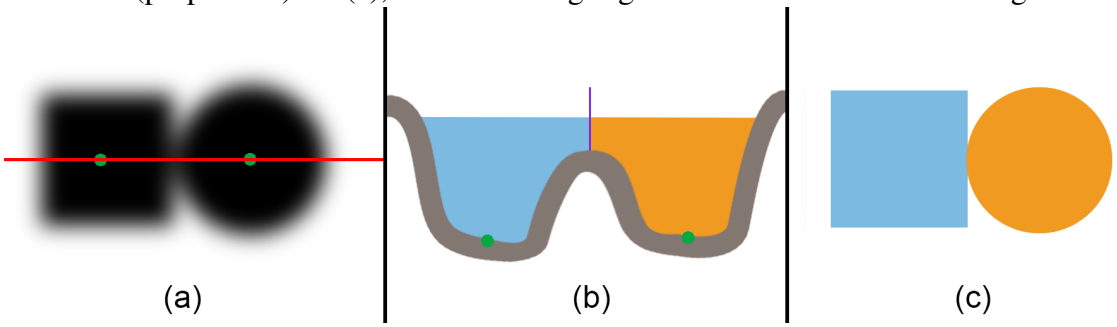

Figure 2.6: Source: original image

structures.

2.2.4 Graph-based

The intuition behind graph-based technique is that they create a weighted graph, where each vertex corresponds to a pixel or region, and each edge is weighted according to similarity between neighboring pixels or regions. A graph $G = (V, E)$ can be partitioned in two disjoints sets A and B by removing edges between them. Hence, graph-based algorithms try to minimize certain cost functions, like *cut*.

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v) \tag{2.1}$$

where $w(u, v)$ is the weight of the edge between u and v. One approach to the minimization is the *minimum cut* (WU; LEAHY, 1993), which partitions a graph into k subgraphs such that the maximum cut across the subgroups is minimized. This cutting criteria however, tends to generate small sets of nodes, since the resulting value of Eq. (2.1) is proportional to the size of the sub-graphs. This bias can be avoided through the *normalized cut* approach (SHI; MALIK, 2000), which replaces Eq. 2.1 with a new cost function *Ncut*:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, V)}{assoc(B, V)} \tag{2.2}$$

$$assoc(X, V) = \sum_{u \in X, t \in V} w(u, t) \tag{2.3}$$

where $assoc(X, V)$ in Eq. 2.3 is the total connection from all nodes in X to all nodes in the graph, hence the normalization aspect. Graph cuts with user interaction were used to remove bones from abdominal CT images (BOYKOV; JOLLY, 2001), as seen on Fig. 2.7.

Performance for the graph cut method is good only for images were background and foreground intensities are well separable. Initial classification can be automated, as in the work by Wels *et. al* (WELS et al., 2008). They use probabilistic boosting trees to classify voxels as background or foreground, in brain tumors segmentation.

Figure 2.7: Bone segmentation in a CT image using interactive graph cut (BOYKOV; JOLLY, 2001). Markings "O" and "B" for object and background respectively, are manually initialized. Bone regions are marked by the horizontal lines.

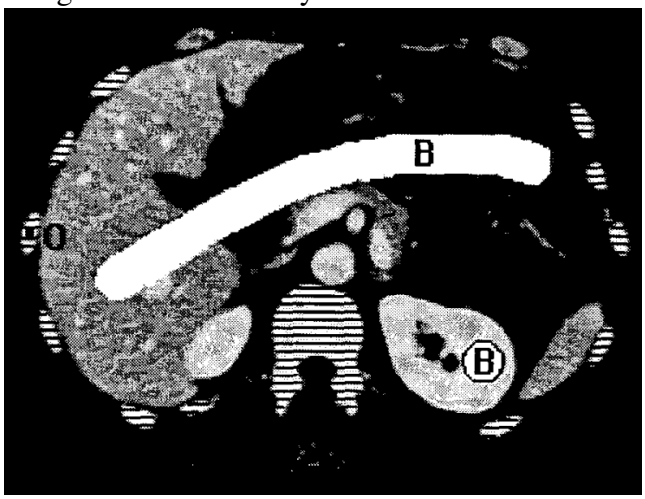

Figure 2.8: Source: Boykob et al (BOYKOV; JOLLY, 2001)

Another assumption for the graph-cuts technique is that the object's shape is best described by the shape with smallest boundary length, which is not always the case in the context of medical images, specially soft tissue organs like the liver.

2.2.5 Classification-based

Functions based on classification may use criteria like texture and brightness similarity, contour energy, etc. to train a classifier to identify good segmentation (REN; MALIK, 2003). For that, human segmented images are used as positive examples, and negative examples are generated through randomly matching human segmentation with different images. The algorithm then groups *super-pixels* (which are regions generated by a pre-processing step using the *normalized graph cut* approach(SHI; MALIK, 2000)) into segments.

Another form of classification functions applied in image segmentation are Neural networks (TOULSON; BOYCE, 1991). The training set consists of manually segmented samples. Segmentation then, is performed on each pixel. Inputs for the neural network are the class membership probabilities of the pixels from a neighborhood around the pixel being classified. This improves the spatial consistency of the segmentation.

Gaussian Mixtures are statistical probabilistic classification models that represent the presence of subpopulations within an overall population through the usage of Gaussians. A Gaussian Mixture Model (GMM) can be defined as:

$$g(x) = \sum_{i=1}^{0} ma_i f_i(x) \tag{2.4}$$

where x is the voxel intensity, a_i are normalized coefficients, and $f_i(x)$ are Gaussian distributions with parameters $\theta_i = (a_i, \mu_i, \sigma_i)$. It is possible to automatically estimate θ_i by the Expected Maximization technique (DEMPSTER; LAIRD; RUBIN, 1977). The number of Gaussians can also be automatically determined by an adaptive binning algorithm (LEOW; LI, 2004). An example of Gaussian Mixtures classification of the

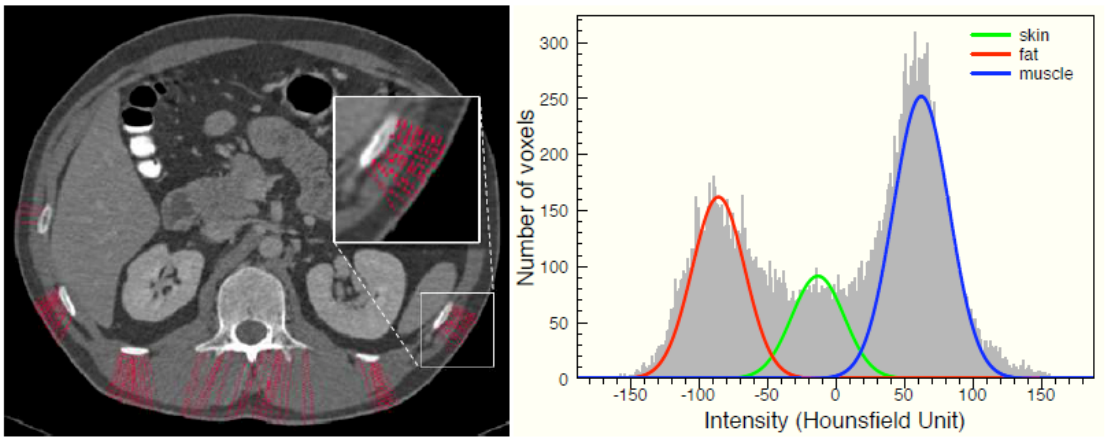

Figure 2.9: (a) CT scan from abdominal region. Red lines are sampling regions for voxel intensities that have bone muscle and fat transitions. (b) Gaussian Mixture Models obtained from the sample data from (a).

Figure 2.10: Source: Ding Feng et Al (DING et al., 2009)

abdominal region is presented on Fig. 2.9. Gaussian Mixture Models have been used for automatic segmentation of MR brain images (GREENSPAN; RUF; GOLDBERGER, 2006) and for the removal of abdominal wall on CT volumes for 3D visualization (DING; LEOW; VENKATESH, 2009). However, certain organs have similar intensities and may share a certain gaussian probability, causing leakage problems in segmentation.

Classification-based segmentation algorithms require training. Parameters for training are usually subjective and empirically defined. Since accuracy of these algorithms heavily depend on the selected training set, and there is a large variation from one person to another for the same target segmentation organ, a large amount of data is required to cover possible different scenarios. These algorithms are also sometimes complex to implement.

2.3 Deformable Models based Segmentation

These methods became popular recently, because of the ability to handle shape variation of the target organ. Domain information, and training shapes may also be incorporated to help on shape variability. There are many different deformable model segmentation algorithms, and we will discuss some of the most relevant below.

2.3.1 Active Contour Models

Active Contours, also known by the popular name ***Snakes*** (KASS; WITKIN; TERZOPOULOS, 1988) are curves defined within an image domain, that can move under the influence of two types of forces: internal and external. Internal forces come from within the curve itself, and external forces are derived from the image data. The algorithm then, deforms the curve iteratively, searching for the configuration which minimizes the total energy given by the sum of forces, as seen in equation 2.5.

$$E = \int E_{int}(s(p)) + E_{ext}(s(p))\, dp \tag{2.5}$$

Figure 2.11: Original binary image (a). The vector field generated by applying eq. 2.7 in (a) is presented in (b). In (c), the resulting convergence of the snake after various iterations. Notice that the snake is unable to reach inside the concavity of the image.

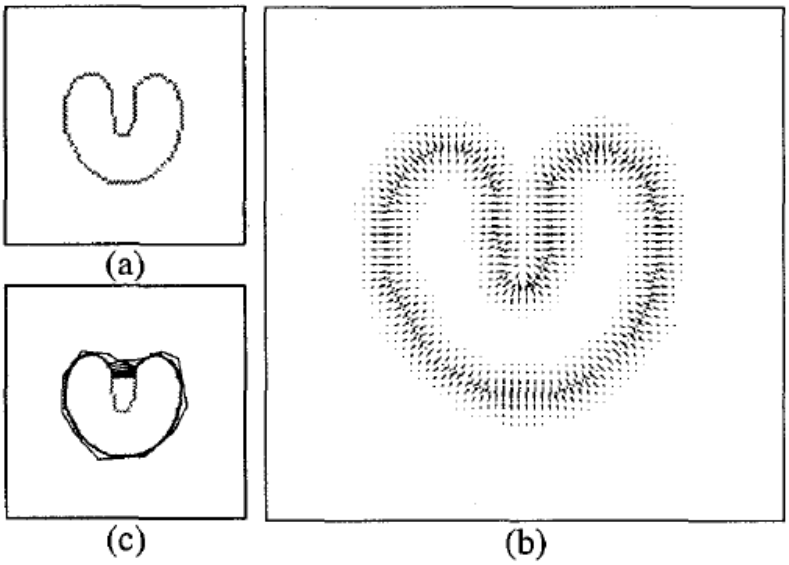

Figure 2.12: Source: Kass et Al (KASS; WITKIN; TERZOPOULOS, 1988)

Ideally, these forces combined displace the *snake* towards an object boundary or other desired feature within an image. There are various segmentation works using snakes. Segmentation of neural tissue with snakes (CARLBOM; TERZOPOULOS; HARRIS, 1994) and tracking of objects in a plane (like cells or heart chambers) (LEYMARIE; LEVINE, 1993) are some of the applications of the original method. Naturally, snakes are good deformable models for edge detection, shape modeling, segmentation and motion tracking, given the smooth curvature of the spline, which is second-order continuous. The external forces from the snakes method (KASS; WITKIN; TERZOPOULOS, 1988) are defined as:

$$E_{ext} = -|\nabla I(x,y)|^2 \tag{2.6}$$

Notice that for that external energy formulation, the method is sensitive to the starting configuration of the snake. It also has problems with convergence to concave parts of the region, or distant intensities. The original paper attempts to solve those cases by defining a different external energy function:

$$E_{ext} = -|\nabla(G_\sigma(x,y) * I(x,y))|^2 \tag{2.7}$$

where $G_\sigma(x,y)$ is a two-dimensional Gaussian function with standard deviation σ, and ∇ is the gradient operator. This spreads the reach of higher intensities and helps to solve some of the possible noise issues. However, larger σ values will blur and distort the original boundaries. Also, the problem of concavities remains, as seen on Fig. 2.11. This problem was tackled and partially solved by the Gradient Vector Flow (GVF) technique (XU; PRINCE, 1998).

Starting from an *edge map* obtainable for example through eq. 2.6, a diffusion of the gradient vectors (hence, **gradient vector flow**) is performed iteratively by solving the

Figure 2.13: In white, the edge map of a black and white U shaped image. In green, the vector field obtained by applying the gradient vector flow diffusion on the edge map. Notice that gradient magnitude is bigger near the edges of the image, meaning stronger forces close to high frequencies and small forces on homogeneous regions. The concavity also has better force spread, allowing convergence where the original snake's external energy formulation would fail.

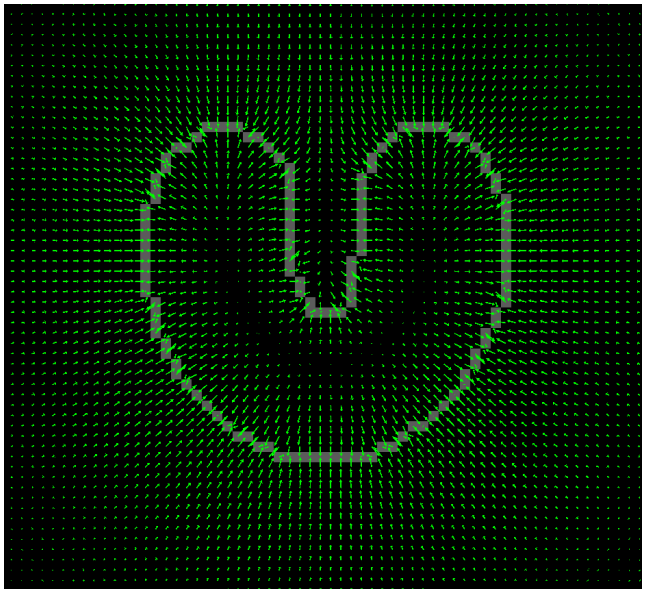

Figure 2.14: Source: Xu et Al (XU; PRINCE, 1998)

following euler equation:

$$g'(x) = g(x) + \mu\nabla^2 g(x) - (g(x) - f(x)) * |\nabla f(x)|^2 \tag{2.8}$$

where ∇^2 is the Laplacian operator, $f(x)$ is the *edge map* (or gradient) of the image and $|\nabla f(x)|^2$ is the magnitude of the gradient of the image, and $g(x)$ is the current state of the gradient vector flow diffusion. The discrete solution is obtainable, for example, by finite differences. The result of the gradient vector flow diffusion operation is presented on fig. 2.13.

Problems with gradient vector flow are the sensitivity to the snake initialization, and most of all, the high computational cost to compute the iterative diffusion operation. Recent works were able to reduce the computational cost considerably (HAN; XU; PRINCE, 2007), but noise sensitivity and initialization remains a problem. Segmentation using snakes also are susceptible to self-intersections. This happens when vertices are projected in such a way that the snake crosses itself, causing a self-intersection artifact. The T-snake model (MCINEMEY; TERZOPOULOS, 1999) modification solve intersections by discretizing the segmentation space into a grid, and tracking underlying grid points. If a intersection is detected, that deformation is cancelled, and a repulsion force is exerted on those vertices. This approach can lead to slower segmentations and unstable behaviour on complex and detailed shapes.

2.3.2 Level Set

This function (SETHIAN, 1999) is a higher dimensional surface that represents a contour. That contour is the intersection between the level set function and the $x - y$ plane.

Figure 2.15: A level set function 3D surface (top row) evolving as the contour (bottom row) deforms and adapts to the image plane (SETHIAN, 1999).

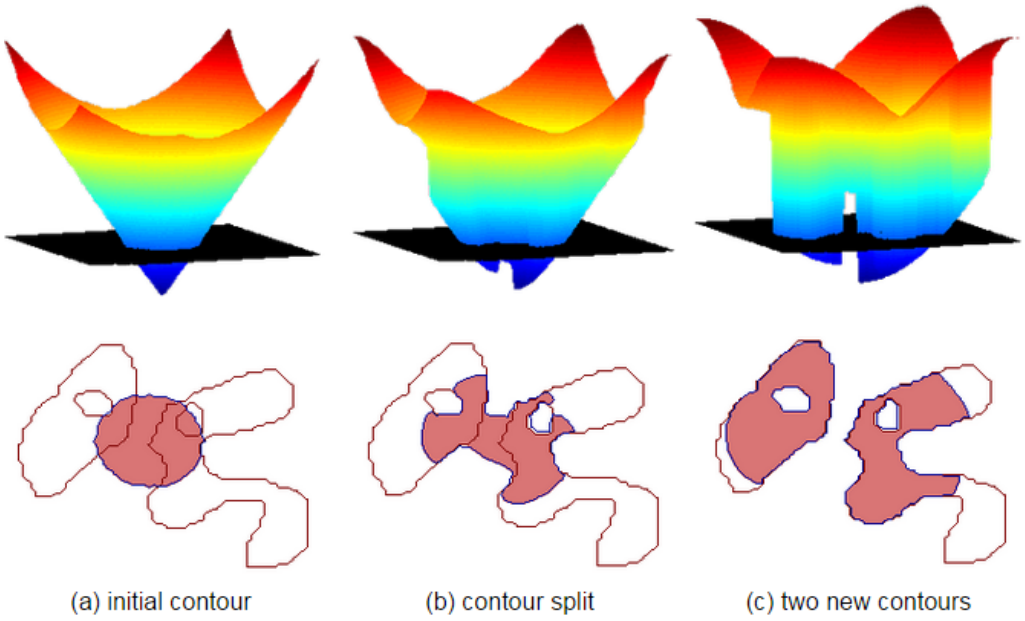

Figure 2.16: Source:Sethian et Al (SETHIAN, 1999)

For a 2D contour, the level set function $z = \phi(x, y, t = 0)$ is represented as a 3D surface (for a 3D volume, the level set function is a 4D hypersurface). The desired object contour is the zero level set of the function, formally: $\phi(x, y, t) = 0$. As in Snakes (KASS; WITKIN; TERZOPOULOS, 1988), the contour is propelled by image forces F. Since the level set function is a representation of the states of evolution of the contour, it cannot be pre-computed. The solution is to compute the zero level set function iteratively based on the force F. Since the changes on the level set function values are only near the current object boundary, this allows for an optimization called narrow band algorithm (SETHIAN, 1999). If the contour has only one direction of propagation, another optimization possible is the fast marching algorithm (SETHIAN, 1999). An example of the level set function is shown on fig. 2.15. The level set technique are widely used on medical volume segmentations.

Segmentation of cardiac 2D MRI images using level set function (PLUEMPITIWIRIYAWEJ et al., 2005) was achieved by combining stochastic region information with edge information. Contour smoothness constraints were adjusted by minimizing overall contour length. The level set function is computationally expensive to calculate.

2.3.3 Active Shape Model and Active Appearance Model

Both methods use statistical analysis to obtain average shapes of organs. For that, a collection of shapes (a set of training samples) is required.

The Active Shape Model (ASM) also known as Statistical Shape Model (SSM) (COOTES et al., 1993) represents a training shape by a $2n$-dimensional vector containing coordinates of points on the shape. These training samples form a point cloud in the eigen space. ASM uses Principal Component Analysis (PCA) on this point cloud to identify eigenvectors that describe that point cloud. An arbitrary shape can be represented by linear combination

Figure 2.17: Two cases of a mesh self-intersection in a 2D representation. In (a), a self intersection caused by two directly connected vertices. In (b), a self-intersection caused by multiple vertices which are not necessarily connected.

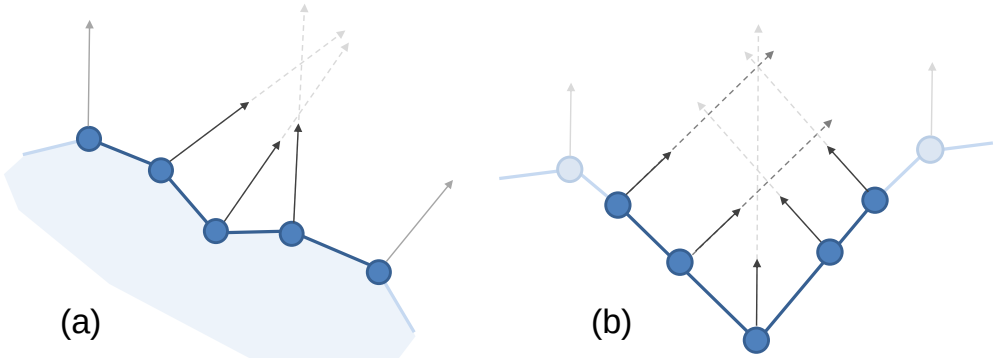

Figure 2.18: Source:original image

of these eigenshapes with different coefficients. Similarly, a model can be deformed by changing these coefficients. An initial guess can be randomly generated, and an optimization algorithm (like genetic algorithm or a direct search in the eigen space) can be used to find the optimal solution iteratively. Active Appearance Models (AAM) (COOTES; EDWARDS; TAYLOR, 2001) also incorporates voxel intensity information, improving the robustness of ASM.

Fully automated liver segmentation with a reduced number of training samples was performed using a multi-level statistical shape model approach (OKADA et al., 2008). This was achieved by partitioning hierarchically the model in patches, which have to be overlapped during deformation to ensure continuity. This introduces more complexity to the technique.

The advantage of both ASM and AAM models is that the shape can be deformed in a more controlled way when compared to snakes or level set method. However, it also requires a lot of training samples (to avoid bias or bad results) to build the point distribution functions in the eigen space. There is also the problem that an eigen space with too many eigenshapes may introduce a higher complexity in finding the optimal combination of these for the optimal solution.

2.3.4 Problems in Deformable Model Based Segmentation

Despite deformable models being more resilient to noise and that they consider the overall information of a target organ (in comparison to local information based methods, which are more sensible to noisy data), there still are problems that can affect the quality of the segmentation obtained. Deformable models are susceptible to self-intersections that happens if a mesh is not deformed properly. Two examples of self intersection are presented on Fig. 2.17.

There are different approaches to solve self-intersection of the mesh during segmentation. One possible solution is to detect the artifacts and solve them. For instance, proximity conditions can be used (LACHAUD; MONTANVERT, 1999) between vertices. Small displacements are performed, and when a violation of a proximity condition is detected, the model is remeshed to remove self-intersections. The method by Delingette et. al. (DELINGETTE; MONTAGNAT, 2001) uses collision detection to check for self-intersections and performs remeshing to remove the artifacts. In general, these approaches

are slow, contributing to most of the computational cost of the algorithm.

2.3.5 Summary

Medical image segmentation algorithms are categorized in terms of interaction modes in one of the following: manual, fully automatic, interactive and semi-automatic. Manual segmentation methods rely heavily on human interaction, are accurate and time consuming. They have large variability as it relies on the observer notion of anatomy. Fully automatic segmentation methods do not use human labour, and its results are repeatable. They rely on training sets or statistical measures, and currently are not robust enough to be used clinically. Interactive and semi-automatic methods have the best trade-off, incorporating minimum user input and yielding clinically acceptable results.

In terms of category, we presented two different types of techniques: model-less and deformable model-based. Model-less methods (like thresholding, edge-based, graph-based and classification-based) use local information of the image. Usually they are time-efficient and have good results for cases where tissues have very different intensities for example, or low noise ratio with low intensity variation (like bones and air on a CT scan). They do not perform well on cases where intensities are inhomogeneous, noisy or have highly irregular shape and low contrast, or indefinite edges and intensity similarities are high. Deformable Model-based segmentation, however, are defined as a contour/surface deformation. Global information (like geometric properties) are used to ensure more robustness and coherence. Therefore, they better deal with inhomogeneous intensities or noisy datasets. They also have higher computational complexity.

Techniques which are based on deformable models suffer from problems like self-intersection of the mesh. This problem happens when a mesh is not properly deformed, causing its surface to intersect with itself. Most solutions to this problems require heavy computational steps or complex implementations.

The technique introduced in this thesis is a semi-automatic deformable model method. It includes minimal user input, and can segment organs in the complex abdominal region such as the liver or spleen. It does not rely on statistical shapes or training sets. It also avoids problems like self-intersection of the mesh by simple constraints integrated in the deformation framework. To achieve such results, we partially rely on ideas from previous deformable models techniques (LAW et al., 2011; SORKINE et al., 2004; DING et al., 2009). As stated, volume segmentation based on deformable models usually suffer from problems like self-intersection of the mesh, and undersampled regions. The presented previous solutions for those problems required costly and complex operations. Our method addresses these issues avoiding bottlenecks, allowing for significantly faster computation in comparison with other algorithms of this kind.

3 DEFORMABLE SPHERES SEGMENTATION FOR PATIENT SPECIFIC MESHES

In this section, we explain our technique for obtaining a patient-specific mesh, which is based on previous segmentation techniques using deformable models (ZHOU et al., 2012; DING; LEOW; VENKATESH, 2009; DING et al., 2009). An overview of the steps of the algorithm is observable on Fig. 3.1. Details are in the subsections 3.1-3.6 below.

3.1 Initialization Step

Medical images usually have different resolutions on each sampling direction. If we consider the volume to be aligned with the *x*-*y*-*z* axis, that means that a voxel will not have the same length on each direction. The reason for that is related to the sampling rate on the acquiring means (CT machine, for example). While it does not interfere directly with segmentation techniques, it is of interest for our algorithm to have isotropic voxels (meaning that their length is the same on all directions). Since we will be using an explicit mesh to perform segmentation, no posterior scaling will be necessary to create a shape which actually resemble the organ of the patient. Additionally, since we use geometrical constraints to guide the mesh deformation, having isotropic voxels would make calculations of correlation between vertices position and volume position easier. Finally, simply stretching the convergence state may create unappealing results, while using isotropic voxels has better chances to capture soft tissues smooth curves and irregularities.

For those reasons, the initialization step performs a regularization of the volume, to ensure voxels size is the same in all sampled directions. To achieve that, we create a new volume with the *real* dimensions of the input volume. Discretization considers the smallest length direction of the input volume voxels. That means that a volume with number of voxels 512x, 512y and 183z with voxel sizes $0.75mm$ in x, $0.75mm$ in y, $1.5mm$ in z has real dimensions of 512 by 512 by 366 voxels considering isotropic voxels with $0.75mm$ size in all directions. To generate the new isotropic volume, tri-linear interpolation is performed to ressample the volume. After this step, the volume is shown with a volume visualization framework, with the possibility to also visualize individually sampled slices spatially aligned with the entire volume.

Before the algorithm starts, the user manually places seed points inside the target organ using this visualization. The number of points is predefined according to the target organ, as shown in Fig. 3.1b. That is the only human intervention needed.

A seed point is regarded as the center of a special spherical mesh, defined as a set of vertices V, each of them, a member of two intersecting orthogonal contours. Thus, each vertex v_i in V is connected to exactly four vertices, resembling a quad-sphere structure, as

Figure 3.1: The overall steps of the algorithm. A close up of one spherical mesh (a). Each vertex of the mesh is part of two orthogonal contours. Notice the lack of poles on the spherical mesh, to avoid concentration of vertices. In **(b)** we have four spheres placed in the volume space. Each vertex searches for *feature voxels* in the normal direction. Presented in **(c)** are deformed vertices using a Laplacian preservation scheme while avoiding self-intersection and stopping when at boundary regions of other meshes. In **(d)** is the final liver mesh obtained by joining the four partial meshes.

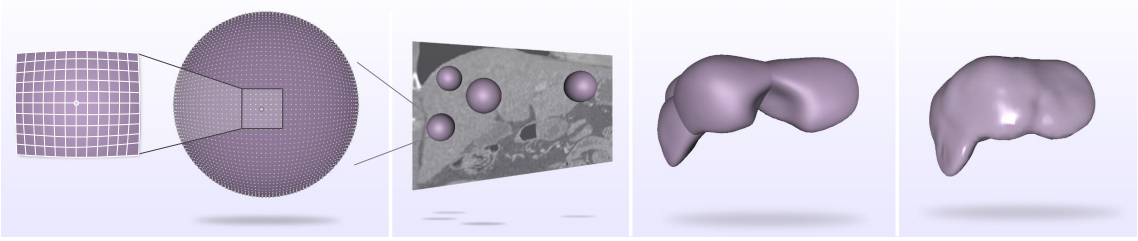

Figure 3.2: Source:original image

shown in Fig. 3.1a. A linear ordering of vertices is defined along each contour, making the later avoidance of neighbor flipping easier. The *side* of the quadrilateral mesh, measured by the number of vertices, defines how much detail information the mesh will be able to sample. For instance, the total amount of vertices is given by $V = side^2 * 6$.

During the first step of the algorithm, presented on subsection 3.2, a search for *feature-voxels* is performed. We defined *feature-voxels* as values outside an intensity window. The intensity window was obtained by evaluating all voxels inside each spherical mesh before the algorithm starts. The maximum and minimum intensities found within each sphere are the response ranges for that sphere. Voxels with lower or higher intensities than those maximun and minimum intensities are considered as our *feature-voxels*. We chose this approach because it is generic and not biased to a particular type of organ.

After initialization, the mesh is deformed according to the steps describe in Algorithm 1 presented below.

Algorithm 1 Multiple spheres growth algorithm

```
1:  Spherical Meshes initialization
2:  while convergence criteria not met do
3:      for all Meshes m in M do
4:          3.2 Correspondence search
5:          3.3 Detect and avoid flipping
6:          3.4 Detect and avoid self-intersection
7:          3.5 Perform mesh deformation
8:      end for
9:      for all Mesh m in M do
10:         3.6 Inter-mesh intersection detection.
11:     end for
12: end while
```

3.2 Correspondence search

The deformation of the spherical mesh is performed for each vertex v_i of the mesh. Ideally, the target position v_i is at the boundary region of the desired organ. Each vertex v_i searches along a projection line $P(v_i)$ for a corresponding point v_i'. This projection line can be defined as the surface normal at v_i. The point v_i' is the intersection of $P(v_i)$ and a *feature voxel* on the surface of the volume. If a projection $P(v_i)$ fails to intersect a feature voxel, v_i is labelled as a *solitary vertex*. The length of that projection line is defined as half the width of the volume, and decreases on each iteration.

Given that the projection's direction may greatly vary within a small region of the mesh, an iterative local averaging of the displacements vector is performed as:

$$v'_{avgi} = vi + \frac{1}{N+1}[v_i' - v_i + \sum_{v_j \in N(v_i)} (v_j' - v_j)] \tag{3.1}$$

where v_j are the vertices connected to v_i that have a target correspondence v_j', and N is the total number of these vertices. This operation leads to more uniform projections $P(v_i)$. This operation also yields valid target positions to former solitary vertices, since it is expected that a boundary in a organ is continuous within a direct neighborhood.

3.3 Detection and avoidance of mesh flips

Displacing vertices along their projection line may cause mesh flippings. The flipping of a non-solitary mesh vertex v_i and its non-solitary neighbors v_j after mesh deformation can be identified by:

$$\frac{v_i - v_j}{||v_i - v_j||} \cdot \frac{v_i' - v_j'}{||v_i' - v_j'||} \leq \tau \tag{3.2}$$

where $\tau \in [0, 1)$ is a predefined threshold. Pairs of vertices that form flipping edges are labelled as *flipping vertices*. Each vertex undergoes flipping detection for all of its non-solitary neighbors.

After checking and labelling all flipping candidates, the algorithm iterates each closed contour. Let $v_i, ...v_n$ denote a consecutive sequence of flipping vertices for a given contour, bounded by two non-solitary and non-flipping vertices v_{i-1} and v_{n+1}. The method identifies the central flipping vertex v_c and considers it as non-flipping. All other flipping vertices in the sequence are labelled as solitary and have their correspondence discarded. As illustrated in Fig.3.3, this process avoids edge flips because vertices labelled as solitary are not projected directly, having their new positions given by a Laplacian preservation scheme, explained in Section 3.5.

This step avoids flipping of directly connected vertices, and has been previously used in volume segmentation (LAW et al., 2011; DING et al., 2009). However, it has one critical limitation: it does not prevent global self-intersections on the mesh. Global self-intersections can be responsible for leaking problems: vertices pierce the mesh and find target positions in wrong directions. After deformation, the meshes inflate incorrectly and lose sharp features, as seen in Fig. 3.5. This problem can also cause under sampling, due to the wrong target position that vertex may find. Our solution to that limitation is presented next.

Figure 3.3: Flipping detection on connected vertices: Presented in **(a)** is the vertex search for a feature voxel. In **(b)**, vertices detected as flipping (marked in dashed light blue) are labelled as solitary and their correspondence is discarded. Iteration of mesh deformation with Laplacian preservation guiding solitary vertices is presented in **(c)**

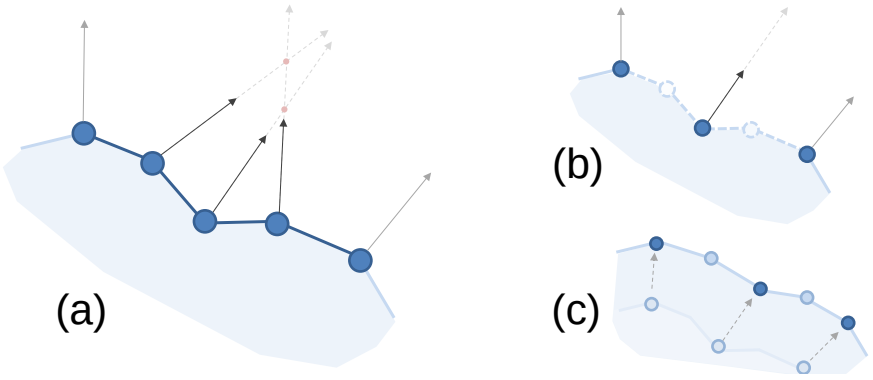

Figure 3.4: Source:original image

Figure 3.5: Mesh self intersection. Internal vertices have their normal direction flipped, causing them to be attracted to wrong target surfaces on further iterations of the algorithm. The outside **(a)** and inside **(b)** view of a liver mesh. The region in red has a self-intersection artifact. Observe that shading differences in the mesh were caused by normal flippings.

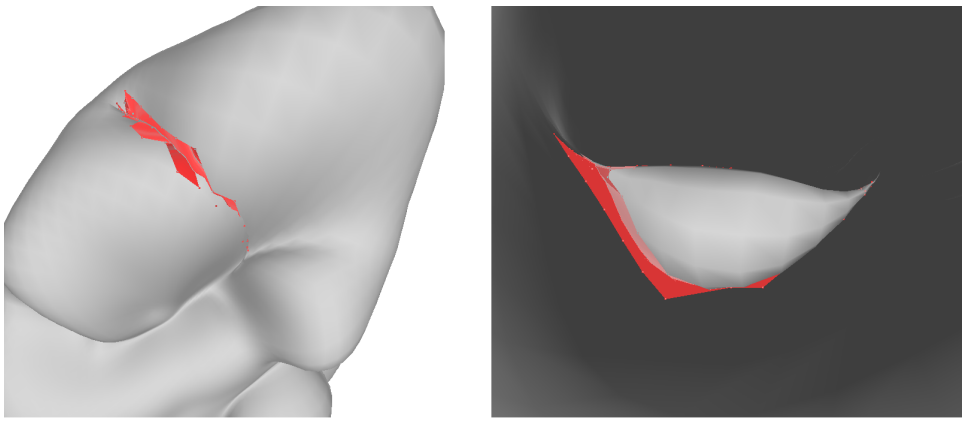

Figure 3.6: Source:original image

3.4 Detection and avoidance of global self-intersections

A global self-intersection happens with vertices that are not directly connected to each other (and thus, can not be marked as flipping because of each other). If vertices are displaced after projection, the mesh surface may end up intersecting itself. An example can be seen in Fig. 3.7.

Current solutions for that problem generally include computationally heavy operations

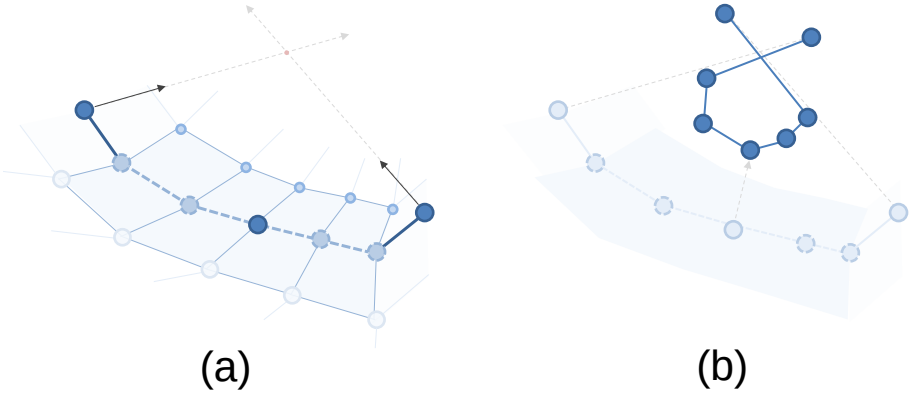

Figure 3.7: **(a)**Two vertices not marked as flipping but have crossing projections. **(b)**After deformation, projecting the vertices cause a self-intersection. Since the vertices were not neighbors, they were not checked for flipping with each other.

Figure 3.8: Source:original image

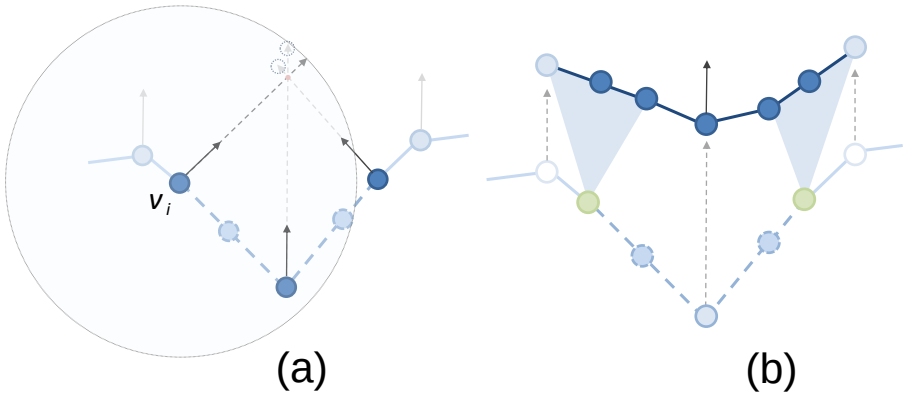

Figure 3.9: **(a)**Same scenario of Fig. 3.7 with non-neighbor vertices in an intersecting path. A vertex v_i will search within the spherical region with radius equal to the size of its projection line. If a vertex (or its target position is found, both this vertex and v_i are marked as *mesh-intersecting*. **(b)**All *mesh-intersecting* vertices have their target position discarded, and are interpolated using geometrical constraints.

Figure 3.10: Source:original image

like remeshing or re-triangulations. But we do not wish to loose our mesh connectivity properties, because that would imply in re-computations during the deformation step, and also bring problems during future steps of our technique.

The approach we devised to solve the problem of global self-intersection is to test non-solitary vertices against other close vertices within a specific spherical region. For instance, consider v_i and v_k as a pair of non-neighbor vertices to be tested: we create a bounding sphere region S_{v_i} centred at v_i with radius equal to $P(v_i)$. If either v_k or v'_k are inside S_{v_i}, the edge flip test (eq. 3.2) is performed. However, instead of labelling these vertices as flipping, they are labelled as *mesh-intersecting* (Fig. 3.9). Vertices labelled as

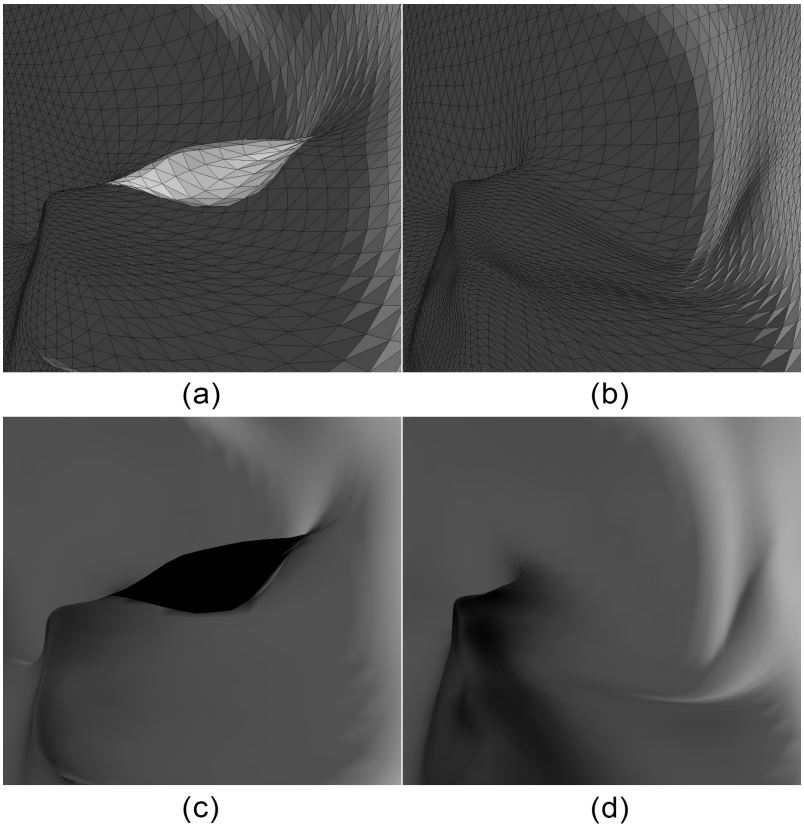

Figure 3.11: Inside of a spherical mesh deformed to fit a medical volume without (a) and with (b) our self-intersection avoidance technique. c)Ambient occlusion mapping of the mesh obtained in (a). Since ambient occlusion maps how much a vertex is visible, vertices that are "inside" of the mesh (and occluded by the geometry) are in a darker color. This shows the loss of details of the region, caused by the artifact of the intersection. d)Ambient occlusion map of (b). Notice the smoother visibility of the vertices, meaning that they capture the variations and details of the organ properly.

Figure 3.12: Source:original image

mesh-intersecting also have their target position discarded, but are treated in a different manner from solitary vertices during deformation step, which will be explained in the next section. A mesh with self intersection and with the described step to avoid the self-intersection is presented on Fig. 3.11.

3.5 Mesh deformation

After every vertex has found its target position or been labelled as *solitary* or *mesh-violating*, a Laplacian preservation scheme (SORKINE et al., 2004) is used to deform the mesh. Vertices labelled as *solitary* have no target position. After deformation, we preserve their Laplacian coordinates using an energy minimization approach, which creates a smooth deformation while preserving previous features.

To calculate the Laplacian deformation, first we obtain all δ_i coordinates as the result

of applying the discrete Laplacian operator to v_i:

$$\delta_i = [\sum_{i,j \in N} w_{ij} v_j] - v_i \tag{3.3}$$

where $\sum_{ij \in N} w_{ij} = 1$ and a uniform weighting factor $w_{ij} = \frac{1}{N}$. Since the entire mesh can be represented as an NxN Laplacian matrix L, with elements

$$L_{i,j} = \begin{cases} 1 & i = j; \\ w_{ij} & (i,j) \in N; \\ 0 & otherwise; \end{cases}$$

the Δ_v vector of all δ_i coordinates can be computed in the matricial form as $\Delta_v = LV$ where $\Delta_v = [\delta_{v_1}, \delta_{v_2}, ...\delta_{v_n}]^T$. All vertices marked as *mesh-intersecting* have their Laplacian coordinate δ_v set to 0 in the Δ_v vector. As the mesh grows, even if vertices have their correspondence discarded, there is no way to ensure that Laplacian preservation will not cause intersections of the mesh. Setting $\delta_v = 0$ for all *mesh-intersecting* vertices centralize them as a linear combination of their non-mesh-intersecting neighbors.

All vertices which have a target-position v_i' are used as position constraints. The resulting over-determined linear system $AV' = b$ has the following structure:

$$\left[\frac{L_{NxN}}{P_{MxN}}\right] V' = \left[\frac{\Delta_v}{V_p}\right] \tag{3.4}$$

where L_{NxN} is the Laplacian matrix, P_{MxN} is a sparse matrix with a 1 on the index of the vertex corresponding to that positional constraint, i.e, P is a constraint matrix where $P_{i,j} = 1$ when $i == j$ and 0 otherwise for each non-solitary vertex v_i. The Δ_v vector are the Laplacian coordinates, and V_p are the absolute coordinates of the positional constraints.

The system is solved in the least square sense by the pseudoinverse formula $V' = (A^T A)^{-1} A^T b$. Likewise, V_p are the absolute coordinates for each non-solitary v_i. The system can be independently solved for coordinates x, y and z of V'. The resulting effect is the preservation of Laplacians interpolating positions for solitary vertices, and interpolation yielding valid positions to *mesh-intersecting* vertices as shown in Fig. 3.13. Given that the system matrix is largely sparse, the pseudo-inverse can be efficiently solved by Cholesky decomposition. After obtaining new positions V_l', vertices are updated by a predefined time-step α with the following equation:

$$v_{new_i} = (1 - \alpha) v_i + \alpha v_{l_i}' \tag{3.5}$$

Convergence in our algorithm is defined as either a maximum number of iterations or by checking if the average difference between vertices displacements after some iterations is below a predefined threshold.

3.6 Inter-mesh Intersection Detection

The reason for using more than one sphere during segmentation is related to the undersampling problems. During mesh deformation, using one single sphere would cause vertices to get constrained as they approach their target surface. That means that convergence would be slower, and that farther regions would have less vertices available to

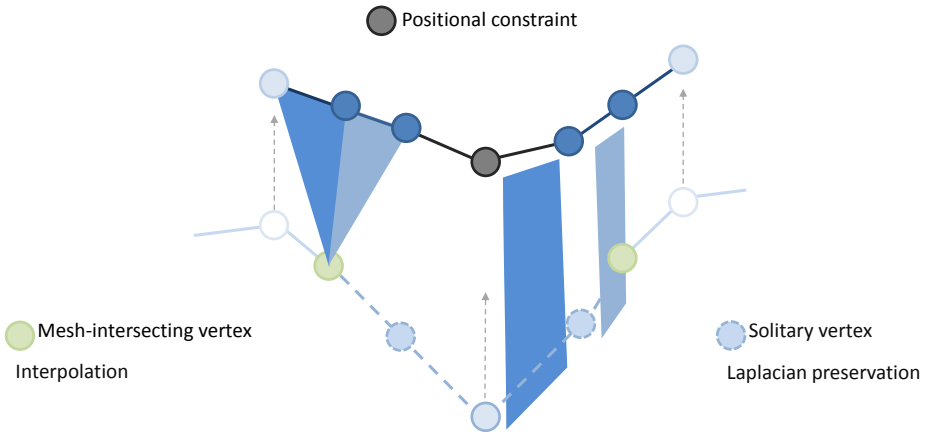

Figure 3.13: Three types of deformation on the mesh: positional constraints (black vertex) are target positions of vertices which were not marked as *solitary* or *mesh-intersecting*. Light-blue vertices are vertices marked as *flipping*. Their positions are given by their former Laplacian coordinates. Green vertices are those marked as *mesh-intersecting*. Their position are given by an interpolation of their neighbors.

Figure 3.14: Source:original image

sample the area. This leads to crude approximations and leaking problems, or to under-segmentations. Increasing (or removing) vertices from the mesh would change the mesh connectivity and require recalculation of the Laplacian matrix. It would also be more difficult to check for flips of the mesh without the linear ordering structure of the contours. Finally, increasing the number of vertices would slow down the pseudo-inverse calculation.

For that reason, we chose to use more than one sphere. With more spheres, the number of vertices for each sphere can also be smaller. Since each sphere can be processed independently, the algorithm is by nature, parallel. However, to avoid oversegmentation and to make it easier to obtain a single final mesh, it is important to also constrain intersection between different spheres.

After all spheres have been deformed by the previously described steps, we perform a two-phase intersection detection. First, a simple bounding sphere region of each mesh is created. For instance, consider two meshes M_i and M_j: we check for vertices that are in the intersection of both bounding spheres from M_i and M_j. This will yield two subset candidates for intersection between those two spheres which we will call C_i and C_j.

After these candidates have been obtained, we test for intersection using the following equation:

$$\frac{v_{new_i} - v_{new_j}}{||v_{new_i} - v_{new_j}||} \cdot \frac{v_i - v_j}{||v_i - v_j||} \leq \tau \tag{3.6}$$

where v_{new_i} and v_{new_j} are a pair of vertices from C_i and C_j but with their original positions from before the deformation, and v_{new_i} and v_{new_j} are the same pair of vertices, but after the deformation for this iteration. The threshold interval this time is $\tau \in [0, -1]$. This interval includes only vertices that are directly expanding against vertices of the other mesh, and allows both meshes to "slide" over each other ensuring a smoother cov-

Figure 3.15: Checking for intersection between two deforming meshes. In **(a)**, the bounding sphere region of both meshes. Vertices within the green region will be used to check for potential mesh intersection. In **(b)**, an example of a pair of vertices (one from each mesh) to be tested for causing mesh intersection. The dashed areas represent both meshes state before deformation. In **(c)**,the same pair of vertices, but in their position after deformation. Notice that the vectors direction from before and after the deformation are opposites, which is a characteristic of inter-mesh intersection. Example of vertices set defined as *static* while the two meshes evolve over time in **(d)**.

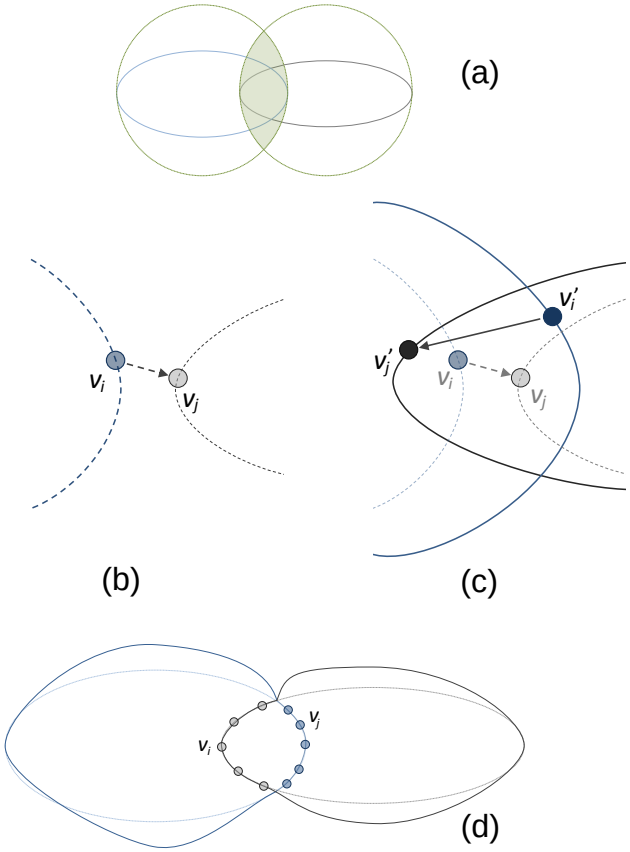

Figure 3.16: Source:original image

erage of the volume on the joining area. These vertices are then marked as *static*. They will not search for new target positions, and will always be a positional constraint during the energy minimization process. This step can be observed on Fig.3.15. As the deformation step may displace them, they are always restored to the position they were first constrained.

Since the algorithm requires to check only against vertices that are not *static*, and within a small subset of vertices, convergence of intersection is relatively fast when compared to remeshing and re-triangulations. The segmentation of a liver resulting of the described algorithm is presented in Fig. 3.17.

Figure 3.17: A liver segmentation mesh composed of four deformed meshes. In (a), the full liver. In (b), the same liver without one of the deformed meshes. In (c), the missing mesh from (b).

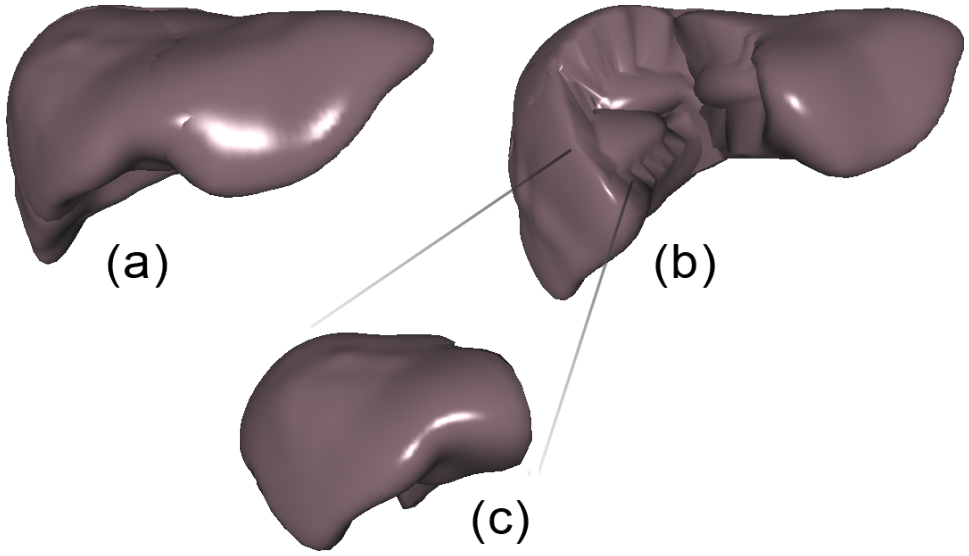

Figure 3.18: Source:original image

3.6.1 Patient-Specific Meshes

After convergence has been achieved, the output are several meshes that represent the patient-specific model. A segmentation mask can be obtained if desired, by simply checking if a voxel is inside any of the meshes applying a point in polygon algorithm. A boolean union operator can also join the meshes if the application requires so. However, since we already have the information of meshes intersection, removing the inner structures of the meshes is easy: all vertices that were marked as *static* just have to be deleted. To generate a single mesh while maintaining the overall structure of the partial meshes, a stitching algorithm can be applied (GUÉZIEC et al., 2001). The position of vertices created to weld the meshes can be interpolated by a step of Laplacian smoothing optimization (NEALEN et al., 2006). Alternatively, an entire mesh can be obtained by reconstruction techniques (KAZHDAN; BOLITHO; HOPPE, 2006). This can be observed on Fig. 3.19.

One of the downsides of using multiple seed-points (one for each mesh) is the fact that the user has to interact with the application during initialization. However, this interaction is small when compared to techniques such as active contours (KASS; WITKIN; TERZOPOULOS, 1988) or other interactive approaches. Even for large organs as the liver, four clicks (and thus, four meshes) are enough for the entire reconstruction. Moreover, there is the possibility of using automated approaches for the seed point of the spheres through statistical shape models (SSM) (HEIMANN; MEINZER, 2009), or by knowledge of human embriology (KAINMÜLLER; LANGE; LAMECKER, 2007). A statistical approach will require a good training set to avoid a wrong placement of the seed-points.

Figure 3.19: In **(a)**, the result of removing vertices labelled as *static*. Since the structure of all meshes are regular thanks to the contour lines, a stitching algorithm should not have degenerated scenarios when joining meshes. In **(b)** , a single mesh of **(a)**.

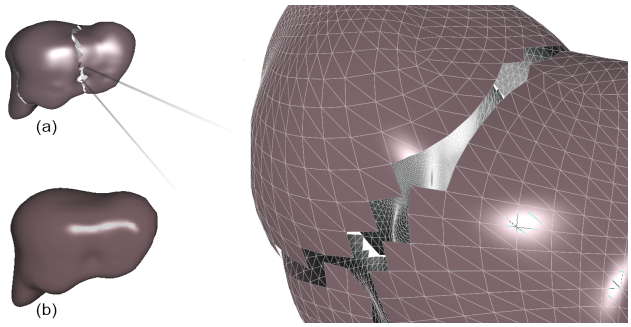

Figure 3.20: Source:original image

4 EXPERIMENTS AND RESULTS

For evaluation, we tested our technique and compared against algorithms that rely on deformable models, and use no training dataset. Graph Cut was evaluated through its variation: GrabCut (ROTHER; KOLMOGOROV; BLAKE, 2004). The level set technique (SETHIAN, 1999) implementation was the one present in ITK-SNAP. We also compared our algorithm to the Flipping-free mesh deformation (FFMD) (DING et al., 2009). For this latter, beside the results from the original paper, we also compared with results from our implementation of the FFMD, since the original source code is not available.

For testing, we considered a human liver segmentation scenario using medical images acquired by CT scan. We evaluated the average symmetric distance from the segmentations performed to a ground-truth segmentation. A smaller average distance means that the segmentation is more precise. However, distance alone is not enough to determine the quality of a segmentation. We also evaluated volume overlap percentage, which informs how much of the organ was covered in the segmentation. A high volume overlap combined with a small average distance means that the segmentation covered the organ properly while reducing leakage problems.

The medical images used in the test were 8 volumes of the liver with slice thickness varying from 1mm to 3mm, obtained from the dataset provided by the MICCAI challenge from 2007: 3D Segmentation in the Clinic: A Grand Challenge (HEIMANN et al., 2009). The dataset consisted of various CT scans acquired in transversal direction. Most of the livers were pathologic (with tumors, metastasis and cysts). Each of the volumes also had a binary mask for the corresponding liver region, which was considered as the ground-truth for our evaluations. All experiments were performed on a modest Intel Core 2 Quad 2.4 GHz with 6G memory using a single core, except by the level set, which uses more threads (due to the ITK-SNAP implementation). Parameters for time-step and intersection test in our algorithm were fixed as $\alpha = 0.5$ and $\tau = 0.4$. The number of spheres was defined as 4.

4.1 Results and Evaluation

In this section we evaluate our technique in comparison to the results obtained by the previously mentioned algorithms. Before running our algorithm, volumes were smoothed with an anisotropic filter before segmentation started to reduce some of the noise in the dataset. We applied the same filtering before running our implementation of FFMD.

In the original paper, FFMD uses Gaussian Mixtures (HABAS et al., 2008) with Expected Maximization to create a probabilistic chance of a given voxel being within a gaussian. Voxels with a probability below a certain threshold are considered their *feature-voxels*. However, the paper gives no details on the construction of the Gaussian Models,

Figure 4.1: Average symmetric distance in millimetres to the ground-truth. Smaller values mean that the technique is more precise. The two shades of gray on FFMD represents the two different implementations. The lighter gray is the result obtained in our implementation. The darker shade of gray is the result from the original paper. Notice that even in that case, the best results of our algorithm performed better than FFMD with the probabilistic preprocessing step.

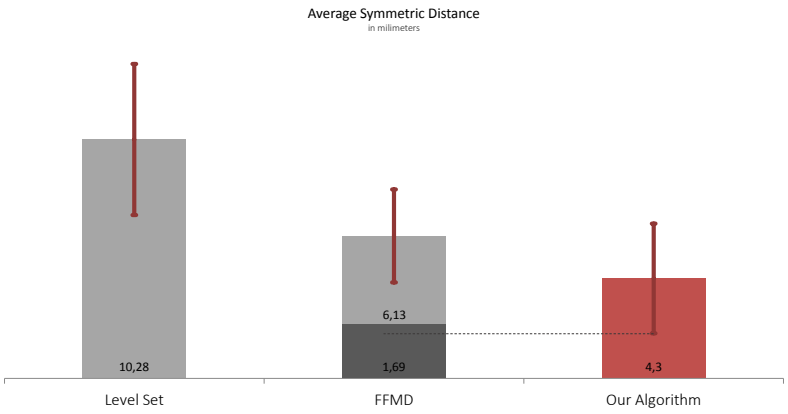

Figure 4.2: Source:original image

Figure 4.3: Volumetric overlap percentage. Higher values mean that more of the volume was covered. However, we also have to take into account the distance to the ground-truth surface to ensure that over-segmentation did not happened. Results on FFMD bar have two values: lighter gray corresponds to results reported on the paper, darker gray were obtained with our implementation results.

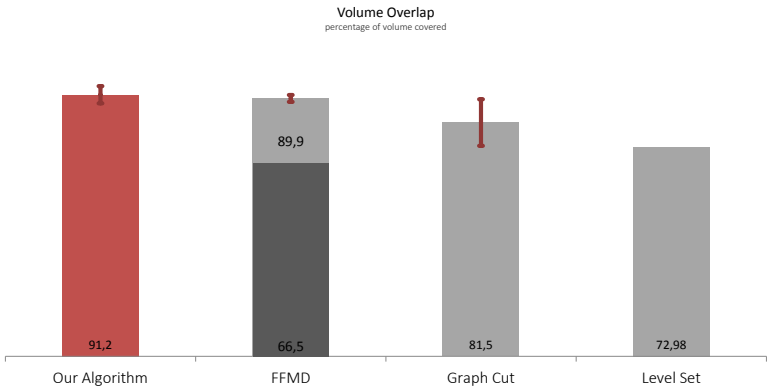

Figure 4.4: Source:original image

as to how background and foreground are defined, or if it uses a training dataset. So in our implementation, we used the same intensity-based definition we use in our algorithm. For comparisons, results both from our implementation and from original paper were used, since the dataset is the same.

The first measurement performed was the distance between the obtained segmenta-

Figure 4.5: The graph above shows the total time in seconds until convergence for the evaluated techniques (smaller is better). The graph below shows the number of iterations for each technique. Despite having to compute mesh intersection, since constrained vertices are not recomputed , the technique has a fast convergence rate. The deformation step is also independent for each sphere and can be parallelized which would also boost the performance.

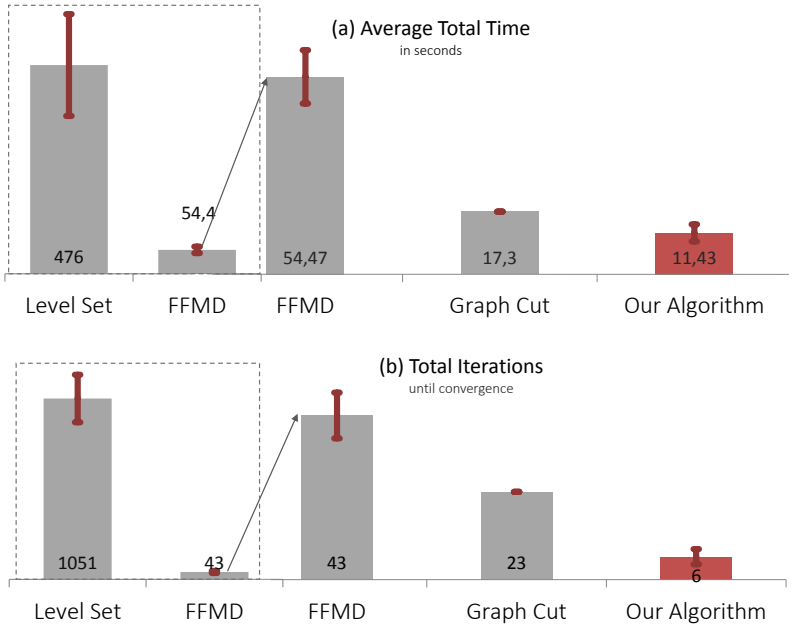

Figure 4.6: Source:original image

Figure 4.7: On the left, a mesh generated by FFMD algorithm (a). On the left, leaking of the volume close to the ribs caused a bumpy pattern. In the center, a mesh of the same liver obtained by our algorithm (b). Our method required fewer vertices and was capable of generating a more plausible result. On the right, we generated a mesh of the spleen (c), to show that our method works with different organs.

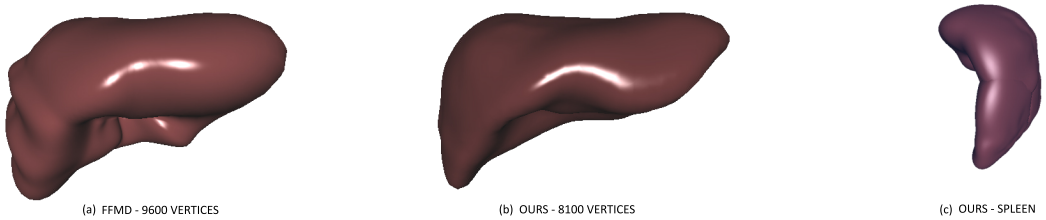

Figure 4.8: Source:original image

tion surface and the ground-truth surface. An ideal segmentation should have an average distance of 0, meaning both surfaces overlap. Our algorithm obtained a good result when compared to graph cut and Level Set, greatly reducing the distance to the target surfaces.

We also performed better than our implementation of FFMD. When compared to the results from the original FFMD paper, our technique performed worse. We are uncertain about the reasons for this, but this outcome could be originated from their implementation of the Gaussian Mixture Models. The results can be observed on Fig. 4.1. The improvement seen in our technique is given by the placement of more than one sphere, and thanks to the avoidance of self-intersection, which causes loss of details on sharp features and leaking of the volume.

As previously stated, evaluating the average surface distance alone is not a good measurement. So we also have to take into account the volumetric overlap. This measures how much of the volume was covered, which allow us to check for under-segmentations. The ideal segmentation should have a complete overlap of the volume. The volumetric overlap is observable on Fig. 4.3. A comparison of livers meshes obtained from our algorithm and FFMD, which had the second best results are presented on Fig. 4.7.

Time for each technique to achieve convergence was also measured (and is mapped on Fig. 4.5). As expected, our algorithm has a small time until convergence, specially when compared to slower techniques such as the Level Set method, thanks to the placement of the multiple spheres. Given that the multiple spherical meshes are sampling different regions of the volume, convergence is rapidly achieved. Computation of the deformation step is also faster when multiple meshes with a smaller number of vertices are used, instead of a single mesh with higher discretization. The number of iterations mapped on Fig. 4.5 is also greatly reduced for the same reason.

5 CONCLUSION

In this thesis we presented an algorithm for mesh reconstruction from patient specific anatomy data. Our algorithm is based on the inflation of multiple deformable meshes from inside the volume data and does not rely on any dataset for training or statistical calculations and requires minimal interaction. The novelties in relation to previous methods are the avoidance local and global self-intersections, and the achievement of sampling quality suitable to capture details for a number of medical applications.

Besides the easier to detect occurrences of flipping-vertex previously approached in the literature, we combined different types of constraints in the Laplacian deformation step for the vertices that are defined as candidates to non-flipping intersection. This allowed us to avoid zself-intersections.

The use of multiple spherical meshes instead of one single mesh is responsible for an improved sampling quality. As a result, high frequency features of the original volume are more accurately captured by the resulting mesh without increasing the overall number of vertices. The use of multiple spheres also leads to high parallelization, as each sphere is independent and can be computed in parallel with the others. Since there are fewer vertices on each sphere, the energy minimization method is as faster to compute. It also does not require additional constraints to perform vertices displacement (which may cause unintended intersections and leakage, and are slow to compute) or constant remeshing (which changes mesh connectivity and also introduces complexity).

Thanks to these features, our technique allows for patient-specific meshes to be obtained, with enough quality for clinical usage. These meshes can be integrated in surgery simulators for medical training and surgery planning scenarios.

Finally, there is an increasing interest in hepatic surgery applications for the automatic modeling of the liver functional segments, also known as Couinaud's segments (COUINAUD, 1986). We suggest further developments to consider the placement of seed-spheres in such way that each sphere correspond to one of the 8 functional segments defined by Couinaud. Another non-excludent possibility is using knowledge from human embryology to guide automatic placement of more than one spheres depending on the target organ.

REFERENCES

AMERICA, I. of Medicine (US). Committee on Quality of Health Care in. **Crossing the quality chasm**: a new health system for the 21st century. Washington, USA: National Academy Press, 2001.

BEUCHER, S.; LANTUÉJOUL, C. Use of watersheds in contour detection. In: IMAGE SEGMENTATION, 1979. **Proceedings...** [S.l.: s.n.], 1979. p.253–260.

BOMANS, M. et al. 3-D segmentation of MR images of the head for 3-D display. **Medical Imaging, IEEE Transactions on**, California, USA, v.9, n.2, p.177–183, 1990.

BOYKOV, Y. Y.; JOLLY, M.-P. Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In: COMPUTER VISION, 2001. **Proceedings...** [S.l.: s.n.], 2001. p.105–112.

CANNY, J. A computational approach to edge detection. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, [S.l.], v.8, n.6, p.679–698, 1986.

CARLBOM, I.; TERZOPOULOS, D.; HARRIS, K. M. Computer-assisted registration, segmentation, and 3D reconstruction from images of neuronal tissue sections. **Medical Imaging, IEEE Transactions on**, [S.l.], v.13, n.2, p.351–362, 1994.

COOTES, T. F.; EDWARDS, G. J.; TAYLOR, C. J. Active appearance models. **IEEE Transactions on pattern analysis and machine intelligence**, [S.l.], v.23, n.6, p.681–685, 2001.

COOTES, T. F. et al. The use of active shape models for locating structures in medical images. In: INFORMATION PROCESSING IN MEDICAL IMAGING, 1993. **Proceedings...** [S.l.: s.n.], 1993. p.33–47.

COUINAUD, C. **Anatomie Chirurgicale du Foie**. Paris, France: Masson, 1986. 337-342p. v.112, n.5.

DELINGETTE, H.; MONTAGNAT, J. Shape and topology constraints on parametric active contours. **Computer Vision and Image Understanding**, Cedex,France, v.83, n.2, p.140–171, aug. 2001.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the EM algorithm. **Journal of the royal statistical society. Series B (methodological)**, [S.l.], v.1, n.1, p.1–38, 1977.

DING, F. et al. 3D segmentation of soft organs by flipping-free mesh deformation. In: APPLICATIONS OF COMPUTER VISION, 2009. **Proceedings...** [S.l.: s.n.], 2009. p.1–7.

DING, F.; LEOW, W. K.; VENKATESH, S. Removal of abdominal wall for 3D visualization and segmentation of organs in CT volume. In: IMAGE PROCESSING, 2009. **Proceedings...** [S.l.: s.n.], 2009. p.3377–3380.

DORNHEIM, J. et al. Segmentation of neck lymph nodes in CT datasets with stable 3D mass-spring models. In: **Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006**. [S.l.]: Springer, 2006. p.904–911.

GOSHTASBY, A.; TURNER, D. A. Segmentation of cardiac cine MR images for extraction of right and left ventricular chambers. **Medical Imaging, IEEE Transactions on**, Chicago, USA, v.14, n.1, p.56–64, 1995.

GRAU, V. et al. Improved watershed transform for medical image segmentation using prior information. **Medical Imaging, IEEE Transactions on**, [S.l.], v.23, n.4, p.447–458, 2004.

GREENSPAN, H.; RUF, A.; GOLDBERGER, J. Constrained Gaussian mixture model framework for automatic segmentation of MR brain images. **Medical Imaging, IEEE Transactions on**, [S.l.], v.25, n.9, p.1233–1245, 2006.

GUÉZIEC, A. et al. Cutting and stitching: converting sets of polygons to manifold surfaces. **Visualization and Computer Graphics, IEEE Transactions on**, California, USA, v.7, n.2, p.136–151, aug. 2001.

HABAS, P. A. et al. Atlas-based segmentation of the germinal matrix from in utero clinical MRI of the fetal brain. In: **Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008**. [S.l.]: Springer, 2008. p.351–358.

HACIHALILOGLU, I. et al. Bone Segmentation and Fracture Detection in Ultrasound Using 3D Local Phase Features. In: METAXAS, D. et al. (Ed.). **Medical Image Computing and Computer-Assisted Intervention ? MICCAI 2008**. [S.l.]: Springer Berlin Heidelberg, 2008. p.287–295. (Lecture Notes in Computer Science, v.5241).

HAN, X.; XU, C.; PRINCE, J. Fast numerical scheme for gradient vector flow computation using a multigrid method. **Image Processing, IET**, [S.l.], v.1, n.1, p.48–55, 2007.

HEIMANN, T. et al. Comparison and evaluation of methods for liver segmentation from CT datasets. **Medical Imaging, IEEE Transactions on**, Ilinois,USA, v.28, n.8, p.1251–1265, Aug. 2009.

HEIMANN, T.; MEINZER, H.-P. Statistical shape models for 3D medical image segmentation: a review. **Medical image analysis**, New York, USA, v.13, n.4, p.543–563, Aug. 2009.

KAINMÜLLER, D.; LANGE, T.; LAMECKER, H. Shape constrained automatic segmentation of the liver based on a heuristic intensity model. In: MICCAI WORKSHOP 3D SEGMENTATION IN THE CLINIC: A GRAND CHALLENGE, 2007. **Proceedings...** [S.l.: s.n.], 2007. p.109–116.

KASS, M.; WITKIN, A.; TERZOPOULOS, D. Snakes: active contour models. **International journal of computer vision**, [S.l.], v.1, n.4, p.321–331, 1988.

KAUS, M. et al. Adaptive Template Moderated Brain Tumor Segmentation in MRI. In: EVERS, H. et al. (Ed.). **Bildverarbeitung für die Medizin 1999**. [S.l.]: Springer Berlin Heidelberg, 1999. p.102–106. (Informatik aktuell).

KAZHDAN, M.; BOLITHO, M.; HOPPE, H. Poisson Surface Reconstruction. In: FOURTH EUROGRAPHICS SYMPOSIUM ON GEOMETRY PROCESSING, 2006, Aire-la-Ville, Switzerland, Switzerland. **Proceedings...** Eurographics Association, 2006. p.61–70. (SGP '06).

KOHN, L. T. et al. **To Err Is Human**:: building a safer health system. Washington, USA: National Academies Press, 1999.

LACHAUD, J.-O.; MONTANVERT, A. Deformable meshes with automated topology changes for coarse-to-fine three-dimensional surface extraction. **Medical Image Analysis**, Cedex,France, v.3, n.2, p.187–207, sep. 1999.

LAW, G. H. et al. Rapid generation of patient-specific anatomical models for usage in virtual environment. **Computer-Aided Design and Applications**, [S.l.], v.8, n.6, p.927–938, aug. 2011.

LEOW, W. K.; LI, R. The analysis and applications of adaptive-binning color histograms. **Computer Vision and Image Understanding**, New York, USA, v.94, n.1, p.67–91, 2004.

LEYMARIE, F.; LEVINE, M. D. Tracking deformable objects in the plane using an active contour model. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, [S.l.], v.15, n.6, p.617–634, 1993.

LORENSEN, W. E.; CLINE, H. E. Marching cubes: a high resolution 3d surface construction algorithm. In: ACM SIGGRAPH COMPUTER GRAPHICS, 1987. **Proceedings...** [S.l.: s.n.], 1987. v.21, n.4, p.163–169.

MCINEMEY, T.; TERZOPOULOS, D. Topology adaptive deformable surfaces for medical image volume segmentation. **Medical Imaging, IEEE Transactions on**, Cambridge, England, v.18, n.10, p.840–850, oct 1999.

NEALEN, A. et al. Laplacian mesh optimization. In: COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES IN AUSTRALASIA AND SOUTHEAST ASIA, 4., 2006, New York, USA. **Proceedings...** [S.l.: s.n.], 2006. p.381–389.

OKADA, T. et al. Automated segmentation of the liver from 3D CT images using probabilistic atlas and multilevel statistical shape model. **Academic Radiology**, Washington,USA, v.15, n.11, p.1390–1403, 2008.

PLUEMPITIWIRIYAWEJ, C. et al. STACS: new active contour scheme for cardiac mr image segmentation. **Medical Imaging, IEEE Transactions on**, California, USA, v.24, n.5, p.593–603, May 2005.

REN, X.; MALIK, J. Learning a classification model for segmentation. In: COMPUTER VISION TECHNIQUES, 2003. **Proceedings...** [S.l.: s.n.], 2003. p.10–17.

ROTHER, C.; KOLMOGOROV, V.; BLAKE, A. Grabcut: interactive foreground extraction using iterated graph cuts. **ACM Transactions on Graphics (TOG)**, Washington, USA, v.23, n.3, p.309–314, Sep. 2004.

SETHIAN, J. A. **Level set methods and fast marching methods**: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. [S.l.]: Cambridge university press, 1999. v.3.

SEYMOUR, N. E. et al. Virtual reality training improves operating room performance: results of a randomized, double-blinded study. **Annals of surgery**, Virginia, USA, v.236, n.4, p.458, 2002.

SHI, J.; MALIK, J. Normalized cuts and image segmentation. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, [S.l.], v.22, n.8, p.888–905, 2000.

SORKINE, O. et al. Laplacian surface editing. In: EUROGRAPHICS/ACM SIGGRAPH SYMPOSIUM ON GEOMETRY PROCESSING, 2004., 2004, New York, USA. **Proceedings...** [S.l.: s.n.], 2004. p.175–184.

STRAKA, M. et al. 3D Watershed Transform Combined with a Probabilistic Atlas for Medical Image Segmentation. In: MIT 2003, 2003. **Proceedings...** [S.l.: s.n.], 2003. p.1–8.

TOULSON, D.; BOYCE, J. F. Segmentation of MR images using neural nets. In: **British Machine Vision**. [S.l.: s.n.], 1991. p.284–292.

WELS, M. et al. A Discriminative Model-Constrained Graph Cuts Approach to Fully Automated Pediatric Brain Tumor Segmentation in 3-D MRI. In: METAXAS, D. et al. (Ed.). **Medical Image Computing and Computer-Assisted Intervention MICCAI 2008**. [S.l.]: Springer Berlin Heidelberg, 2008. p.67–75. (Lecture Notes in Computer Science, v.5241).

WILLAERT, W. et al. Patient-specific simulation for endovascular procedures: qualitative evaluation of the development process. **The International Journal Of Medical Robotics And Computer Assisted Surgery**, Maryland, USA, v.6, n.2, p.202–210, june 2010.

WU, Z.; LEAHY, R. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, [S.l.], v.15, n.11, p.1101–1113, 1993.

XU, C.; PRINCE, J. L. Snakes, shapes, and gradient vector flow. **Image Processing, IEEE Transactions on**, [S.l.], v.7, n.3, p.359–369, 1998.

ZHOU, J. et al. Liver workbench: a tool suite for liver and liver tumor segmentation and modeling. In: **Advances in Bio-Imaging**: from physics to signal understanding issues. [S.l.]: Springer, 2012. p.193–208.

APPENDIX A GERAÇÃO DE MALHAS PARA PACIENTES ESPECIFICOS

Resumo da Dissertação em Português

Simuladores de cirurgia se tornaram essenciais para treinamento de procedimentos. Pesquisas indicam que o uso geral de simuladores permitem a aquisição, melhoria e manutenção de habilidades complexas sem o risco de infligir danos ao paciente real (SEYMOUR et al., 2002). Mais recentemente, o uso de simuladores voltados para pacientes específicos se tornou uma possibilidade, graças aos avanços principalmente no campo de processamento de imagens, permitindo a incorporação de imagens médicas de um paciente específicos (obtidas via ressonância magnética ou tomografia) em técnicas existentes (WILLAERT et al., 2010). Porém, a maior parte dos ambientes de simulação ainda utilizam modelos anatômicos genéricos para a representação dos órgãos do paciente. Essa simplificação ainda é mais comum em órgãos de tecido mole, como os encontrados na região abdominal, e é justificada pela dificuldade envolvida em se gerar malhas poligonais a partir das imagens médicas. Isso é causado por limitações nos processos de aquisição das imagens médicas. Órgãos da região abdominal são deformáveis, exibindo formas complexas. Além disso, a intensidade luminosa desses órgãos é similar, o que causa regiões de contato entre os órgãos a não ter bordas definidas. Portanto, esses simuladores não são adequados para o planejamento de cirurgias.

Nessa tese, nós apresentamos uma solução unificada para a segmentação do volume de imagens médicas, e para a geração da malha poligonal para obtermos malhas de pacientes específicos. Nossa técnica supera problemas existentes em outros algoritmos de segmentação de imagens, e nos permite obter malhas plausíveis, e altamente similares ao órgão do paciente.

Destacamos portanto, as seguintes contribuições desta dissertação:

Nossa abordagem é robusta, encontrando limites plausíveis para o órgão, mesmo quando estes não são visíveis. A malha resultante apresenta uma boa qualidade em sua amostragem, permitindo a extração de formas irregulares comumente encontradas na anatomia humana. Nossa técnica também evita artefatos como auto-intersecção da malha triangular, problema encontrado em outros algoritmos do estado da arte.

- *um algoritmo de segmentação com baixo custo computacional*. Nossa técnica utiliza múltiplas malhas com uma topologia especial inicialmente esférica, que através de cálculos pré-fatorados, permitem um menor custo computacional quando comparando a técnica a outros algoritmos da mesma classe;

- *uma técnica que gera malhas sem auto-intersecção e sem convergência para mínimos locais*. Através de heurísticas geométricas, evitamos que as malhas se interseccionem ao longo da deformação. Alem disso, através do uso de buscas em distâncias e limitadores na deformação, garantimos que a convergência das malhas não seja apenas local, o que contribui para uma melhor amostragem do órgão desejado ;

- *obtermos uma malha para o órgão do paciente*. Como utilizamos malhas geométricas para a segmentação, ao término da técnica, pode-se optar por utilizar as malhas obtidas no processo da deformação. Nossa abordagem é robusta, encontrando limites plausíveis para o órgão, mesmo quando estes não são visíveis. A malha resultante apresenta uma boa qualidade em sua amostragem, permitindo a extração de formas irregulares comumente encontradas na anatomia humana;

Para obter as contribuições citadas, foi necessário realizar um estudo sobre segmentação de uma coleção de imagens médicas, denominada também volumes médicos, que será brevemente apresentada na próxima sessão.

A.1 Segmentação de Volumes

Métodos para segmentação de volumes médicos podem ser classificados nas seguintes categorias: manuais, semiautomáticos e automáticos.

A.1.1 Segmentação manual

Métodos nessa classe são precisos mas consomem tempo para execução. É necessário que pessoas treinadas desenhem contornos manualmente. Por depender do entendimento humano dos limites do órgão, estes métodos estão mais suscetíveis a variabilidade de uma segmentação entre variados observadores (KAUS et al., 1999). O "valor verdade" portanto, é obtido através da média entre várias segmentações efetuadas manualmente. Avalia-se que o tempo para uma segmentação de um volume de 1500 imagens por um profissional treinado leve de duas a quatro horas (STRAKA et al., 2003).

A.1.2 Segmentação Automática

Métodos nessa categoria não dependem de interação humana. Para alcançar essa independência, são necessários conjuntos de segmentações modelo para cálculos estatísticos e algoritmos de treinamento. Isso cria uma dependência que impede sua aplicação em casos clínicos, visto que a qualidade do resultado é fortemente atrelado a qualidade das amostras para o treinamento do método.

A.1.3 Segmentação Semiautomática

Métodos classificados como semiautomáticos incorporam interação mínima por parte do usuário, geralmente na forma de parâmetros de inicialização. Por requisitarem baixa interação e não depender de usuários especificamente treinados, são considerados os mais propícios para uso clínico. Nessa tese, apresentaremos uma abordagem pertencente a esse grupo de técnicas, na próxima sessão.

Figure A.1: Passos para a segmentação e obtenção da malha de um fígado

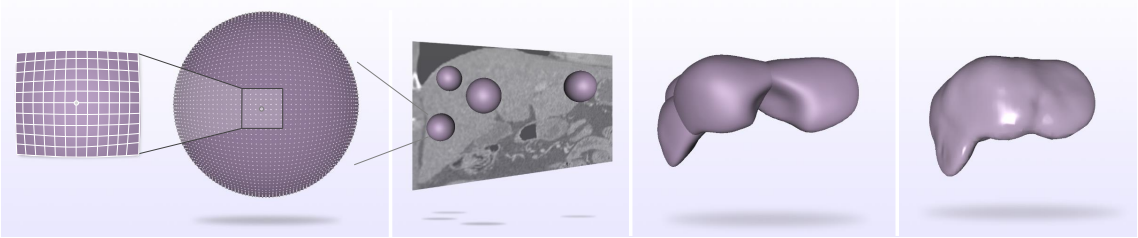

Figure A.2: Source:original image

A.2 Segmentação e Obtenção de Malhas para Pacientes Específicos

O método proposto utiliza uma inicialização por parte do usuário: o volume é exibido, e o usuário deve clicar um número de vezes em posições distintas dentro do órgão que ele deseja obter a malha. Esse número varia de acordo com o tamanho do órgão. Nos casos de testes realizados com o fígado, definimos como sendo quatro cliques.

A posição dos cliques é dada como o valor de inicialização para uma malha esférica especial. No caso de quatro cliques, quatro malhas esféricas serão utilizadas. Essas malhas esféricas irão crescer, com seus vértices sendo projetados na direção das normais. Para evitar auto-intersecção, limitações geométricas como interpolação e preservação dos laplacianos são utilizadas (ZHOU et al., 2012; DING; LEOW; VENKATESH, 2009; DING et al., 2009). A medida que as malhas se aproximam uma das outras, utilizamos a mesma abordagem de verificação de intersecção para limitar as buscas entre esses vértices de duas malhas diferentes. Essa abordagem evita o vazamento da malha para fora do órgão em regiões com contornos irregulares.

Esse processo é realizado interativamente, com passos de projeção, de checagem de intersecção, e deformação com preservação laplaciana. A sequencia desses passos é apresentada na Figura A.1. A cada interação, a malha é atualizada com uma interpolação entre seu estado atual e as novas posições encontradas. O critério de parada é definido ou como um número máximo de interações ou pela média das diferenças entre as posições antes da deformação e após a deformação ser inferior a um determinado valor de distância.

Após a convergência, é possível se gerar uma mascara de segmentação a partir dos vóxels, que são o equivalente a um pixel em um caso de volume. A partir dessa mascara, um algoritmo de triangulação pode ser utilizado para se obter uma malha única (KAZHDAN; BOLITHO; HOPPE, 2006), ou uma abordagem diferente pode ser utilizada: visto que as esferas usadas na segmentação já são malhas, um algoritmo de costura pode ser aplicado para fundi-las (GUÉZIEC et al., 2001).

A.3 Avaliação e Resultados

Para avaliação, comparamos nossa técnica com outros três métodos de segmentação de volumes médicos que não utilizam aprendizagem de maquina ou dependam de treinamento (ROTHER; KOLMOGOROV; BLAKE, 2004; SETHIAN, 1999; DING et al., 2009). Para o cenário de teste, optamos pela obtenção de malhas do fígado, por ser um órgão que apresenta grande similaridade de intensidades com os órgãos que estão ao seu redor. Alem disso, o fígado pode apresentar grande variabilidade de formato, por ser um órgão defor-

mável. Como métricas para a avaliação, comparamos os resultados de mascaras binárias obtidas pelos algoritmos testados contra um resultado-verdade, obtido no banco de dados disponibilizados por uma competição anterior de segmentação de fígados (HEIMANN et al., 2009). O conjunto de dados consiste em oito volumes diferentes de tomografia, acompanhados das respectivas segmentações do fígado de cada volume, por meio de um volume binário exibindo somente a região considerada como pertencente ao fígado do indivíduo.

Como métricas para a avaliação, consideramos:

- *a distância entre a superfície da segmentação e sua equivalente na segmentação verdadeira.* Quanto menor a média dessas distâncias e sua variação, mais exata a segmentação.

- *a diferença volumétrica das segmentações.* Esse valor indica quanto do volume foi devidamente coberto pelo algoritmo. A combinação de dados da proximidade entre as superfícies e cobertura volumétrica é um bom indicador da qualidade da segmentação realizada (HEIMANN et al., 2009).

- *tempo até convergência.* Avaliamos o custo computacional dos algoritmos testados para saber o tempo necessário para se obter uma segmentação.

Os resultados obtidos neste trabalho mostraram uma maior precisão na segmentação obtida comparado aos outros algoritmos testados, graças a eliminação de auto-intersecções e diminuição do vazamento da segmentação para outros órgãos. Além disso, tivemos uma grande diminuição de custo computacional, quando comparado aos outros métodos de segmentação. Nossa implementação utiliza somente um núcleo do processador, havendo a possibilidade de paralelização para um desempenho ainda melhor.

A.4 Conclusão

Nesse trabalho, nós apresentamos um algoritmo para a obtenção de malhas a partir de dados da anatomia de um paciente específico. Nosso algoritmo se baseia em múltiplas malhas esféricas deformáveis, sem a dependência de dados para treinamento ou cálculos probabilísticos. O método necessita de interação mínima, e evita auto-intersecções locais e globais.

Graças ao uso de múltiplas malhas esféricas, a segmentação obtida possui uma boa amostragem, evitando uma convergência local, e possibilitando uma melhor cobertura do volume e uma maior proximidade dos limites reais do órgão quando comparado a outros métodos concorrentes. Também obtivemos um melhor desempenho, com um tempo para convergência inferior aos métodos concorrentes.

Como trabalhos futuros, sugerimos estudar como inicializar as malhas automaticamente dentro do órgão de interesse. Dessa maneira, o método passa a ser completamente automático e independente da supervisão do usuário para iniciar a técnica. Acreditamos que o conhecimento de distâncias antropométricas combinado com embriologia e a padronização da posição do paciente no ambiente de aquisição das imagens podem apresentar informações úteis para a solução desse problema.