UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Mandible and skull segmentation in cone beam computed tomography data

Oscar Alonso Cuadros Linares

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Oscar Alonso Cuadros Linares

Mandible and skull segmentation in cone beam computed tomography data

Doctoral dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. João do Espírito Santo Batista Neto

USP – São Carlos
February 2018

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

C961m Cuadros Linares, Oscar Alonso
Mandible and skull segmentation in cone beam
computed tomography data / Oscar Alonso Cuadros
Linares; orientador João do Espírito Santo Batista
Neto. -- São Carlos, 2018.
134 p.

Tese (Doutorado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional) --
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2018.

1. Cone beam computed tomography. 2. Bone
segmentation. 3. Super-voxels. 4. Graph clustering.
5. Mandible and skull. I. Batista Neto, João do
Espírito Santo, orient. II. Título.

Oscar Alonso Cuadros Linares

**Segmentação da mandíbula e o crânio em tomografia
computadorizada de feixe cônico**

Tese apresentada ao Instituto de Ciências
Matemáticas e de Computação – ICMC-USP,
como parte dos requisitos para obtenção do título
de Doutor em Ciências – Ciências de Computação e
Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e
Matemática Computacional

Orientador: Prof. Dr. João do Espírito Santo
Batista Neto

USP – São Carlos
Fevereiro de 2018

to my wife, Aurea.

ACKNOWLEDGEMENTS

First and foremost I would like to thank my wife for her constant love and unyielding support throughout my life.

I would like to express my sincere gratitude to my advisor Prof. Dr. João do Espírito Santo Batista Neto for the continuous support during my Ph.D. research.

My sincere thanks also goes to Prof. Dr. Bernd Hamann, for offering me the internship opportunities at The Institute for Data Analysis and Visualization (IDAV), University of California, Davis.

I would like to acknowledge Prof. Dr. Dirceu Raveli and his team at UNESP – Araraquara for all their support.

I would also like to thank my parents Oscar and Ruth, my sister Alejandra, and my aunt Luz for their wise counsel and constant love. You are always there for me.

Besides, I would like to acknowledge friends for all your support and make my time at USP São Carlos, and UC Davis a lot more fun.

I am grateful for the funding sources that allowed me to pursue my doctoral degree: Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), process numbers: 2012/24036-1 and 2015/12771-7

“We are not simply in the universe, we are part of it. We are born from it. One might even say that the universe has empowered us, here in our small corner of the cosmos, to figure itself out. And we have only just begun.”

(Neil deGrasse Tyson)

“The nobility of the human race lies on our drive to know, and our uniqueness as a species is reflected in the success we have achieved, after millennia of effort, in deciphering the puzzle that is nature.”

(Leonard Mlodinow)

“Far, far into the future, protons and neutrons will decay, matter will disappear, and the universe will approach a state of maximum simplicity and symmetry.”

(Lawrence M. Krauss)

ABSTRACT

OSCAR ALONSO CUADROS LINARES. **Mandible and skull segmentation in cone beam computed tomography data**. 2018. 134 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

Cone Beam Computed Tomography (CBCT) is a medical imaging technique routinely employed for diagnosis and treatment of patients with cranio-maxillo-facial defects. CBCT 3D reconstruction and segmentation of bones such as mandible or maxilla are essential procedures in orthodontic treatments. However, CBCT images present characteristics that are not desirable for processing, including low contrast, inhomogeneity, noise, and artifacts. Besides, values assigned to voxels are relative Hounsfield Units (HU), unlike traditional Computed Tomography (CT). Such drawbacks render CBCT segmentation a difficult and time-consuming task, usually performed manually with tools designed for medical image processing. We introduce two interactive two-stage methods for 3D segmentation of CBCT data: i) we first reduce the CBCT image resolution by grouping similar voxels into super-voxels defining a graph representation; ii) next, seeds placed by users guide graph clustering algorithms, splitting the bones into mandible and skull. We have evaluated our segmentation methods intensively by comparing the results against ground truth data of the mandible and the skull, in various scenarios. Results show that our methods produce accurate segmentation and are robust to changes in parameter settings. We also compared our approach with a similar segmentation strategy and we showed that it produces more accurate segmentation of the mandible and skull. In addition, we have evaluated our proposal with CT data of patients with deformed or missing bones. We obtained more accurate segmentation in all cases. As for the efficiency of our implementation, a segmentation of a typical CBCT image of the human head takes about five minutes. Finally, we carried out a usability test with orthodontists. Results have shown that our proposal not only produces accurate segmentation, as it also delivers an effortless and intuitive user interaction.

Keywords: Cone Beam Computed Tomography, Bone Segmentation, Super-voxels, Graph Clustering, Mandible, Skull.

RESUMO

OSCAR ALONSO CUADROS LINARES. **Segmentação da mandíbula e o crânio em tomografia computadorizada de feixe cônico**. 2018. 134 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

Tomografia Computadorizada de Feixe Cônico (TCFC) é uma modalidade para obtenção de imagens médicas 3D do crânio usada para diagnóstico e tratamento de pacientes com defeitos crânio-maxilo-faciais. A segmentação tridimensional de ossos como a mandíbula e a maxila são procedimentos essenciais em tratamentos ortodônticos. No entanto, a TCFC apresenta características não desejáveis para processamento digital como, por exemplo, baixo contraste, inhomogeneidade, ruído e artefatos. Além disso, os valores atribuídos aos voxels são unidades de Hounsfield (HU) relativas, diferentemente da Tomografia Computadorizada (TC) tradicional. Esses inconvenientes tornam a segmentação de TCFC uma tarefa difícil e demorada, a qual é normalmente realizada por meio de ferramentas desenvolvidas para processamento digital de imagens médicas. Esta tese introduz dois métodos interativos para a segmentação 3D de TCFC, os quais são divididos em duas etapas: i) redução da resolução da TCFC por meio da agrupamento de voxels em super-voxels, seguida da criação de um grafo no qual os vértices são super-voxels; ii) posicionamento de sementes pelo usuário e segmentação por algoritmos de agrupamento em grafos, o que permite separar os ossos rotulados. Os métodos foram intensamente avaliados por meio da comparação dos resultados com padrão ouro da mandíbula e do crânio, considerando diversos cenários. Os resultados mostraram que os métodos não apenas produzem segmentações precisas, como também são robustos a mudanças nos parâmetros. Foi ainda realizada uma comparação com um trabalho relacionado, gerando melhores resultados tanto na segmentação da mandíbula quanto a do crânio. Além disso, foram avaliadas TCs de pacientes com ossos faltantes e quebrados. A segmentação de uma TCFC é realizada em cerca de 5 minutos. Por fim, foram realizados testes com usuários ortodontistas. Os resultados mostraram que nossa proposta não apenas produz segmentações precisas, como também é de fácil interação.

Palavras-chave: Tomografia Computadorizada de Feixe Cônico, Segmentação de ossos, Super voxels, Agrupamento em grafos, Mandíbula, Crânio.

LIST OF FIGURES

Figure 1	– Two samples of our CBCT dataset. A typical CBCT image is $400 \times 400 \times 554$ (male and female patients), with an isotropic voxel size of 0.4 mm, acquired with an i-Cat scanner. Upper row: sample 1; Lower row: sample 2. From left to right: sagittal, coronal and axial sections, respectively. Low contrast and sheet-like bones, which are difficult to segment, are highlighted with a yellow arrow. Notice that the mandible is anatomically separated from other bones, see yellow arrows in (a) and (b). In contrast, there is no natural division between the maxilla and the skull base, thus specialists separate them by an axial straight cut, see red lines in (b) and (e).	31
Figure 2	– Segmentation examples obtained during the doctoral period. From left to right: seeds, an axial sample of resulting segmentation and 3D rendering, respectively. First row: mandible segmentation example, and second row: skull segmentation, detail of the maxilla.	33
Figure 3	– Retina cellular-level segmentation obtained with method MSM.	34
Figure 4	– Helical fan-beam CT scanner design showed in two positions.	38
Figure 5	– Example of a CT scan of a human head (PIXMEO-SARL, 2016). Initially, an axial slice image series is reconstructed. Next, coronal and sagittal are calculated by interpolation of the axial stack. This process is called Multi-planar reformatting. CT sample of resolution $(512 \times 512 \times 460)$ anisotropic voxel dimensions $(0.4883 \times 0.4883 \times 0.7)mm$ (SEERAM, 2015).	39
Figure 6	– Example of beam hardening (yellow frame). Axial CT section of a human head, detail of lower teeth. Slice extracted from a CT sample with resolution $(512 \times 512 \times 130)$ and anisotropic voxel dimensions $(1 \times 1 \times 3)mm$ (SHARP PAOLO ZAFFINO; RAUDASCHL, 2016)	40
Figure 7	– Anisotropic and isotropic resolution in CT. Isotropic voxels have equal dimensions (perfect cubes) whereas anisotropic voxels do not have equal dimensions.	41
Figure 8	– Representation of Cone beam computed tomography technique.	42
Figure 9	– Free-hand illustration of human skull bones (NETTER, 2014)	44
Figure 10	– CBCT data: coronal, axial and sagittal sections of human skull bones, respectively. Bones indicated by red arrows are, from an image segmentation perspective, big challenges.	45

Figure 11 – Statistical atlas building process. An average image is computed from a set of representative images and a reference image. During the process are applied affine registration and deformable registration algorithms to align all images with the reference image (GUIMOND; MEUNIER; THIRION, 2000).	49
Figure 12 – Input CT data and segmentation produced by the method <i>Automated Segmentation of Dental CBCT Image with Prior-guided Sequential Random Forests</i> (WANG <i>et al.</i> , 2016). Notice that the interior part of the mandible (low-density bone) are not included in the segmentation.	51
Figure 13 – Example of CT data and statistical atlas used in the method <i>Automatic Segmentation of Mandibles in Low-Dose CT-Data</i> (LAMECKER <i>et al.</i> , 2006).	52
Figure 14 – Extraction of Upper Mandible-arch: distance and positions constraints to consider connected components as a part of the upper mandible arch (SPAMPINATO <i>et al.</i> , 2012).	54
Figure 15 – CT data used in the method <i>Automatic 3D Segmentation of Mandible for Assessment of Facial Asymmetry</i> (SPAMPINATO <i>et al.</i> , 2012), and a 3D view a segmented mandible.	55
Figure 16 – Mandibular segmentation produced by the method <i>Automatic Extraction of Mandibular Nerve and Bone from Cone-Beam CT Data</i> (KAINMUELLER <i>et al.</i> , 2009). Color represents distance to ground truth data.	57
Figure 17 – Segmentation of the anterior wall of the maxilla using the method <i>3D Segmentation of Maxilla in Cone-beam Computed Tomography Imaging Using Base Invariant Wavelet Active Shape Model on Customized Two-manifold Topology</i> (CHANG <i>et al.</i> , 2013). Training shape with landmarks (left), respective patch (middle) and segmentation (right).	59
Figure 18 – Hippocampus segmentation obtained with the method <i>Supervoxel-Based Segmentation of Mitochondria in EM Image Stacks with Learned Shape Features</i> (LUCCHI <i>et al.</i> , 2012). Hippocampus demarcated in red.	61
Figure 19 – Segmentation of prostate generated with the method <i>Prostate MRI Segmentation Using Learned Semantic Knowledge And Graph Cuts</i> (MAHAPATRA; BUHMANN, 2014). Ground truth segmentation represented with green line whereas red line represents the proposed method.	63
Figure 20 – Heart blood pool segmentation produced with the method <i>An Effective Interactive Medical Image Segmentation Method Using Fast GrowCut</i> (ZHU <i>et al.</i> , 2014).	64
Figure 21 – Anatomical changes in the mandible with age.	65
Figure 22 – An example of low-density bone. Several voxels of the mandible internal region (yellow arrow) have HU values lower than HU values of tissue. The voxel value indicated by the yellow arrow is 154 whereas the value indicated by the red arrow is 240.	68

Figure 23 – Example of Lung’s Bronchial arteries.	69
Figure 24 – Shee-like bones in CBTC data are hard to extract due to contrast problem. Results obtained with the Sheetness score are shown in (c), successfully enhancing bones in the data, especially thin ones.	71
Figure 25 – Entire bone structure, including thin bones, extracted with the proposed threshold-based filter.	73
Figure 26 – SLIC algorithm convergence process. Pixels are iteratively associated to their nearest super-pixel centers. After each iteration the centers are recomputed.	74
Figure 27 – SUTP algorithm process after 5 iterations. Boundary Pixels are iteratively moved to adjacent super-pixels.	75
Figure 28 – SUSV convergence after 5 iterations. Notice that in this example the input image (a) is the bony structure previously segmented with the method described in Section 4.1.	77
Figure 29 – Example of a network with 7 vertex and 9 edges.	79
Figure 30 – Example of two complex networks.	80
Figure 31 – Graph Partitioning of a graph with nine vertex and fourteen edges using the Fiedler vector $\mathbf{v}_2 = (0.21, 0.41, 0.28, 0.50, -0.19, -0.37, -0.17, -0.32, -0.35)$ and corresponding $\mathbf{s} = (1, 1, 1, 1, -1, -1, -1, -1, -1)$. Thus $g_1 = (1, 2, 3, 4)$ and $g_2 = (5, 6, 7, 8)$. The red dashed line shows the solution of the minimum cut on the graph.	85
Figure 32 – Network with three communities within dashed red circles. The goal of community detection algorithms is to group vertices so that the number of internal connections (black edges) outnumbers external connections (blue edges).	88
Figure 33 – Multilevel optimization of modularity algorithm. Each iteration is performed in two steps: 1) modularity optimization by moving vertices from their communities to adjacent ones, 2) creating a new graph where vertices within communities found in the first step are aggregated to form a single vertex. In this example two communities were found after two iterations.	90
Figure 34 – Mandible and Skull Segmentation with Super-voxels and Laplacian-based Graph Clustering (MSSL): a) automatic segmentation of bony structures, followed by grouping of similar voxels into super-voxels and a graph representation where each vertex is a super-voxel; b) interactive segmentation with seed placement (brush strokes) on the bone of interest. Segmentation can be improved by either adding or removing seeds interactively.	94

Figure 35 – Visualization of adjacent super-voxels for MSSL method. We create an edge between vertices I (green) and J (blue) only if super-voxels Ω_I and Ω_J are adjacent (similar to Moore neighborhood). Notice that adjacent background super-voxels (yellow) do not belong to the ROI (bony structure), thus they are excluded from the graph-building process. As a result, any super-voxel Ω_I has ≈ 26 neighbors. However, most of them are eliminated, thus the degree of the vertex I is far lower than 26. In this example, the super-voxel under evaluation (green) is the same in (a), (b), (c).	96
Figure 36 – Mandible segmentation with Super-voxels and Modularity optimization (MSM): a) preprocessing. Similar voxels are grouped into super-voxels; b) Interactive segmentation. User selects the mandible by placing seeds on it (brush strokes). Next, a bounding box is computed to delimit the segmentation area. We then build a graph where each vertex is a super-voxel inside the bounding box. Segmentation can be improved by either adding or removing seeds interactively.	97
Figure 37 – Visualization of neighboring super-voxels in method MSM. An edge between vertices I (blue) and J (yellow) exists only if their respective super-voxels centroids are inside of a cubic region of length (in voxels) r . The set of super-voxels, whose centroids are inside the cube, is defined as V_I . Notice the number of neighboring super-voxels increases in a cubic form regarding to r . This neighborhood can be seen as an adaptive-size version of the Moore neighborhood.	99
Figure 38 – Axial view of the anterior wall of the maxilla. In method MSSL an accurate placement of seeds is not needed (red strokes). This is useful because the maxilla is mainly composed of sheet-like bones. Notice that red brush strokes surpass thin-bones borders.	100
Figure 39 – Laplace probability density function generated with Equation 6.6, we set $x = [-10, +10]$, $\mu = 0$ and $b = 1$	102
Figure 40 – Upper row: segmentation of skull of a male patient (including maxilla) with method MSSL. Lower row: segmentation of mandible of a female patient. From left to right: axial slices, seed placement process and 3D view of final segmentation. Red brush strokes indicate the bone of interest (foreground), whereas green strokes label undesired bones (background). Brush strokes do not have to be precise and may be drawn beyond bone borders, as long as they do not touch other bones. More example results are exhibit in Section A.	105

- Figure 41 – Influence of parameters t , σ and l on segmentation quality. The curves represent the mean Jaccard index of the four samples (720 segmentation results). We measured the segmentation quality by evaluating several combinations of the parameters mentioned, for instance: $t = 100$, $\sigma = 0.5$ and $t = [100 - 300]$. The most influential parameter is the threshold t (a) with best quality obtained for $t > 150$. In a lesser scale, variations in σ and l also affect the segmentation quality, seen as minor fluctuations in the curves. 106
- Figure 42 – Axial view of skull segmentation of the maxilla for different values of σ , with $t = 200$ and $l = 10$. Sheet-like bones, such as the anterior and posterior walls of the sinus (yellow arrows in (a)), are accurately segmented irrespectively to changes in σ values. J is the Jaccard index. 106
- Figure 43 – Axial view for skull segmentation of the maxilla for different values of t , with $\sigma = 0.8$ and $l = 10$. Changes in threshold t affect segmentation quality, unlike variations in σ , see Fig. 42. The retroantral fat pad (yellow arrow in (e)) is not entirely segmented for $t \geq 250$ 107
- Figure 44 – Influence of parameters t , σ , and l on mandible segmentation quality. We produced 180 segmentation of the mandible for four CBCTs (hence, a total of 720 segmentations). Curves show the mean Jaccard index for the four datasets evaluated. We measured segmentation quality by evaluating several combinations of the parameters, for instance: $t = 100$, $\sigma = 0.5$ and $t = [100 - 300]$, and so on. Like skull segmentation (Fig. 41), the most influential parameter is the threshold t (a), with best results obtained for $150 < t < 300$. However, σ and l also affect, marginally, the segmentation quality, seen as fluctuations in the curves. Local extrema, is noticeable for $t < 150$ (a); these are explained by the presence of noise in segmentation (Fig. 46.a). Noise can be interactively eliminated with extra brush strokes and, hence, improve segmentation quality. 108
- Figure 45 – Axial view of the mandible segmentation (condyle and coronoid process), with $t = 200$ and $l = 10$. Results obtained with parameter σ at different scales and best results for $\sigma \geq 0.7$. Some noise is still present (yellow arrows in (e)), but can be interactively eliminated with additional brush strokes. . . 108
- Figure 46 – Axial view of mandible segmentation (condyle and coronoid process), using $\sigma = 0.8$ and $l = 10$. Results shown for different values of threshold t . Best quality is achieved for $100 < t < 300$. Higher values of t do not yield proper segmentations of thin-bone regions, e.g., the coronoid process (yellow arrow in (e)). Noise can still be seen in the final segmentation (yellow arrows in (a)) for $t < 100$, but can be interactively eliminated with brush strokes. 109

Figure 47 – Axial view of mandible segmentation for Zhu et al.’s and our method. Two seeding setups, (a) and (d), with red strokes indicating the desired bone and green strokes the undesired regions; (b) and (e) our results, for seeding conditions as defined in (a) and (d), respectively. Likewise, (c) and (f) are Zhu et al.’s results for the same seeding setups. Our method uses less brush strokes, and Zhu et al.’s results exhibit “leaks” (yellow frames) in low-density areas, even with seed placement 2.	110
Figure 48 – Sagittal view of skull segmentation for Zhu et al.’s and our approach. Seeding setup is shown in (a), with red indicating the desired bone and green undesired regions. Our method properly segments thin bones, even when brush strokes cover adjacent tissue types, see yellow frames in (b). Zhu et al.’s method produces “leaks” in soft-tissue regions and air (cranial cavities).	111
Figure 49 – Skull segmentation. CT of patient with facial deformities after surgical repair. CT sample of $512 \times 512 \times 361$ resolution and anisotropic voxel spacing of 0.42mm, 0.42mm and 0.70mm.	111
Figure 50 – Segmentation of mandible. Patient with missing teeth. CT sample of resolution $512 \times 512 \times 460$ and anisotropic voxel spacing of 0.48mm, 0.48mm and 0.70mm.	112
Figure 51 – Usability testing: skull segmentation of CBCT 1 by user A, seed placement, and the respective result, in the first row. Mandible segmentation of CBCT 1 by user B in the second row. User A finished the segmentation process after three iterations, whereas user B finished it after only one iteration.	113
Figure 52 – Segmentation of mandible obtained with method MSM on 4 CBCT images. From left to right: seed placement in the first column, our result in the second column, and ground truth in the third column. Seed placement on thin structures is quite complicated, for example the coronoid process in (d). Besides separating touching lower an upper teeth also needs more precise seeds (j).	114
Figure 53 – Retina cellular-level segmentation obtained with method MSM. From left to right: seed placement in the first column; our results in the second column and the ground truth in the third column.	116
Figure 54 – Retina cellular-level segmentation: before (a) and after (b) application of the morphological erosion and dilation filter. H is the Hausdorff distance between segmentation and ground truth (c). A close-up view of erosion process within the yellow frame.	116
Figure 55 – From top to bottom: 1) a regular mandible, 2) a skull after surgical repair of facial deformity, and 3) a mandible with missing teeth.	133
Figure 56 – Segmentation obtained with method MSM, images from Berkeley dataset (MARTIN <i>et al.</i> , 2001).	134

LIST OF TABLES

Table 1 – Hounsfield Unit (HU) of common substances.	40
Table 2 – Theoretical properties of the Descoteaux’s ratios.	71
Table 3 – Parameter ranges evaluated	104
Table 4 – Parameter values for most accurate segmentation	104
Table 5 – Mean and standard (std) deviations of Jaccard index and Dice coefficient for skull segmentation of all datasets.	106
Table 6 – Mean and standard (std) deviations of Jaccard index and Dice coefficient of mandible segmentation.	107
Table 7 – Comparing our method with Zhu et al.’s in a mandible segmentation scenario. Values are the Jaccard similarity with respect to ground truth.	110
Table 8 – Jaccard value achieved by users A and B in a mandible and skull segmentation test	112
Table 9 – Mandible segmentation accuracy obtained with method MSM	115
Table 10 – Segmentation of cat retina accuracy ratios obtained with method MSM. . . .	115

LIST OF ABBREVIATIONS AND ACRONYMS

CBCT	Cone Beam Computed Tomography
CT	Computed Tomography
EM	Electron Microscopy
GP	Graph Partitioning
GT	Ground Truth
HU	Hounsfield Unit
LBP	Local Binary Pattern
MOM	Multilevel Optimization of Modularity
MRF	Markov Random Field
MRI	Magnetic Resonance Imaging
MSM	Mandible segmentation with Super-voxels and Modularity optimization
MSSL	Mandible and Skull segmentation with Super-voxels and Laplacian-based graph clustering
PCA	Principal Component Analysis
PSF	Parabolic Structuring Function
RFC	Random Forest Classifier
ROI	Region of Interest
SLIC	Simple Linear Iterative Clustering
SUSV	Speeded-up Super Voxels
SUTP	Speeded-up Turbo Pixels
SVM	Support Vector Machine
TCFC	Tomografia Computadorizada de Feixe Cônico
VOC	Voice of the Costumer
XR	X-Ray

CONTENTS

1	INTRODUCTION	29
1.1	Aim and Research Hypotheses	32
1.1.1	<i>Research Hypotheses</i>	32
1.2	Contributions	33
1.3	Publications	34
2	CONE BASED COMPUTED TOMOGRAPHY	37
2.1	Computed Tomography Fundamentals	37
2.1.1	<i>Technical Notes on Computed Tomography</i>	39
2.1.1.1	<i>CT Image Quality</i>	40
2.1.2	<i>CT Radiation Dose</i>	41
2.2	Cone Based Computed Tomography Fundamentals	42
2.2.1	<i>CBCT Pros and Cons</i>	43
2.3	Human Skull Anatomy Basics	43
2.4	Chapter Notes	45
3	RELATED WORK	47
3.1	Mandible and Skull Segmentation	49
3.1.1	<i>Automated Segmentation of Dental CBCT Image with Prior-guided Sequential Random Forests</i>	49
3.1.2	<i>Automatic Segmentation of Mandibles in Low-Dose CT-Data</i>	51
3.1.3	<i>Automatic 3D Segmentation of Mandible for Assessment of Facial Asymmetry</i>	53
3.1.4	<i>Automatic Segmentation of the Mandible in Cone-Beam Computer Tomography Images</i>	55
3.1.5	<i>Automatic Extraction of Mandibular Nerve and Bone from Cone-Beam CT Data</i>	56
3.1.6	<i>3D Segmentation of Maxilla in Cone-beam Computed Tomography Imaging Using Base Invariant Wavelet Active Shape Model on Customized Two-manifold Topology</i>	58
3.2	Other Related Methods	59
3.2.1	<i>Supervoxel-Based Segmentation of Mitochondria in EM Image Stacks with Learned Shape Features</i>	60

3.2.2	<i>Prostate MRI Segmentation Using Learned Semantic Knowledge and Graph Cuts</i>	62
3.2.3	<i>An Effective Interactive Medical Image Segmentation Method Using Fast GrowCut</i>	63
3.3	Chapter Notes	64
4	IMAGE PROCESSING	67
4.1	Bony Structure Segmentation	67
4.1.1	<i>Sheetness Score</i>	69
4.1.2	<i>Threshold-based Bony Structure Segmentation</i>	72
4.2	Super-Voxels	73
4.2.1	<i>Simple Linear Iterative Clustering</i>	73
4.2.2	<i>Seeded-up Turbo Pixels</i>	75
4.2.2.1	<i>Seeded-up Super Voxels</i>	75
4.3	Chapter Notes	77
5	GRAPH CLUSTERING	79
5.1	Graph Theory Fundamentals	81
5.2	Laplacian Partitioning	84
5.2.1	<i>Seed-based Graph Partitioning</i>	86
5.3	Community Structure Based on the Optimization of Modularity	87
5.3.1	<i>Modularity</i>	88
5.3.2	<i>Multilevel Optimization of Modularity</i>	89
5.3.3	<i>Seed based Multilevel Optimization of Modularity</i>	90
5.4	Chapter Notes	91
6	PROPOSED CBCT SEGMENTATION METHODS	93
6.1	Mandible and Skull Segmentation with Super-voxels and Laplacian-based Graph Clustering (MSSL)	93
6.1.1	<i>Stage 1: Automatic Extraction of Bony Structure</i>	94
6.1.2	<i>Stage 2: Interactive Segmentation.</i>	95
6.2	Mandible Segmentation with Super-voxels and Modularity Optimization (MSM)	96
6.2.1	<i>Stage 1: Super-voxel Generation</i>	96
6.2.2	<i>Stage 2: Interactive Segmentation:</i>	98
6.3	Chapter Notes	99
7	RESULTS	103
7.1	Method MSSL	104
7.1.1	<i>Parameter Optimization</i>	104
7.1.1.1	<i>Skull Segmentation</i>	105

7.1.1.2	<i>Mandible Segmentation</i>	107
7.1.2	<i>Comparison with a Related Segmentation Method</i>	109
7.1.3	<i>Experiment with Public CT Data</i>	111
7.1.4	<i>Usability Testing</i>	112
7.2	Method MSM	113
7.2.1	<i>Mandible Segmentation</i>	114
7.2.2	<i>Retina Cellular-level Segmentation</i>	115
7.3	Chapter Notes	117
8	CONCLUSIONS	119
8.1	Limitations	120
8.2	Future Work	121
	BIBLIOGRAPHY	123
	GLOSSARY	131
APPENDIX A	MISCELLANEOUS SEGMENTATION	133

INTRODUCTION

Cone Beam Computed Tomography (CBCT) plays a key role in dentistry. It is employed in the diagnosis and treatment planning of dental and maxillo-facial structures (MOZZO *et al.*, 1998). CBCT uses a cone-beam technique, instead of the traditional helical fan-beam technique, to produce images whose quality is adequate for diagnostic purposes. Moreover, its radiation is approximately 98% lower and the scanning time is shorter when compared with fan-beam Computed Tomography (CT) systems (MOZZO *et al.*, 1998; SCARFE; FARMAN; SUKOVIC, 2006). However, CBCT has some drawbacks such as artifacts caused by beam hardening, scattered radiation, low contrast resolution, inhomogeneity, and limited range of the X-ray area detectors. Moreover, values assigned to voxels are relative Hounsfield Units (HU) (RICHMOND, 2004) which cannot be used as a threshold value to identify bone areas, as usually done in traditional CT (MILES; DANFORTH, 2007). These drawbacks have an impact not only on image quality but also on the outcome of image processing tasks (LOUBELE *et al.*, 2006).

CBCT is routinely used to study skull bone changes that occur over time, following orthodontic treatments. For instance, the study of location and magnitude of facial asymmetry, congenital deformities, face deformities of mandibular joins, cranio-maxillo-facial deformities, and plan surgical correction (SPAMPINATO *et al.*, 2012; CHANG *et al.*, 2013; XIA; GATENO; TEICHGRAEBER, 2009). Due to the complex nature of human skull, such treatments require extensive planning (CHANG *et al.*, 2013). Hence, CT 3D reconstruction techniques and segmentation of skull bones, such as mandible or maxilla, are essential procedures for planning, performing, and following-up orthodontic treatments. (SPAMPINATO *et al.*, 2012)

Researchers from the School of dentistry at the Paulista State University “Júlio de Mesquita Filho” (UNESP), Araraquara – Brazil, study skull bone changes over time using CBCT. This study is usually performed with the aid of image processing tools and targets the segmentation of skull bones such as mandible, maxilla, and skull base. Once segmented, these bones are used in cephalometric analysis. Commonly, the segmentation of a CBCT data is manually performed and combines several computer tools specifically designed for medical

image processing (i.e., ITK-Snap (YUSHKEVICH *et al.*, 2006), 3D Slicer (FEDOROV *et al.*, 2012), Dolphin Imaging (SOLUTIONS, 2017)). Attaining accurate 3D segmentation requires extensive training as it not only demands knowledge about the complex anatomy of the human skull but also skills to properly operate these medical image processing tools.

Usually, such tools provide thresholding, level set, and region growing approaches which require intensive user interaction. Moreover, they are also very sensitive to artifacts, noise, low contrast and inhomogeneity (typical problems of CBCTs (WANG *et al.*, 2014)). Nevertheless, these tools are popular among orthodontists despite the painstaking process involved: need to manually guide the entire segmentation process and the excessive time spent to generate acceptable results (about 6 hours per bone).

Other approaches to CT bone segmentation can be found in the literature, ranging from thresholding, region growing, deformable models, level set, graph-cut, random forest classifiers, and statistical atlases (ZHENG; COMANICIU, 2014; WANG *et al.*, 2014; WANG *et al.*, 2016). Most of them are *ad-hoc* applications which handle one type of bone only: segmentation of either the mandible or the maxilla (CHANG *et al.*, 2013; SPAMPINATO *et al.*, 2012; DESCOTEAUX *et al.*, 2006; LOUBELE *et al.*, 2006; BRANDARIZ *et al.*, 2014).

Atlas-based methods, for example, aim to segment one category of object and can generate precise segmentation. However, they are very sensitive to anatomical variations, bone growth, for instance. Although such limitations may be circumvented by the use of multiple atlases, the computational efficiency is compromised, especially when non-rigid registration is needed (ZHENG; COMANICIU, 2014). Besides, building atlases requires multiple training volumes, which can be hard to obtain in clinical practice (WANG *et al.*, 2014; WANG *et al.*, 2016).

Graph-based methods are also widely used in medical image segmentation. In general, these methods build a graph, in which each pixel represents a vertex in the graph. The graph is then divided into clusters of vertices, which correspond to objects of interest in the input image (PENG; ZHANG; ZHANG, 2013). Finding the best partition of a graph is an NP-Hard problem, although optimal approximations do exist. Usually, such approximations divide a graph into clusters by minimizing an energy function (KOLMOGOROV; ZABIN, 2004). Even though graph-based methods generate accurate segmentation, their major limitation is related to the graph cardinality (number of vertex and number of edges): a single image with N pixels generates a graph whose adjacency matrix has N^2 elements, whereas commonplace 3D medical data can easily yield a graph with millions of vertex and an even larger number of edges. On the other hand, graph-cut approaches usually segment images automatically i.e. no user interaction is needed during the process. There are, however, some graph-based methods in which users can interactively guide the segmentation process by manually placing seeds (labels) over objects of interest (YI; MOON, 2012).

This thesis focuses on the use of graph clustering techniques for interactive CBCT segmentation. This includes the mandible and the skull altogether. It is important to note that the mandible is the only bone anatomically separated from the rest of the skull. Other skull bones are only “separated” via sutures, for instance, maxilla and skull base. This anatomical issue is visualized in Figure 1. Therefore, in this thesis, we refer to *skull* as the anatomical structure consisting of all bones of the head, except the mandible.

Figure 1 – Two samples of our CBCT dataset. A typical CBCT image is $400 \times 400 \times 554$ (male and female patients), with an isotropic voxel size of 0.4 mm, acquired with an i-Cat scanner. Upper row: sample 1; Lower row: sample 2. From left to right: sagittal, coronal and axial sections, respectively. Low contrast and sheet-like bones, which are difficult to segment, are highlighted with a yellow arrow. Notice that the mandible is anatomically separated from other bones, see yellow arrows in (a) and (b). In contrast, there is no natural division between the maxilla and the skull base, thus specialists separate them by an axial straight cut, see red lines in (b) and (e).

1.1 Aim and Research Hypotheses

The overall aim of this thesis is to develop an interactive approach to segment the mandible and the skull in CBCT data to be used in real clinical applications.

Specific Aims:

- To support anatomical variations and different patient types.
- To provide an interactive method in order to include user knowledge in the segmentation process.
- To segment 3D CBCT data in short processing time so that user interaction can be considered friendly.

As far as image processing is concerned, CBCT medical modality shows important specificities: a) images of high resolution; b) inherently existence of artifacts; c) low contrast among tissues of interest, and d) its ability to exhibit very complex anatomical structures. Adding to that list, the delicate anatomy of some skull bones (with thin shapes) and the anatomical variations according to age and gender. To overcome of these constraints and deliver an accurate method for real clinical practice, we adopted a two-stage strategy:

1. *Pre-segmentation*: image resolution reduction by grouping voxels into super-voxels. Besides, to segment bones such as the maxilla, sheet-like structures are enhanced using Hessian matrices.
2. *Interactive segmentation*: a graph is built from the super-voxels produced in stage 1. Next, user-placed seeds are used as input to an interactive graph clustering algorithm, whose output corresponds to the bones of interest (mandible or skull).

1.1.1 Research Hypotheses

It is hypothesized that a method based on super-voxels and graph clustering can produce accurate segmentation of the mandible and the skull in CBCT data, which can be used in real clinical applications. Besides, it provides flexibility to anatomical variations and processing time short enough to allow user interaction.

1.2 Contributions

The major contributions of this thesis are two new methods for interactive segmentation of CBCT data. The first is mainly focused on thin bones segmentation and the second on the interior region of the mandible. As mentioned before, both proposed methods reduce the image resolution by grouping similar voxels into super-voxels (we introduced a 3D version of a super-pixels (2D) algorithm). Besides, to segment thin bones, we developed a fast version of a sheet-like structures enhancement algorithm. We call our proposed methods: Mandible and Skull segmentation with Super-voxels and Laplacian-based graph clustering (MSSL) and Mandible segmentation with Super-voxels and Modularity optimization (MSM)

Method MSSL uses a Laplace-based graph clustering algorithm to segment the mandible and the skull. This algorithm guarantees a unique solution and the clustering process is guided with user-placed labels (seeds). In contrast, method MSM segments the mandible with a modularity optimization algorithm. We introduced a seed-guided version of this algorithm. Although it is not deterministic, it guarantees similar accuracy at each execution. Besides, this algorithm is scalable and its computational complexity is linear in sparse matrices ([BLONDEL *et al.*, 2008](#)). Figure 2 illustrates some results obtained during the doctoral research.

Figure 2 – Segmentation examples obtained during the doctoral period. From left to right: seeds, an axial sample of resulting segmentation and 3D rendering, respectively. First row: mandible segmentation example, and second row: skull segmentation, detail of the maxilla.

Other important contributions are:

1. Selection of bone of interest with brush strokes on any plane (sagittal, coronal and axial), which facilitates the interaction process.
2. Segmentation of CBCT for any patient (female/male, infants/adults).
3. Short processing time: only five minutes to segment a typical CBCT scan with $400 \times 400 \times 554$ resolution.

As a final additional contribution, we optimized method MSM to segment 2D images of cellular level cat retina, which present not only very irregular contours but also complex texture as shown in Figure 3.

Figure 3 – Retina cellular-level segmentation obtained with method MSM.

1.3 Publications

- (LINARES *et al.*, 2015) Oscar Cuadros Linares, Glenda Botelho, Francisco Rodrigues and João Batista Neto. An Adjustable Error Measure for Image Segmentation Evaluation. *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, IEEE.
- (LINARES *et al.*, 2017a) Oscar Cuadros Linares, Glenda Botelho, Francisco Rodrigues and João Batista Neto. Segmentation of large images based on super-pixels and community detection in graphs. *IET Image Processing journal*
- (LINARES *et al.*, 2017b) Oscar Cuadros Linares, Jonas Bianchi, Dirceu Ravelli, João Batista Neto and Bernd Hamann. Mandible and Skull Segmentation in Cone Beam Computed Tomography Using Super-voxels and Graph Clustering. **Submitted to** *The Visual Computer International Journal of Computer Graphics*. Springer.

- (LINARES *et al.*, 2017b) Oscar Cuadros Linares, João Batista Neto and Bernd Hamann. Biological Image Segmentation with super-pixels and modularity optimization in graphs. **To be submitted.**
- D. Coelho, O. Cuadros Linares, A. Oliveira, M. Andrade Jr., E. Rios, A. M. Zimer, L. H. Mascaro, J. D. E. S. Batista Neto, O. M. Bruno, and E. C. Pereira. A New and Cheap Methodology to Study Tridimensionally Pitting Corrosion in Stainless Steel. *The electrochemical Society (ECS), 233rd Meeting.*

Organization

This thesis is organized as follows:

- Chapter 2 presents an overview of CT and CBCT data, with details about their origin, acquisition process, pros and cons. A section with the basics of human head anatomy is also provided.
- Chapter 3 presents a literature review of the main medical image segmentation methods. We also present an overview of related work of mandible and skull segmentation in CT and/or CBCT data. In addition, we detail some correlated work based on methods similar to those used in this thesis.
- Chapter 4 details relevant concepts and methods related to image enhancement and resolution reduction.
- In Chapter 5 we provide the main concepts related to graph theory and complex networks. We introduce two graph clustering algorithms based on the Laplacian Matrix and Modularity optimization, respectively.
- In Chapter 6 we introduce two methods of skull and mandible segmentation in CBCT data. Both methods use super-voxels and graph clustering algorithms to segment CBCT data.
- Chapter 7 presents the main results of mandible and skull segmentation obtained with our CBCT dataset. In addition, results of segmentation of CT data of patients with cranio-maxillo-facial deformities are presented. A comparison with a related work is also shown in this chapter.
- Finally, chapter 8 provides conclusions, limitations, and future work of our proposal.

CONE BASED COMPUTED TOMOGRAPHY

Medical imaging plays a crucial role in many clinical procedures such as the detection and the diagnosis of different human diseases. Imaging modalities provide an effective way for the noninvasive mapping of the anatomy not only of the human body but also animals, plants, and even objects (PHAM; XU; PRINCE, 2000; SINGH; SINGH, 2003; HARDING, 2004). It is possible to reach the interior part of the human body without opening it (ELANGO VAN; JEYASEELAN, 2016). There exists a number of modalities to obtain a three-dimensional view of the human body e.g. Ultrasound, Magnetic Resonance Imaging (MRI), X-Ray (XR), Thermography and Computed Tomography (CT). In this chapter we present an overview of computed tomography, its history, principles, acquisition, advantages and drawbacks. We also provide a section about a relatively new CT imagining modality called Cone Beam Computed Tomography (CBCT). This is basically a CT that uses low radiation doses, designed to image the human head. Finally, to better understand the goals, methods, results and limitations related to this work, a brief overview of the human skull anatomy by CBCT is also presented.

2.1 Computed Tomography Fundamentals

In 1935 the term *tomography* (from the Greek *tomos*, meaning *section*) was introduced by Grossman, who improved a radiography based technique to image a specific layer or section of the human body that had been previously developed in 1920. Grossman's technique gives an image of a body parallel-oriented to the film. In 1937 Watson proposed another tomography method with transverse sections (cross sections). These methods (conventional tomography) had problems such as poor clarity and low detail to be used accurately in diagnostic (SEERAM, 2015).

In 1970 Sir Godfrey Hounsfield developed the *Computed Tomography* that uses a X-ray equipment to create detailed pictures, or scans, of areas inside a body. CT overcomes limitations of conventional tomography with a mathematical method to construct a 3D image from cross-

sectional slices (2D images). Sir Godfrey Hounsfield and Allan McLeod Cormack won the 1979 Nobel Prize for Physiology or Medicine for having developed the diagnostic technique of X-ray CT (ORANSKY, 2004).

Since its introduction, CT has revolutionized the diagnostic radiology and its use has increased rapidly. A study of 2009 estimated that there is 1 CT scanner per 100 thousand people in Brazil (GUTIERREZ, 2009), whereas in 1996 there were 2.6 CT scanners in the United States and 6.4 in Japan (ELANGO VAN; JEYASEELAN, 2016).

A CT scan is built in three steps: data acquisition, image reconstruction and image display:

1. The data acquisition is the process in which an X-ray collection is obtained from the patient using an electronic scanner (X-ray tube) that translates a detector in straight line across the patient's head and moves from left to right (revolutions). The X-ray source emits rays with a fan-beam geometry. This method of scanning a patient is referred to as *helical* or *spiral* CT. Figure 4 shows a design of a helical fan-beam scanner.
2. After the X-rays have been collected they are sent to a computer. An image reconstruction algorithm is applied to build the CT image. The Hounsfield's method is called *algebraic reconstruction technique* (RATLIFF, 2013; SEERAM, 2015).
3. In the beginning, the reconstructed images were displayed in a cathode ray tube. Nowadays they can be stored on any personal computer, displayed and handled with specialized software for medical image processing (i.e. ITK-Snap (YUSHKEVICH *et al.*, 2006), 3D Slicer (FEDOROV *et al.*, 2012), Dolphin Imaging (SOLUTIONS, 2017)). Figure 5 shows an example of a CT image.

Figure 4 – Helical fan-beam CT scanner design showed in two positions.

Figure 5 – Example of a CT scan of a human head (PIXMEO-SARL, 2016). Initially, an axial slice image series is reconstructed. Next, coronal and sagittal are calculated by interpolation of the axial stack. This process is called Multi-planar reformatting. CT sample of resolution $(512 \times 512 \times 460)$ anisotropic voxel dimensions $(0.4883 \times 0.4883 \times 0.7)mm$ (SEERAM, 2015).

2.1.1 Technical Notes on Computed Tomography

From a digital image point of view, a CT scan is a 3D image (or volume) with $n = x \times y \times z$ 3D pixels (or voxels). CT pixel values are not intensity numbers or color vector channels, as usually found in digital images. A CT pixel value is denominated Hounsfield Unit (HU) or CT number, which is computed with a linear transformation of the fraction of a beam of X-rays that is absorbed or scattered per unit thickness of the body. This fraction is called *Attenuation Coefficient* and is denoted by μ . The corresponding HU value is therefore given by:

$$HU = 1000 \times \frac{\mu - \mu_{water}}{\mu_{water} - \mu_{air}}, \quad (2.1)$$

where μ_{water} and μ_{air} are the attenuation coefficients of water and air, respectively. If the data are recorded with sufficient accuracy, different substances may be revealed with a certain degree of statistical confidence, although this is not common in practice (RATLIFF, 2013). Table 1 presents the HU of common substances (BUZUG, 2008; ŠPELIĆ; MONGUS; ŽALIK, 2011).

Table 1 – Hounsfield Unit (HU) of common substances.

Substance	HU
Air	-1000
Fat	-120 to -90
Fluids	-30 to +75
Soft Tissue	+100 to +300
Craniofacial Bone	+200
Cancellous bone	+700 to +3000
Cortical Bone	+3000

2.1.1.1 CT Image Quality

The image quality of CT is directly related to the attenuation, which is hard to setup. Basically, a low-energy X-ray radiation with larger wavelength is more strongly attenuated when passing through matter than a high-energy X-ray. Low energy is generally avoided in X-ray imaging, as it increases the radiation dose and does not contribute to the image quality as it is absorbed by the body (BUZUG, 2008). On other hand, with high energy, photons are not correctly attenuated and this may cause the so called *beam hardening* and *scatter radiation* which produces artifacts in the reconstructed CT image (BOAS; FLEISCHMANN, 2012). Figure 6 shows an example of beam hardening in a CT section.

Figure 6 – Example of beam hardening (yellow frame). Axial CT section of a human head, detail of lower teeth. Slice extracted from a CT sample with resolution $(512 \times 512 \times 130)$ and anisotropic voxel dimensions $(1 \times 1 \times 3)mm$ (SHARP PAOLO ZAFFINO; RAUDASCHL, 2016)

There are several other image quality problems related to CT, for example:

- *Poisson noise* due to statistical errors of low photon count which results in random thin bright and dark streaks (BOAS; FLEISCHMANN, 2012).
- *Metal artifacts*, scatter and beam hardening are hard in regions with metallic matter.

- *Out of field artifact*. CT sections can be acquired with a field of view smaller than the object being scanned, i.e. parts of the patient are positioned outside the scan field of view. When tissue is located outside the view it may cause attenuated or harden beams ([HSIEH, 2003](#)).
- *Motion artifact* due to patient movements such as breathing, bowel or cardiac. This may produce blurring and double effect in CT sections.

Modern CT scanners have been developed to produce more than 320 slices per revolution and they can achieve *isotropic* resolution, with a perfect cubic 3D pixel form (equal width, height and length). The opposite is *anisotropic* resolution, where 3D pixels are not perfect cubes. CT images with isotropic resolution have higher quality than anisotropic samples, as the former do not exhibit the stair-step effect as the latter. Besides, modern scanners can manage isotropic resolution lower than 0.4mm ([SEERAM, 2015](#)). Another advantage of isotropic resolution is that image processing algorithms, such as registration and geometric transformations, can be applied directly to the image without losing perspective. Isotropic and anisotropic voxels are shown in Figure 7.

Figure 7 – Anisotropic and isotropic resolution in CT. Isotropic voxels have equal dimensions (perfect cubes) whereas anisotropic voxels do not have equal dimensions.

2.1.2 CT Radiation Dose

Despite the important advances in radiographic diagnosis, CT has adverse effects related to the radiation dose, unfortunately. High radiation doses may produce high-quality images, whereas images acquired with lower doses are noisier and exhibit lower contrast ([KRILLE *et al.*, 2015](#)). Ionizing radiation by X-rays can knock electrons out of their orbits, thereby creating ions. When biological matter is exposed to X-rays, hydroxyl radicals can be created from the interaction of X-rays with water molecules. These radicals then interact with nearby DNA which may cause strand breaks or base damage. Or even worse, X-rays can ionize DNA directly. Most of the damage caused by radiation is repaired within the cell. However, DNA double-strand

breaks are occasionally misrepaired, leading to point mutations, chromosomal translocations, and gene fusion. These are related to induction of cancer (BRENNER; HALL, 2007).

Due to the increased incidence of cancer after CT scan exposure, a large number of studies about the correlation between X-ray absorption and risk of cancer have been carried out in several countries (MATHEWS *et al.*, 2013; KRILLE *et al.*, 2015; CONSTINE *et al.*, 1988). Therefore, it is highly recommended to limit CT scans to situations where there is a definite clinical indication and also to produce images with the lowest possible radiation dose (MATHEWS *et al.*, 2013).

2.2 Cone Based Computed Tomography Fundamentals

CT has supported the diagnosis, treatment and research in many medical specialties, for instance Gastroenterology, Neurology, Surgery, Pathology, and Dentistry (SEERAM, 2015). In dentistry it has been applied in surgical facial reconstruction, orthodontic surgery, dental implants and complicated extractions (SHINTAKU *et al.*, 2009). Nevertheless, for most dental applications, the use of this imaging modality has been impaired due to its high costs, difficulty to manipulate and, mainly, the high radiation dose absorbed by the patient (SCARFE; FARMAN; SUKOVIC, 2006). To overcome these limitations Mozzo *et al.* (1998) developed a new type of CT machine designed to image dental and maxillo-facial structures called Cone Based Computed Tomography (CBCT).

Unlike traditional helical 2D fan-beam CT scanners, CBCT uses a 3D cone-beam geometry, that involves a single 360° scan where the X-ray source and the acquisition system synchronously move around the patient's head who is sat and stabilized with a head holder (SCARFE; FARMAN; SUKOVIC, 2006). Figure 8 shows the design of a CBCT scanner.

Figure 8 – Representation of Cone beam computed tomography technique.

2.2.1 CBCT Pros and Cons

Compared with traditional fan-beam computed tomography, the advantages of CBCT are threefold: the low X-ray radiation doses (approximately 98% lower), shorter scanning time and lower cost (MOZZO *et al.*, 1998). In spite of the low X-ray radiation, CBCT scanners produce images whose quality is adequate for clinical practice in dentistry applications. In addition, all CBCT images have isotropic resolution with voxels size ranging from 0.125mm to 0.4mm. Furthermore, CBCT is a reasonable and reliable alternative considering radiation dosage, image quality, and comfort for the patient (SHINTAKU *et al.*, 2009).

Nevertheless, CBCT has several limitations such as (VOS; CASSELMAN; SWENNEN, 2009; SHINTAKU *et al.*, 2009):

- Voxel values in CBCT are relative Hounsfield units and cannot be used to estimate bone density, nor as a threshold discriminator.
- The low X-ray doses radiation generates low contrasts CBCT images. Theoretically, it is expected that bones have HU higher than soft tissue. In CBCT, however, low-density bones may have HU lower than nearby tissue. Besides, soft tissue and hard tissue areas may have similar HU values (similar density).
- All artifacts found in CT also appear in CBCT. Nevertheless, scatter radiation effect and beam hardening are much higher in CBCT than in conventional CT.

2.3 Human Skull Anatomy Basics

The human skull is the bony structure that forms the head in the human body. Skull bones anatomy is extremely complex. They exhibit both high and low density, thick and thin thickness, and intricate shape. Figures 9 and 10 depict the skull bones of a human being from a free-hand illustration and CBCT imaging modality, respectively.

(a) Anterior view

(b) Lateral view

Figure 9 – Free-hand illustration of human skull bones (NETTER, 2014)

(a) Red arrows indicate mandibular bones

(b) Red arrows indicate skull bones

Figure 10 – CBCT data: coronal, axial and sagittal sections of human skull bones, respectively. Bones indicated by red arrows are, from an image segmentation perspective, big challenges.

2.4 Chapter Notes

In this chapter we have introduced the main principles, benefits, and limitations of CT and CBCT. A brief section on human skull anatomy has also been provided. As mentioned here, CT is an imaging modality that has supported successfully medical diagnosis and treatment in many clinical fields. Nevertheless, according to medical researchers at the University of São Paulo - Riberão Preto, the use of CT has been reduced due to excessive X-ray radiation issues. Besides, it is argued, in many cases, whether CT is really necessary for diagnosis, or a simple 2D X-ray sample will suffice.

Alternatively, CBCT is a CT technique developed to image the maxillo-facial region that uses low radiation doses and does not expose the entire patient's body to the radiation. In

addition, CBCT scanners are cheaper than CT counterparts, and the scanning process is much faster. However, CBCT images have a poorer quality which may impose extra challenges in digital processing such as image segmentation, image quantization, or noise filtering. In addition, the complexity of the skull bones makes the digital processing still more complicated.

RELATED WORK

The goal of medical image segmentation is to cluster pixels into image regions or components, usually anatomical structures such as bones, organs or tissue. These components are used in many medical applications such as surgery and treatment planning, and diagnoses (HU *et al.*, 2009). There exists a number of medical image segmentation methods ranging from simple thresholding, region growing, watershed to the more sophisticated statistical atlases, deformable models, level set, and graph clustering (ZHENG; COMANICIU, 2014).

Thresholding is a simple fast region-based method in which a label is assigned to a pixel by comparing its gray-level value to one or multiple threshold constants. Threshold values can be global, if applied to the whole image, or dynamic, if different values are used for different regions in the image. There are several ways to define a good threshold value: for instance, build a histogram or use prior knowledge. In CT, for example, the Hounsfield scale can be used as prior knowledge to segment different substances such as bone, organs, air or water (See Table 1) (HU *et al.*, 2009). Nevertheless, in practice, simple thresholding does not yield accurate segmentation. However, it is usually applied as a pre-segmentation step, for example, Region of interest (ROI) extraction.

Deformable models are, maybe, the most popular methods in medical image segmentation (YUSHKEVICH *et al.*, 2006). For example, the active contours technique (commonly known as *snakes*) iteratively expands a contour to fit object boundaries. The fitting process minimizes an energy function that keeps the balance between external and internal forces. The original discrete energy functional equation proposed by Kass, Witkin and Terzopoulos (1988) is defined as:

$$E_{snake} = \sum_{i=1}^n E_{int}(i) + E_{ext}(i), \quad (3.1)$$

where n is the number of pixels in the image, $E_{int}(i)$ and $E_{ext}(i)$ are, respectively, the internal and external energies given by local derivatives.

Level set methods are similar to the deformable models, but with no explicit boundary representation. Instead, the boundary is embedded in an implicit function that produces negative values inside the object of interest, positive values outside, and zero for the object boundary. Level set methods are more robust than active contours to structural changes during the boundary fitting process (ZHENG; COMANICIU, 2014).

Graph cut is also a widely used technique in medical image segmentation, first introduced by Boykov and Jolly (2001). Since then, numerous related approaches have been developed not only for medical image segmentation, but also for video and natural scene image segmentation (YI; MOON, 2012). Basically, this method divides a graph, built from an image, so that sets of vertices correspond to objects in the image. The idea is to find the best division of the graph, which is an NP-hard problem. However, several good approximations are available. Boykov and Jolly (2001)'s method, for example, approximates the minimum cut of the graph by minimizing a cost function defined as:

$$E(A) = \sum_{i \in P} R_i(A_i) + \sum_{(i,j) \in N} B_{(i,j)} \cdot \delta(A_i, A_j), \quad (3.2)$$

where R_i is an regional term (histogram) that reflects how the intensity of the pixel i fits into the object and the background; $B_{(i,j)}$ is a coefficient to penalize discontinuities between i and j ; P is the set of vertices; (i, j) is the edge connecting the vertices i and j . The δ function returns 1, if $A_i = A_j$ or zero, otherwise. In Chapter 5 we give an extensive explanation of graph clustering methods.

To improve the segmentation of specific organs or regions in clinical applications, prior knowledge is often included. Since fully automatic approaches are not capable of accurately segmenting such structures, prior information can be requested by the process at running time. To take advantage of users' knowledge, many graph clustering methods require manual placement of labels and/or landmarks during the process.

Classifiers have also become popular in many fields of medical image segmentation. These methods use a set of predefined rules or representative images to train classifiers or build atlases. One very popular classifier is the so-called Random Forest Classifier (RFC) which constructs a set of decision trees (during training) and then assigns a class (classification) to individual trees (HO, 1995).

In the last years, a number of medical image segmentation methods based on statistical atlases have been proposed, mainly focused on brain segmentation in MRI data (CABEZAS *et al.*, 2011). An atlas is a labeled image used for guiding the segmentation process. There are two types of atlases: 1) topological atlas, which employs a single labeled image, and 2) statistical atlas, which builds an average image from a set of representative images. To build a statistical atlas, a reference image is selected from a set of images and then a registration process involving the reference image and all other images is carried out. Figure 11 illustrates the atlas building

process.

Figure 11 – Statistical atlas building process. An average image is computed from a set of representative images and a reference image. During the process are applied affine registration and deformable registration algorithms to align all images with the reference image (GUIMOND; MEUNIER; THIRION, 2000).

In the next sections, we present an overview of some selected methods which are related to the problem addressed in this thesis. Most are related to the segmentation of the mandible and the maxilla in CT or CBCT datasets, while others perform segmentation of mitochondria cells and prostate. These latter methods have been included as they address similar problems found in this thesis and/or employ strategies we intended to use. Thus, a detailed review of such methods will certainly provide a better understanding of the challenges we face and the goals of this thesis.

3.1 Mandible and Skull Segmentation

In this section we present: a) a method for a unified segmentation of mandible and skull in CBCT data; b) four methods of mandible segmentation in CT and CBCT data and c) an approach for the segmentation of the anterior wall of the maxilla in CBCT data. All methods are based on a combination of several techniques such as atlas, random forest classifiers, graph clustering and others. Most of them require user interaction to position landmarks and/or manually align the input image.

3.1.1 Automated Segmentation of Dental CBCT Image with Prior-guided Sequential Random Forests

This method was introduced by (WANG *et al.*, 2016) aimed to segment both the mandible and the skull in CBCT data. The authors formulate the segmentation as a classification problem. The method uses probability maps (maxilla, mandible and, background) produced with a Random forest multi-class classifier. The final segmentation is obtained by assigning the label with largest probability at each voxel location. The method is divided into four steps:

1. *Estimation of Initial Probability Maps with Majority Voting Strategy:* a dataset of CBCT scans, previously segmented by experts, is used as training atlases. The majority voting strategy is used to count the votes for each label in order to estimate the initial probability maps for voxels of both the mandible and the skull.
2. *Extraction of CBCT Appearance and Context Features:* Haar-like features are extracted for training classifiers. Besides, context features are extracted from the previous estimated probability maps. Both features are used to coordinate the segmentation. More specifically, for each voxel x in the input CBCT image or probability map, Haar-like features are computed as the local mean intensity of randomly displaced cubical regions R_1 and R_2 within an image patch R as:

$$f(x, I) = \frac{1}{|R_1|} \sum_{u \in R_1} I(u) - b \frac{1}{|R_2|} \sum_{v \in R_2} I(v), \quad (3.3)$$

where R is the patch centered at voxel x , I is the input CBCT image or probability map, and the parameter $b \in \{0, 1\}$ indicates whether one or two cubical regions are used.

3. *Training of Random Forest Classifiers:* in order to refine the segmentation, random forest classifiers are trained to learn the relationship between local appearance features and their corresponding manual segmentation labels.
4. *Repeat steps 2 and 3 until convergence:* the classifiers are trained sequentially. The probability maps are updated based on the classifiers trained in step 3. Then, in step 2, context features are extracted from the updated segmentation probability.

Results

The segmentation quality of this method was evaluated with 30 CBCT scans (12 males and 18 female) with nonsyndromic dentofacial deformity. The 30 CBCT images were manually labeled (segmented) by two experts to be used as ground truth data. The authors used the Dice coefficient D and the Hausdorff distance H to measure the segmentation quality. The average ratios reported are: mandible $D = 0.94$ and $H = 0.74$, and maxilla $D = 0.91$ (Hausdorff was not used for the maxilla). Figure 12 shows an example of the segmentation produced by this method.

Discussion

According to the quantitative analysis, this method produces very accurate segmentation of the mandible and the skull in CBCT data. Besides, the method separates upper and lower teeth in closed mouth cases (touching upper and lower teeth). Due to the irregular 3D geometry, separating lower and upper teeth is an extremely difficult process. Such task is usually neglected or addressed separately (LE *et al.*, 2009). On the other hand, as shown in Figure 12.b, low-density

Figure 12 – Input CT data and segmentation produced by the method *Automated Segmentation of Dental CBCT Image with Prior-guided Sequential Random Forests* (WANG *et al.*, 2016). Notice that the interior part of the mandible (low-density bone) are not included in the segmentation.

bone regions, such as the interior part of the mandible, are not included in the segmentation. Since CBCT data are not developed to study bone-quality, this is not considered as a limitation of the method. Nevertheless, like many atlas-based approaches, this one also presents drawbacks: a) high sensitiveness to anatomical variations; b) atlas quality dependence on the number of labeled samples, which are often hard to obtain.

3.1.2 Automatic Segmentation of Mandibles in Low-Dose CT-Data

Proposed by Lamecker *et al.* (2006), this method segments the mandible using a 3D statistical mandible model (statistical atlas) and a strategy based on deformation to detect the mandibular bone. The algorithm is divided into two phases:

1. *Training*: a statistical analysis on a set of 3D shapes (mandibles) is performed. This method was originally introduced by Lamecker, Lange and Seebass (2004). Each training shape is iteratively decomposed into a number of corresponding regions. Then, each region is mapped onto a common base domain under constraint of a minimizing metric distortion. All training shapes are represented in an n -dimensional vector, where n is the number of sample points used to discretize the shapes. The Principal Component Analysis algorithm is then applied to compute the variability within the training set, which results in a linear model:

$$S(b, T) = T(v + Pb), \quad (3.4)$$

where T is a linear transformation, b is a vector of eigenmodes of the covariance matrix (shape weights) and v are degrees of freedom.

2. *Segmentation*: a registration process is performed using the statistical model computed in the previous stage. This process is initialized by positioning the model on the input image

I , so that bounding box's centers are aligned. The registration is modeled as a minimization problem:

$$(b^*, T^*) = \arg \min_{b, T} \| R - S(b, T) \|^2 \quad (3.5)$$

where R denotes the surface of the shape in the input CT. The final segmentation is given by an approximation to R such $R^* = (b^*, T^*)$. The registration process is performed iteratively, and at each iteration several positions are adjusted until convergence. Points adjustments are controlled with a weighted least squares approximation.

Results

The authors tested their method using a statistical model built from a dataset composed of 13 individual mandible shapes extracted from conventional CT data. To perform a quantitative segmentation analysis, 15 CT images, segmented by experts, were used as ground truth (GT) data. A squared surface distance was used to compute the *mesh similarity* between segmentation and its respective GT data. The best segmentation quality reported is 2 cm around the ramus region. No more details about the results have been provided. Besides, segmented data have not been shown. Figure 13 shows an example of the input CT data used and a 3D view of the statistical atlas built.

Figure 13 – Example of CT data and statistical atlas used in the method *Automatic Segmentation of Mandibles in Low-Dose CT-Data* (LAMECKER *et al.*, 2006).

Discussion

Lamecker *et al.* (2006)'s method incorporates prior knowledge (atlas) to segment the mandible in CT data. Atlases were built using *regular* mandible training shapes without taking into account the teeth region. Even though no extensive details about segmentation quality have been reported, we can infer accuracy, as atlas-based methods are usually very accurate. However,

the method is prone to most common drawbacks of atlas-based approaches. For example, it is very sensitive to inaccuracies in the atlas. Besides, the registration process is also sensitive to the initialization scheme (in this case, a simple alignment of the bounding box centers). There are numerous other more sophisticated approaches to initialize a 3D registration (JOHNSON *et al.*, 2013). Finally, according to the authors, their method requires manual interaction as post-processing to correct some inaccuracies. They argue that increasing the number of samples to build the atlas may avoid manual corrections completely. Again, building good atlases depends of both quality and quantity of samples, which are hard to obtain in practice.

3.1.3 Automatic 3D Segmentation of Mandible for Assessment of Facial Asymmetry

This method, proposed by Spampinato *et al.* (2012), automatically segments the mandible in every 2D section of a CT scan, and then constructs a surface model from each segmented CT section. The algorithm is performed in two stages:

1. *Extraction of Inferior Mandible-arch*: computation of a mask used as a filter for the lower mandible-arch. The volume is first manually aligned (axial reference) and then each 2D CT section is evaluated from the bottom as follows:
 - a) The section is divided into n images using a bit-plane slicing coding, in which each image represents the contribution of a specific bit. For simplicity, it is assumed that the input slice is an 8-bit image, thus 8 images are obtained ($n = 8$).
 - b) Morphological operations (erosion and dilation) are applied to each bit-image.
 - c) For each bit-image, the largest connected component is identified.

To select the largest connected component among all the identified ones, a ratio is computed as:

$$R_i = \frac{A_{cc}}{A_{ch}}, \quad (3.6)$$

where A_{cc} is the area of the i^{th} connected component, and A_{ch} is the area of the convex hull that contains i . The slice in which R_i is lowest is then selected as the inferior mandible-arch mask.

2. *Extraction of Upper Mandible-arch*: since the upper mandible arch consists not only of multiple connected components (from the ramus to the condyle and coronoid process), but also other bones-parts, prior knowledge on mandible shape and position should be included in this step. It starts by applying the same process used to extract the inferior arch (steps 1.a and 1.b). Next, connected components are considered as part of the mandible only if they satisfy the following constraints:

- *Distance Constraint*: let L_R and L_L be, respectively, the top-right and the top-left points of the inferior arch previously found. And let U_R^i and U_L^i be, respectively, the bottom right and bottom left points of the i^{th} connected component of the slice under evaluation. Then, connected components, for which the distance between L and R points are within a predefined range, are possible candidates to be part of the mandible (See Figure 14.a).
- *Position Constraint*: selection of those candidates that rely on two regions defined by four tangents from the midpoint of the inferior mandibular arch to the most right and to the most left molar of the inferior arch (See Figure 14.b).
- *Shape Constraint*: the upper mandibular arch is identified by selecting components with a specific shape: thin and elongated in lower slices and rounded in the higher ones.

Figure 14 – Extraction of Upper Mandible-arch: distance and positions constraints to consider connected components as a part of the upper mandible arch (SPAMPINATO *et al.*, 2012).

Results

Ten CT datasets have been segmented. “Ground truth” data consists of measurements manually taken by specialists on the original CT dataset (Menton-Gonion distances). Final segmentation is given as a rendered surface of the entire mandible. Specialists are then asked to manually take the same measurements over the segmented data. Both set of measurements are then presented to the reader. Figure 15 shows an example of the CT data used in this method and a 3D view of a segmentation.

Discussion

Spampinato *et al.* (2012)’s method requires a manual orientation of the input CT image prior to segmentation. Next, a two-stage algorithm is performed to segment a regular mandible.

Figure 15 – CT data used in the method *Automatic 3D Segmentation of Mandible for Assessment of Facial Asymmetry* (SPAMPINATO *et al.*, 2012), and a 3D view a segmented mandible.

The authors use cephalometric analysis as quantitative analysis. More specifically, they manually compute the Menton-Gonion distance for the resulting segmentation and compare those with the same distance, over the input data. The measurements reported are apparently good. However, specific details about their results have not been provided. For example, they do not show how touching upper and lower teeth were separated (since teeth are included in the segmentation). Besides, the authors do not discuss the performance of their method on more challenging bone regions such as condyle or coronoid (the distance takes into account only the bottom part of the mandible). Moreover, the method is clearly sensitive to the initial manual orientation, and it is indicated only for “well shaped” mandibles i.e. no deformities, no broken or missing parts.

3.1.4 Automatic Segmentation of the Mandible in Cone-Beam Computer Tomography Images

Proposed by Brandariz *et al.* (2014), this method defines a three-step strategy to segment the mandible in CBCT data.

1. *Location of Bounding Box*: the bounding box of the mandible is located by computing the mean gray level of every axial section. It is assumed that the mean gray-level is higher in slices that contains bony region. The local minimum at both ends of the highest intensity values gives the mandible boundaries. After that, the bounding box is smoothed with a Gaussian kernel.
2. *Association to Axial Plane*: the volume is flattened into a 2D image (axial) by mapping the maximum corresponding value of the volume. Then, the mandible is located using a particle-based technique called *Particle Swarm Optimization* (EBERHART; KENNEDY, 1995). An elliptical particle evolves until convergence is reached. The criterion of convergence is given by the sum of intensities, evaluated at each iteration.

3. *Discrete Snake Evolution*: the semi-ellipse obtained in step 2 is used as input to a *snake* algorithm which evolves in the mandible according to internal and external energies that are computed from the first and second order derivatives of a spline defined over the semi-ellipse (KASS; WITKIN; TERZOPOULOS, 1988).

Results

This algorithm was assessed with a dataset of 30 CBCT scans randomly selected from a larger set of 350 scans. Results were compared with manual segmentation made by an expert. Quantitative analysis is given by computing the distance between the mandibular arch center, manually located, and its equivalent point automatically detected. The percentage of overlapping pixels was 95%. The authors did not show any segmentation results.

Discussion

This method segments the mandible in CBCT data. The mandible is automatically located using a mean intensity strategy. However, it is assumed that the CBCT sample is well aligned to the axis. The authors do not discuss the performance of their method on the most critical mandibular regions. Besides, they do not provide any qualitative results. In addition, according to the authors, their method does not segment the teeth region accurately.

3.1.5 Automatic Extraction of Mandibular Nerve and Bone from Cone-Beam CT Data

Proposed by Kainmueller *et al.* (2009), this method segments the mandibular nerve in CBCT data using prior knowledge about the mandibular nerve and a Dijkstra-based optimization procedure. The statistical model (atlas) is built by resampling 106 mandibular shapes with a common reference mesh and aligning them in a common reference frame. Next, Principal Component Analysis (PCA) is performed. The statistical model is represented as a bi-linear function:

$$S(b, T) = T(\mathbf{v} + \sum_k b_k \times p_k), \quad (3.7)$$

where b_k are the weights of the PCA eigenmodes p_k , T is a global affine transformation and $\mathbf{v}$ is the average shape vector whose dimension is three times the number of bone and nerve points.

Once the atlas of the mandibular bone and nerve has been constructed, the segmentation process is performed as follows:

1. *Initialization*: a Generalized Hough Transform is applied to detect 3D objects (mandible) in the input CBCT data.

2. *Image Driven Atlas Adaptation*: parameters b and transformation T are iteratively found so that the shape $S(b, T)$ approximates the unknown target shape R^* as accurately as possible. Then the parameters b and the transformation T are adapted toward R^* in the underlying image I .

The atlas-based segmentation described above, accurately segments the mandibular bone and “approximates” the mandibular nerve. In order to optimize the method, the authors introduced a Dijkstra-based optimization process. A directed graph is built and the path with minimal cost from source to target represents the “darkest tunnel” through the CBCT image. It is assumed that regions with a dark tunnel are surrounded by a brighter border. The cost $C(p_{k,i,i})$ of the edge k is computed from intensities inside a cylinder with center $p_{i,j}$, orientation t_k , length H and radius r . If a graph node position $p_{k,i,j}$ lies outside the mandibular bone as reconstructed by the statistical model, the cost is set to infinity.

Results

The accuracy of this method was tested with 106 CBCT scans. The average error for the mandible surface was computed by comparing the segmentation with its respective ground truth. The error ratios reported were: mean distance $[0.1 - 0.5]mm$, root mean $[0.2 - 0.8]mm$ and mean square $[2.3 - 6.2]mm$. For the optimized nerve reconstruction the average mean are $[0.6 - 1.0]mm$ right nerve and $[0.9 - 1.2]mm$ left nerve. Figure 16 shows a surface reconstruction of the resulting segmentation.

Figure 16 – Mandibular segmentation produced by the method *Automatic Extraction of Mandibular Nerve and Bone from Cone-Beam CT Data* (KAINMUELLER *et al.*, 2009). Color represents distance to ground truth data.

Discussion

This method segments the mandible by separating the mandible into two regions: the mandibular nerve canal and the “bone” (external surface of the mandible). Prior knowledge is included in the process through a statistical atlas built for the nerve canal and the bone. A Dijkstra-based algorithm is applied in order to improve the segmentation of the nerve canal. To compute the segmentation quality the authors compared their results (mesh comparison) with ground truth data. The metric was not mentioned, although visual comparison is presented (Figure 16).

In order to include the interior canal of the mandible (low-density), the bone was divided into nerve canal and “bone”, and a post-process is performed to accurately segment the nerve canal. The authors do not discuss the reason of including low-density regions in the segmentation.

As expected with atlas-based methods, results exhibit good accuracy rates. Nevertheless, this method also suffers from the same drawbacks of many atlas-based methods.

3.1.6 3D Segmentation of Maxilla in Cone-beam Computed Tomography Imaging Using Base Invariant Wavelet Active Shape Model on Customized Two-manifold Topology

Proposed by [Chang et al. \(2013\)](#), this method employs a wavelet density model to segment the maxilla in CBCT data with particular interest on its anterior wall. The method is divided into five steps:

1. *Training Shape Extraction and Patch Decomposition*: a training shape of the target is identified by manually placed control landmarks on ground truth data of the bone surface. Next, boundaries of the training shape are determined. The boundaries are obtained by calculating the shortest path between two control landmarks, which are composed of several other shortest paths on the ground truth. Shortest paths are found using the Dijkstra algorithm. The patch decomposition is performed by dividing the training shape into several patches. Each anatomical control landmark is used to define four patches (See Figure 17).
2. *Parameterization*: the planar parameterization domain and the mapping of each patch are computed. Since patches are composed of planar triangles, they can be “unfolded” onto a plane so that a planar polygon is created. Those new planar polygons are defined as the parameterization domain. The parameterization mapping is given by the corresponding barycentric point.
3. *Landmarks Digitalization Using Calmull-Clark Subdivision*: the landmarks of each training shape are digitized using the Calmull-Clark subdivision ([CATMULL; CLARK, 1978](#)).

4. *Statistical Model Construction*: two statistical models are built using the training images and their corresponding shapes with the digitized landmarks. The first model is built using a point distribution model (HORN, 1987) and the second using a wavelet density model (BERTRAM *et al.*, 2004) .
5. *Segmentation*: the anterior wall of the maxilla in a CBCT image is then segmented by using statistical priors of the statistical models. The final segmentation is obtained from the image registration process.

Results

Nineteen CBCT scans and their respective ground truth were used to validate this method. Only the anterior part of the maxilla was segmented and evaluated. The authors used the Hausdorff distance to compare the segmentation with the ground truth. The mean distance reported was $0.25mm$ and standard deviation $0.2mm$. Figure 17 shows an example of segmentation of the anterior wall of the maxilla.

Figure 17 – Segmentation of the anterior wall of the maxilla using the method *3D Segmentation of Maxilla in Cone-beam Computed Tomography Imaging Using Base Invariant Wavelet Active Shape Model on Customized Two-manifold Topology* (CHANG *et al.*, 2013). Training shape with landmarks (left), respective patch (middle) and segmentation (right).

Discussion

This method segments the anterior wall of the maxilla using prior knowledge given by statistical models optimized with a wavelet-based algorithm. The authors have used the Hausdorff distance to compute the segmentation quality. The average distance reported is $0.25mm$, which indicates high accuracy. One major concern is the method's high complexity which may hamper implementation.

3.2 Other Related Methods

In this section we present a) a method of mitochondria segmentation in MRI data, b) a method of prostate segmentation also in MRI data, and c) a general purpose medical image segmentation method in CT data. Although not designed for mandible or maxilla segmentation, they employ similar techniques used in this thesis. The first two methods pre-segment the input

image into super-voxels and use a graph clustering technique to produce the final segmentation. The latter uses a seed-based strategy to segment medical images.

3.2.1 Supervoxel-Based Segmentation of Mitochondria in EM Image Stacks with Learned Shape Features

[Lucchi et al. \(2012\)](#) proposed a method for the 3D segmentation of mitochondria in Electron Microscopy (EM). Their proposal is divided into two steps:

1. *Super-voxels Segmentation*: the input EM image is segmented into super-voxels using an adapted 3D version of the super-pixels algorithm known as *Simple Linear Iterative Clustering* (SLIC) ([ACHANTA et al., 2010](#)). All further processing is executed over super-voxels, rather than voxels alone. We provide a detailed explanation of super-voxels segmentation in Section 4.2.
2. *Feature Vector Extraction*: for each super-voxel i , a feature vector $\mathbf{f}_i$ containing information on local shape and texture is computed as:

$$\mathbf{f}_i = [\mathbf{f}_i^{Ray^T}, \mathbf{f}_i^{Hist^T}]^T, \quad (3.8)$$

where $\mathbf{f}_i^{Ray}$ is a ray descriptor and $\mathbf{f}_i^{Hist}$ is a super-voxel histogram. The ray descriptor captures local information around a point ([SMITH; CARLETON; LEPETIT, 2009](#)). [Lucchi et al. \(2012\)](#) extends the original 2D version of this operator to the 3D space. This descriptor is computed by triggering an imaginary ray at an arbitrary direction (θ_l, γ_l) from a given point c . The ray propagates until it reaches an edge point r . The Euclidean distance between points c and r is then computed. Image edges are detected with the Canny algorithm ([CANNY, 1986](#)). The histogram $\mathbf{f}_i^{Hist}$ of a super-voxel is created by concatenating the Local Binary Pattern (LBP) ([GUO; ZHANG; ZHANG, 2010](#)) histogram and the intensity histogram.

3. *Graph Clustering*: a weighted graph $G = (V, E)$, whose vertices V are super-voxels, is created. Vertices i and j will be connected with an edge $(i, j) \in E$ if they are adjacent. The edge weight is given by the Euclidean distance between $\mathbf{f}_i$ and $\mathbf{f}_j$. Once the graph is built, its vertices are grouped into sets (graph clustering) that correspond to objects in the input image. The best graph cut is approximate by minimizing a function energy:

$$E(y|x, \lambda) = \sum_i \phi(y_i|x_i) + \lambda \sum_{(i,j) \in E} \psi(y_i, y_j|x_i, x_j), \quad (3.9)$$

where E is the set of edges, $y_i \in [0, 1]$ is the label of the vertex i and x_i is the corresponding edge weight. The first term of Equation 3.9 is estimated by the Support Vector Machine

(SVM) classifier which is trained with manually placed labels of image background, edges and mitochondria, this term is computed as:

$$\phi(y_i|x_i) = \frac{1}{1 + P_\phi(y_i|x_i)}, \quad (3.10)$$

in which P_ϕ is the probability of i be within a mitochondria. The second term of Equation 3.9 is a pairwise function for intensity discrimination defined as:

$$\psi(y_i, y_j|x_i, x_j) = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), & \text{if } y_i \neq y_j \\ 0 & \text{otherwise} \end{cases}, \quad (3.11)$$

Results

The authors assessed the accuracy of their method by comparing the resulting segmentation with manual segmentation. The metric used was the Voice of the Costumer (VOC) ratio. According to this metric, the proposed method outperformed manual segmentation in 23% for the hippocampus, and 16% for the striatum. Figure 18 shows an example of hippocampus segmentation.

Figure 18 – Hippocampus segmentation obtained with the method *Supervoxel-Based Segmentation of Mitochondria in EM Image Stacks with Learned Shape Features* (LUCCHI *et al.*, 2012). Hippocampus demarcated in red.

Discussion

This method segments the mitochondria in EM images with super-voxels and graph clustering. Prior knowledge is included in the process through manually labeled samples used to train a SVM classifier. Quality is assessed by comparing their results with manual segmentation obtained with a segmentation tool called Ilastik (SOMMER *et al.*, 2011). The main goal of super-voxels is to reduce the processing time. However, it took 23 hours to segment the hippocampus (SVM and segmentation) and 7 hour to segment the striatum. The authors pondered that this was

an improvement rather than a drawback as it would have taken days to finish the segmentation, had super-voxels not been used.

3.2.2 Prostate MRI Segmentation Using Learned Semantic Knowledge and Graph Cuts

[Mahapatra and Buhmann \(2014\)](#) proposed an automatic method for prostate segmentation in MRI data. This method is based on super-voxels, random forest classifiers and graph clustering. It is performed in five steps:

1. *Super-voxels Segmentation*: the input MRI image is segmented into super-voxels.
2. *Region of Interest (ROI) Extraction* : a trained random forest classifier labels super-voxels as either “prostate” or “background”. The set of super-voxels labeled as prostate is defined as ROI.
3. *Construction of a Probability Map*: a second set of classifiers is used to build a probability map. Intensity, texture, curvature, and connectivity are used to train those classifiers. A probability map is then build for each super-voxels belonging to the ROI.
4. *Penalty and Smoothness Computation*: voxels belonging to super-voxels selected as ROI are also labeled as prostate or background. For each voxel in the ROI, a second order Markov Random Field (MRF) process is used to determine a probability of a voxel being prostate.
5. *Graph Clustering*: the final segmentation is obtained by minimizing an energy function defined as:

$$E(L) = \sum_{i \in P} D(L_i) + \lambda \sum_{(i,j) \in N} V(L_i, L_j), \quad (3.12)$$

where P is the set of voxels, N is the local neighborhood of each voxel i , L_i is the corresponding label of i , L is the set of all labels, and λ is a parameter to tune the relative contribution of the penalty. This energy function was originally proposed by [Boykov, Veksler and Zabih \(2001\)](#), in which $D(\cdot)$ is a penalty function given by a Markov Random Field process and $V(\cdot)$ is a smoothness function.

Results

This method was tested on 50 MRI scans with their respective ground truth data. The accuracy was computed with the Dice ratio. The mean ratio reported is $D = 0.81$. Figure 19 shows three segmentations obtained by this method with their respective ground truth.

Figure 19 – Segmentation of prostate generated with the method *Prostate MRI Segmentation Using Learned Semantic Knowledge And Graph Cuts* (MAHAPATRA; BUHMANN, 2014). Ground truth segmentation represented with green line whereas red line represents the proposed method.

Discussion

The method segments the prostate in MRI data with super-voxels, graph clustering and random forest classifiers. The authors argue that 3D prostate segmentation is an extremely time consuming process. However, with their super-voxel-based approach, they have been able to achieve an average processing time (training included) of 20-25 minutes. The average accuracy (Dice) was $D = 0.81$, which means acceptable, but not high accuracy. Nonetheless, the authors claim their method outperformed others.

3.2.3 An Effective Interactive Medical Image Segmentation Method Using Fast GrowCut

Zhu *et al.* (2014) proposed a user-oriented medical image segmentation method based on a reformulated GrowCut algorithm (VEZHNEVETS; KONOUCHINE, 2005), which is basically an automata evolution-based segmentation process. Given the slow convergence of the GrowCut algorithm when applied to 3D images, the authors introduced an adaptive Dijkstra algorithm to approximate the solution of a modified GrowCut function defined as:

$$s = \arg \min_{s \in S} \min_{q_i \in H(s,p)} \{ \|C_{q1} - C_s\|_2 + \|C_{q2} - C_{q1}\|_2 + \dots + \|C_p - C_{q_n}\|_2 \} \quad (3.13)$$

where C_p and C_q are features vectors at points p and q , respectively. S is the set of pixels, and $H(s, p)$ is any path connecting pixels s and p .

To solve Equation 3.13, Zhu *et al.* (2014) employ an adaptive Dijkstra algorithm which uses user-placed seeds as target labels and a Fibonacci-heap to select the smallest distance at each iteration. The process continues until the heap is empty.

Results

This method has been tested on lung, brain tumor, and heart blood pool data. The Dice coefficient and the Jaccard index have been used to measure the accuracy. The authors reported, in average, 97% of similarity (Dice) with Ground truth data and 97% overlap (Jaccard). Figure 20 shows an example of Heart blood pool segmentation.

Figure 20 – Heart blood pool segmentation produced with the method *An Effective Interactive Medical Image Segmentation Method Using Fast GrowCut* (ZHU *et al.*, 2014).

Discussion

Zhu *et al.* (2014) developed a general purpose method for 3D medical image segmentation, capable of extracting many anatomical structures such as organ or bones. It includes expert knowledge with labels (seeds) placed on regions of interest. Besides, the method is voxel-oriented which turns it very expensive in terms of memory usage and processing time. However, the segmentation process is performed inside a bounding box defined by the seeds, which reduces the size and, hence, improves performance. Nonetheless, it is still slow.

3.3 Chapter Notes

We presented in this chapter methods for medical image segmentation, mainly focused on mandible and maxilla of CT and CBCT data. Different strategies have been adopted to address this problem ranging from *ad-hoc* heuristics to sophisticated graph clustering or atlas-based algorithms. They all reported high rates of segmentation accuracy, especially those based on

atlas and graph clustering. It is worth noting that quantitative analysis is different for all methods (not only different datasets, but also distinct metrics), hampering comparison. This is somehow understandable, as it is hard to get access to medical datasets with reliable ground truth, mainly because of proprietary licenses and ethical issues.

In spite of these problems, it is evident that methods which include prior information produces more accurate segmentation. Another interesting aspect is that the most robust methods perform quantitative analysis with either Jaccard index, Dice coefficient or Hausdorff distance.

As for atlas-based segmentation methods, it is worth noting that despite the good segmentation accuracy, they are highly dependent on the atlas quality and are also very sensitive to anatomical variations in the input data. As a result, to sustain good segmentation, different atlases should be created whenever anatomical variations occur. For example, not only the mandible experiences anatomical changes over time, but also female and male mandibles present noticeable differences. Thus, at least one atlas should be built for each mandible type. Figure 21¹ shows an example of anatomical variation of the mandible with age.

Figure 21 – Anatomical changes in the mandible with age.

To create a good quality atlas it is necessary to have a representative dataset labeled by specialists (more than 10 samples). Moreover, the objects of interest must share similar anatomy. This labeling process is a manual task that consumes many hours of a specialist for a simple 3D sample, even when assisted by a medical segmentation tool. Needless to say that specialists not only should be familiar with the anatomy of the structures involved, but also with the segmentation tools. Besides, there are several legal constraints about medical data acquisition. As a result, despite the good quality of atlas-based methods, such constraints impair a wider use in real medical practices.

However, prior information can be included during processing, often as landmarks or brush strokes placed over the object of interest. This strategy takes advantage of the specialist's knowledge and can be used to segment a wide range of anatomical structures. Nonetheless, the specialist is responsible for indicating the object of interest correctly.

¹ Public domain images, created by Henry Vandyke Carter with no clinical purpose.

IMAGE PROCESSING

In this chapter we detail the image processing concepts that are part of the methods proposed in this thesis. To attain a precise segmentation, our methods combine algorithms for sheet-like bones enhancement in CT data, that allow the extraction of the entire bone structure. By doing that, we can separate bones from other *substances* normally present in a CBCT: tissue, air, and image background. We also present a super-voxels strategy to reduce the data redundancy and consequently reduces the computational cost of segmentation tasks.

On the other side, elements of images are usually referred to as *pixels* in 2D space. However, in 3D space there is no convention. These are called 3D pixels, voxels or simply pixels. In this thesis, we will address them as pixels, if no distinction is necessary i.e. when methods can be unrestrictedly applied to 2D and 3D images. When a distinction is needed, we will adopt the term *pixel* for 2D space and *voxel* for 3D space. The same observation applies to images in 2D and 3D spaces. Usually, an image in the 3D domain is called *volume* or 3D image. Likewise, we will use the term *image*, when no distinction is necessary, and *volume*, otherwise.

In this chapter, Ω will define a 2- or 3-dimensional image and Ω_i the value of the pixel $i \in n$, with n being the number of pixels in Ω . $\Omega_i(\mathbf{x})$ is the spacial location of the i^{th} pixel, and $\mathbf{x}$ is a vector of coordinates.

4.1 Bony Structure Segmentation

Bony structure segmentation in CT data refers to the process of separating bones from other substances e.g. soft tissue, air, or image background. Because most of the CT data are composed of non bone matter, it is highly desirable to extract the bony structure prior to any subsequent task in order to reduce memory usage and processing time. In general, this is a relatively simple process that can be successfully performed through threshold-based methods, by far the most popular approach in the field (ZHANG *et al.*, 2010). Nevertheless, sheet-like

bones present discontinuities and diffuse boundaries. This makes traditional threshold-based methods unsuitable for segmenting areas with such bones (DESCOTEAUX *et al.*, 2006).

In addition, common problems in CBCT, such as low contrast and relative HU, make the segmentation of thin bones still more complicated. According to the Hounsfield scale, it is expected that HU values of bony regions be greater than HU values of any other substances. However, as shown in Figure 22, several voxels of bony regions in CBCT data have HU values lower than HU values of adjacent tissue. Thus, the Hounsfield scale cannot be used precisely neither as a threshold discriminant nor as a bone density estimator (MILES; DANFORTH, 2007; VOS; CASSELMAN; SWENNEN, 2009).

Figure 22 – An example of low-density bone. Several voxels of the mandible internal region (yellow arrow) have HU values lower than HU values of tissue. The voxel value indicated by the yellow arrow is 154 whereas the value indicated by the red arrow is 240.

The existence of complex structures, such as vessels or bronchi (See Figure 23¹) and even low contrast, are not exclusive problems of CBCT. Those are also present in many other medical imaging modalities like CT, Magnetic Resonance Imaging (MRI), and X-ray. Numerous methods have been published aimed to segment those complex structures. In general, these approaches apply filters to enhance the intensity level of those structures so that traditional methods (for example, threshold) can be accurately segment objects of interest (DOEL; GAVAGHAN; GRAU, 2015). A common strategy to enhance intricate structures in medical images is to compute the Hessian matrix from the input image and use its eigenvalues to discriminate shape and orientation of such structures. In our work, we also adopt the same idea of enhancing complex structures, followed by a threshold-based strategy to extract the entire bony structure prior mandible or skull segmentation. In the next sections, we propose a bony structure segmentation method which consists of two stages:

1. Bone enhancement based on the eigenvalues of the Hessian matrix (sheetness score).

¹ Figure 23 is distributed under the Creative Commons Attribution-Share Alike 3.0 Licence. Author: James Heilman.

2. Threshold-based bone filtering through the sheetness score.

Figure 23 – Example of Lung's Bronchial arteries.

4.1.1 Sheetness Score

The Hessian matrix denoted by $\mathbf{H}$ is a square matrix $n \times n$ of second-order partial derivatives defined as

$$\mathbf{H}_{i,j} = \frac{\delta^2 f}{\delta x_i \delta x_j}, \quad (4.1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a multi-variable function $f(x_1, x_2, \dots, n)$. Local image derivatives are usually computed as convolutions with some predefined *operator* (e.g. Sobel, Canny, Prewitt, Gaussian kernel) which approximates the derivatives. However, derivative operators applied directly in discrete images could also enhance noise, especially those with higher order derivatives. Nevertheless, applying recursive Gaussian filters allows to approximate derivatives accurately, especially when derivatives are computed over the whole image. This method uses a parametrized Gaussian kernel:

$$g(\mathbf{x}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\|\mathbf{x}\|^2}{2\sigma^2}}, \quad (4.2)$$

in which σ is the standard deviation of the Gaussian function. The implementation of the Recursive Gaussian filter method is fast and requires setting the σ parameter only (MACIA, 2007).

The Hessian matrix contains geometric information about the surface f (e.g. concavity). Since eigenvalues and matrices are intrinsically related, we can use the eigenvalues of the Hessian matrix to obtain the geometric information about the surface f . Therefore, the eigenvalues of the Hessian matrix of an image can be used as indicative of the shape of objects in it. For that reason,

the Hessian matrix is widely applied in medical image processing to enhance vessels and, hence, identify vascular structures in 2D and 3D angiographic images (ANTIGA, 2007; JERMAN *et al.*, 2016).

Through eigenvalue decomposition one can characterize tube-like, sheet-like and blob-like structures. Basically, this process is performed using the three ratios of tube-like properties of eigenvalues defined by FRANGI *et al.* to separate vessels from other structures. However, the Frangi's ratios cannot be used to discern sheet-like structures, such as thin bones, which is one the fundamental goals of this project. On the other hand, DESCOTEAUX *et al.* defined three other ratios R_{sheet} , R_{blob} and R_{noise} to discriminate sheet-like structures from others. They are defined as:

$$\begin{aligned} R_{sheet} &= \left| \frac{\lambda_1}{\lambda_3} \right| \\ R_{blob} &= \frac{|(2|\lambda_3| - |\lambda_2| - |\lambda_1|)|}{|\lambda_3|}, \\ R_{noise} &= \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2} \end{aligned} \quad (4.3)$$

where λ_i are eigenvalues of the Hessian matrix. Without lost of generality we assume $|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$. Theoretical properties of the Descoteaux's ratios are showed in Table 2. Therefore, for each voxel Ω_i , the sheetness score S is computed from the eigenvalues of the local matrix $\mathbf{H}$ as:

$$S_i(\sigma) = \max_{\sigma \in \Sigma} \left\{ \exp \left[\frac{-R_{sheet}^2}{2\alpha^2} \right] \left(1 - \exp \left[\frac{-R_{tube}^2}{2\beta^2} \right] \right) \left(1 - \exp \left[\frac{-R_{noise}^2}{2\gamma^2} \right] \right) \right\}, \quad (4.4)$$

where Σ is a set of σ values in the range ($0.5 \leq \sigma \leq 3.0$). Parameters α , β and γ are set to 0.5, 0.5 and half of maximum Frobenius norm, respectively. These parameter values are suggested by the original authors of the method. To prevent division by null, if $\lambda_3 = 0$, the first and second terms of Equation 4.4 are set as:

$$\begin{aligned} \exp \left[\frac{-R_{sheet}^2}{2\alpha^2} \right] &= 0 \\ 1 - \exp \left[\frac{-R_{tube}^2}{2\beta^2} \right] &= 1 \end{aligned}.$$

To find structures of distinct levels of thickness e.g. vessels or bronchi, Equation 4.4 computes the Hessian matrix (Equations 4.1 and 4.2) at different scales of Σ and then returns the maximum value. In practice, this operation becomes computationally expensive as it is performed for each voxel in Ω , for each element in Σ . Since our goal is to enhance sheet-like structures

Table 2 – Theoretical properties of the Descoteaux’s ratios.

Ratios	Sheet	Tube	Blob	Noise
R_{sheet}	0	1	1	<i>undefined</i>
R_{blob}	2	1	0	<i>undefined</i>
R_{noise}	λ_3	$\sqrt{2\lambda_3}$	$\sqrt{3\lambda_3}$	0

only (no significant levels of thickness), we can drop the multi-scale process and compute the Sheetness Score for a single σ value. The redefined Equation 4.4 is given as follows:

$$S_i(\sigma) = \exp \left[\frac{-R_{sheet}^2}{2\alpha^2} \right] \left(1 - \exp \left[\frac{-R_{tube}^2}{2\beta^2} \right] \right) \left(1 - \exp \left[\frac{-R_{noise}^2}{2\gamma^2} \right] \right). \quad (4.5)$$

This equation requires the Hessian matrix to be computed only once, for each voxel. However, the parameter σ must be given. In Section 7.1.1 we give more details on parameter optimization. We suggest an interval of parameter values which guarantee optimal results.

Notice that the sheetness score assigns a value $S_i \in [0, 1]$ to each voxel Ω_i . This can be interpreted as the probability of such voxel belong to a bony region. Figure 24 shows the result of the sheetness score applied on a CBCT.

Figure 24 – Shee-like bones in CBTC data are hard to extract due to contrast problem. Results obtained with the Sheetness score are shown in (c), successfully enhancing bones in the data, especially thin ones.

The simplified Sheetness score successfully identifies and enhances sheet-like structures. However, it may enhance noise and artifacts too and its use alone will not produce correct results. In the following subsection we introduce a threshold-based strategy that allows the extraction of the entire bony structure, while eliminating most of the enhanced noise.

4.1.2 Threshold-based Bony Structure Segmentation

Extraction of Region of Interest (ROI) is a common task in image processing. This pre-segmentation process aims to reduce unnecessary information and, consequently, cut memory requirements and processing time. Ideally, ROI extraction algorithms should be fast and accurate. Since most of a CBCT image is composed of soft tissue, air and image background, it is desirable to get rid of useless information prior to segmentation, so that only the entire bony structure is extracted as a ROI.

We propose a threshold-based ROI extraction method using the sheetness score to separate the bony structure from other substances so that unnecessary information is reduced as much as possible. This bony structure segmentation method is driven by the following equation:

$$\Omega'_i = \begin{cases} background, & \text{if } \Omega_i \times S_i \leq T \\ \Omega_i \times S_i & otherwise \end{cases}, \quad (4.6)$$

in which Ω'_i is the new HU value computed for each voxel in Ω , T is a threshold parameter and Ω_i and S_i are the original HU value and the sheetness score respectively. The *background* value is equal to the minimal HU value occurring in the input CBCT (Ω). In this thesis we set $background = -1000$, bearing in mind that the value range of all our CBCT datasets is $[-1000, 5669]$.

Notice that Equation 4.5 employs a single parameter σ to compute the sheetness score for every voxel in Ω . This is faster than the original sheetness score proposal. However, the proposed sheetness score is likely to enhance not only thin bone structures, but also some noise. As a result, it cannot be uniquely used to extract the bony structure. However, our ROI extraction method is capable of extracting the whole bony structure accurately, filtering all non bone substances, including most of the wrongly “enhanced” noise in S . Figure 25 shows the bony structure extracted with our threshold-based method.

Figure 25 – Entire bone structure, including thin bones, extracted with the proposed threshold-based filter.

4.2 Super-Voxels

The bony structure extraction performed by the threshold-based filter presented in the previous section reduces the number of to-be-processed voxels considerably. Nevertheless, the number of voxels can and should be still further reduced to speed up the segmentation process. Super-voxels (called super-pixels in 2D image processing) is the outcome of a region-based segmentation method which groups a set of adjacent voxels that share a similar intensity (LINARES *et al.*, 2012). Indeed, image redundancy can be drastically reduced through super-voxels and thus the complexity of further processing tasks will also decrease (ACHANTA *et al.*, 2010). There exist several super-pixels approaches (ACHANTA *et al.*, 2012), but not all of them can be easily adapted to the 3D super-voxel scenario. Two of the most popular super-pixel methods are Simple Linear Iterative Clustering (SLIC) and Speeded-up Turbo Pixels (SUTP). Both methods are built on a similar idea of pixel swapping and both use an equivalent version of the *k-means* algorithm. More details about these methods are given below.

4.2.1 Simple Linear Iterative Clustering

Also known as SLIC, this method, proposed by Achanta *et al.* (2010), makes up super-pixels through grouping pixels using their color similarity and spatial position in the image. This process is performed in the CIELAB color space, in which human perceptual color differences are approximated by considering each color as a point in a three-dimensional space. The Euclidean distance can be used to measure the difference between them (JAIN, 1989).

The SLIC algorithm is defined as follows:

1. The input image is divided into k rectangular super-pixels, all with similar sizes and equidistant centers (Fig 26.a). The size of each super-pixel is approximately N/k , where N is the number of pixels in the image. The distance between two centers is $S = \sqrt{N/K}$.
2. The super-pixels centers are moved to the lowest gradient position in a 3×3 neighborhood.

Figure 26 – SLIC algorithm convergence process. Pixels are iteratively associated to their nearest super-pixel centers. After each iteration the centers are recomputed.

The image gradient is computed as follows:

$$G(x,y) = \|\Omega(x+1,y) - \Omega(x-1,y)\|^2 + \|\Omega(x,y+1) - \Omega(x,y-1)\|^2, \quad (4.7)$$

where $\Omega(x,y)$ is the *lab* vector at the pixel position (x,y) . Next, pixels within an $2S \times 2S$ area around the super-pixel center are associated to the nearest super-pixel center according to the following distance measure:

$$\begin{aligned} d_{lab} &= \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \\ d_{xy} &= \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \\ D_s &= d_{lab} + \frac{m}{S} d_{dx}, \end{aligned} \quad (4.8)$$

in which $[l, a, b]$ are the vector the three channels of the CIELAB color space. The parameter m is used to control the compactness of each super-pixel. The author of this method suggested empirically $m = 10$.

Once the pixel association process is done, a new center is computed considering all pixels belonging to their corresponding super-pixel. This process of associating pixels to the nearest super-pixel center and recalculating the centers are executed iteratively until convergence.

Once the associating process is finished, some pixels within a large segment may not be connected to it. In order to associate those pixels, a post-process of searching and grouping of missed pixels is executed.

This method can be easily adapted to create 3D super-pixels (or super-voxels), where the pixel association process is performed within a three-dimensional region of size $2S \times 2S \times 2S$. Adapting Equations 4.7 and 4.8 to the 3D space is trivial (LUCCHI *et al.*, 2012).

4.2.2 Seeded-up Turbo Pixels

Proposed by ÇIĞLA; ALATAN, this method builds super-pixels using the k-mean clustering algorithm. The process is divided into two steps:

1. The image is divided into squared regions of side length l .
2. Boundary pixels are evaluated by minimizing a cost function, and possibly moved to an adjacent super-pixel (pixel-swapping).

The pixel-swapping step is executed iteratively in a fixed number of iterations (it). Figure 27 shows an example of this algorithm after 5 iterations.

Figure 27 – SUTP algorithm process after 5 iterations. Boundary Pixels are iteratively moved to adjacent super-pixels.

In this thesis, we have generalized this method to produce both super-pixels and super-voxels. A more detailed explanation of this method in a 3D space is given next.

4.2.2.1 Seeded-up Super Voxels

We build on the idea of a 2D Speeded-up Turbo Pixels algorithms which is efficient and simple. It produces high-quality super-pixels and reduces resolution up to 90% (LINARES *et al.*, 2012). We have named the 3D version of SUTP as Speeded-up Super Voxels (SUSV). Like the

original version, SUSV is based on the k-means clustering algorithm and is also executed in two steps:

1. The image is divided into N regular super-voxels Ω_I ($I \in N$), or cubes, with edge length l . Initially, all super-voxels, which contain a subset of the original voxels, have the same volume and equidistant centroids. For simplicity, we assume that the minimal bounding box of the entire 3D image dataset is itself a cube.
2. Iteratively, voxels along a certain super-voxel boundary are evaluated and possibly shifted across the border between neighboring super-voxels.

The voxel-swapping process is based on the minimization of a cost function which takes into account both the voxel intensity and its spatial position. We propose a generic (2D and 3D) version of the cost function as:

$$C_{i,I} = p_1 \|\Omega_i - \Omega_I\|^2 + p_2 \|\Omega_i(\mathbf{x}) - \Omega_I(\mathbf{x})\|^2, \quad (4.9)$$

where Ω_i is the intensity value of the voxel under consideration, Ω_I is the mean intensity value of the I^{th} super-voxel, $\Omega_i(\mathbf{x})$ is the spatial location of the voxel under evaluation, and $\Omega_I(\mathbf{x})$ is the center of gravity of the I^{th} super-voxel. The parameters p_1 and p_2 are weights for intensity similarity and super-voxel border rigidity, respectively. More convex super-voxels are generated for high values of p_2 , although borders may not be accurately identified. Super-voxels are best adjusted to borders when high values of p_1 are considered. Determining the best values for p_1 and p_2 will depend on the application. In this thesis, super-voxels that are well adjusted to borders are preferable to convex ones.

Since Equation 4.9 is equivalent for both 2D and 3D applications, we have extensively analyzed several combination of p_1 , p_2 and the number of iterations it in a 2D scenario in order to find out the best values of these parameters. We found that for $p_1 = 1$, $p_2 = 0.5$, $it = 5$, and super-pixels with size l in the range $[10 - 15]$ super-pixels will be well adjusted to borders. We have published this study in the IET Image Processing Journal (LINARES *et al.*, 2017a). A thorough analysis of optimal values for parameter l is provided in Section 7.1.1. Figure 28 shows a CBCT section and the super-voxels after convergence. Notice how well adjusted super-voxels are to the bone contours.

Figure 28 – SUSV convergence after 5 iterations. Notice that in this example the input image (a) is the bony structure previously segmented with the method described in Section 4.1.

4.3 Chapter Notes

Applying algorithms for the enhancement of anatomical structures prior to other image operations such as segmentation and reconstruction is a common practice in medical image processing. As a result, the low intensity of complex anatomical structures such as bronchi or sheet-like bones is intensified. In this chapter we have presented a method for enhancement of sheet-like bones in CT data. This method, also known as Descoteaux filter, was originally proposed to segment the sinus bone, but it successfully identifies and enhances other sheet-like structures. However, a local Hessian matrix (derivatives are computed using a Gaussian mask with a σ parameter) and its eigenvalues must be computed several times for every voxel. The reasons for this repetitive computation of the hessian matrix are to capture different levels of thickness and avoid to enhance noise all together. By doing that, the processing time is clearly compromised. Empirically, we noticed that in our CBCT dataset, sheet-like structures were successfully identified for various scales of σ . However, noise was also enhanced for any σ

value. As a result, we simplified the original equation and applied it only once for a given single value of σ . Next, we applied a threshold-based filter in order to separate the image background from the entire bony structure. In this process, noise is also drastically reduced.

Both super-pixel methods presented in this section are based on similar versions of the k-means clustering algorithm (See Equation 4.8 and 4.9). Nevertheless, the SLIC algorithm uses the CIELAB color space, whereas SUTP employs the pixel gray-level only. As a result, SLIC may generate super-pixels which are best adjusted to boundaries in color images. However, SUTP can easily be adapted to work with any color model such as RGB, CMYK, or CIELAB, matching the performance of SLIC.

The main goal of this work is the accurate segmentation of CBCT data, whose voxels are not represented by color channels nor simple gray levels. Instead, CBCT data have HU values, which differ from the traditional pixel intensity scale $[0, 255]$. Our HU-oriented CBCT data exhibit values in the range $[-1000, +5669]$ (Notice that HU values can be negative). Therefore, we can use the STUP algorithm without tuning its original cost function (Equation 4.9).

In addition, SUTP not only produces super-pixels in fewer steps than SLIC, as it also goes without any post-process stage to assign missing pixels as SLIC does. Besides, because SUTP evaluates only boundary pixels, we can trigger the first stage (image division into rectangular super-pixels) with non-regular regions, that is, we can initially apply a quadtree strategy as a initial segmentation (LINARES *et al.*, 2017a). For these reasons, both SUTP and SLIC can be considered equivalent methods when applied to similar scenarios. However, SUTP is more flexible and easier to understand and implement than SLIC.

GRAPH CLUSTERING

A network is a set of items called *vertices* or *nodes* with connections among them. Those connections are called *edges* or *links*. Figure 29 illustrates a simple network. Several real world structures or systems such as social interactions, internet connections, web pages links, neural networks and digital images can be represented as networks.

Figure 29 – Example of a network with 7 vertex and 9 edges.

The representation of a system as a network is useful, for it allows several features to be analyzed, measured or discovered (FORTUNATO, 2010). Networks are mathematically modeled and studied through the Graph theory. However, there is no convention about the terms graph and network. In fact, for some authors, there is no difference between them and both terms may be used interchangeably (MALLIAROS; VAZIRGIANNIS, 2013). Alternatively, some authors use the term network to refer to the informal notion of an object composed of elements and connections among them, and the term graph as the formal mathematical concept of an abstract object formed by a set of vertices and a set of edges connecting pairs of vertices (BRANDES, 2005; FORTUNATO, 2010). Although in practice both terms can be used as synonyms, a distinction between them should come in handy. Thus, in this work we will use the term network to refer to the system (image) from which the network is built, and the term graph to emphasize the mathematical approach.

More recently, a theory aimed at studying a subset of networks of complex systems has emerged. They are known as complex networks. A system can be considered “complex” if it has a high number of elements that not only interact with each other but also with the surrounding environment in order to generate structures or dynamic processes. For example, in a social network where vertices are individuals and links represent the friendship among them, it is possible to find clusters of vertices with several connections among their components and a few links connecting vertices of different clusters. This may reveal the network structure. Usually, clusters of a complex network are called *communities* because the term comes from the study of social networks. Additionally, it is possible to study, for instance, the dynamic process of disease spreading in the aforementioned network. Examples of complex networks build from a popular social network and an digital image are showed in Figure 30.

(a) Social circles on Facebook: 4039 nodes and 88234 links (LESKOVEC; MCAULEY, 2012).

(b) Network from an image: 2500 nodes and 20000 links.

Figure 30 – Example of two complex networks.

Another important characteristic of complex systems is that the study of an isolated element does not provide a fully understanding of the structure nor the dynamism of the system (BARRAT; BARTHELEMY; VESPIGNANI, 2008). At this point, it is important to make a distinction between a *complex* system and a *complicated* one. For instance a network that models a computer motherboard architecture with all its interacting components may be complicated but not complex, since every part of the hardware has a known function which can be analyzed independently. Also, it is possible to predict what will happen to the system if any part is removed. On the other hand, a system built from an ant colony is complex, because we can not understand its dynamism if an isolated ant or even a subset of them is analyzed separately.

During the last years the graph theory has benefited from the numerous methods developed in the context of complex networks. Several measures and algorithms have been proposed to study the network dynamics, e.g. perturbation, percolation or cascading failure (COSTA *et al.*,

2007). Besides, numerous community detection algorithms, designed for dealing with networks in which the number of vertex exceeds a million, have been proposed (FORTUNATO, 2010). The Graph theory has also studied the problem of clustering in graphs years before the complex networks theory emerged. As a result, many algorithms, especially those based on the Laplacian spectrum of graphs, have been proposed (GRONE; MERRIS; SUNDER, 1990).

In practice, community detection and graph clustering algorithms share a similar goal: to find the best division of the graph in sets of vertices. Often, the criterion is to divide the graph so that the number of edges connecting clusters is minimum (graph cut). However, there may be additional restrictions, for example: given a number n of clusters, divide the graph into equally (as much as possible) sized clusters. This problem is called Graph Partitioning (GP). In this chapter we present a graph clustering algorithm and a community detection method. The former is based on the Laplacian spectrum of the graph and the latter is based on a multilevel optimization of modularity (NEWMAN, 2003). Both clustering methods use manually placed labels (seeds).

5.1 Graph Theory Fundamentals

Let $G = (V, E)$ be a graph formed by a set V of *vertices* (nodes) and a set E of *edges* (links) which connect pairs of vertices E_{ij} . The *cardinality* (number of vertices) of V is denoted by n and the cardinality of E is denoted by m . If two vertices are joined by an edge, they are *adjacent*. Graphs can be *directed* or *undirected*. In directed graphs edges have an *origin* and a *destination* i.e. $(V_i, V_j) \neq (V_j, V_i)$, whereas, in undirected graphs, edges joining two vertices are the same i.e. $(V_i, V_j) = (V_j, V_i)$. As a short notation, a vertex can be denoted by i and an edge can be denoted by (i, j) or by ij . All methods presented in this theses are based on undirected graphs. Hence, the following definitions are applicable to undirected graphs only.

The *Adjacency matrix* denoted by $\mathbf{A}$ of an undirected graph $G = (V, E)$ is a square matrix $n \times n$ with elements:

$$A_{ij} = \begin{cases} 1, & \text{if } (i, j) \in E, \\ 0, & \text{otherwise} \end{cases}, \quad (5.1)$$

matrix $\mathbf{A}$ is symmetric, its eigenvalues are real and its eigenvectors form an orthogonal basis (BIGGS, 1993).

The *degree* of a vertex i , denoted by k_i , is computed as:

$$k_i = \sum_{j \in V}^n A_{ij}, \quad (5.2)$$

The *average degree* of a graph, denoted by $\langle k \rangle$, is given by:

$$\langle k \rangle = \frac{1}{n} \sum_{i \in V}^n k_i, \quad (5.3)$$

The *degree matrix* of a graph, denoted by $\mathbf{D}$, is a diagonal matrix $n \times n$ with elements:

$$D_{ij} = \begin{cases} k_i, & \text{if } i = j, \\ 0, & \text{otherwise} \end{cases}. \quad (5.4)$$

The *Laplacian matrix* $\mathbf{L}$ of a graph $G = (V, E)$ is a real symmetric matrix with elements:

$$L_{ij} = \begin{cases} k_i & \text{if } i = j, \\ -1 & \text{if } i \neq j \text{ and } (ij) \in E, \\ 0 & \text{otherwise} \end{cases} \quad (5.5)$$

In matrix form, the Laplacian matrix can be written as:

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \quad (5.6)$$

Laplacian matrices have useful properties such as symmetry and positive-semi-definiteness (MERRIS, 1994). Besides the algebraic connectivity of a graph is associated to the second smallest eigenvalue of the Laplacian Matrix (spectrum). The remaining Sections 5.2 and 5.2.1 broadens the discussion about graph clustering based on the Laplacian spectrum.

Numerical values (weights) can be associated to the edges of a graph $G = (V, E)$. Edge weights can describe various properties such as cost, capacity, distance, interaction or similarity between vertices. Graphs with weights assigned to their edges are usually called *weighed graphs* and are denoted by $G = (V, E, W_E)$, where W_E is a set of weights with elements $W_{E(i,j)} \in W_E$ assigned to edge (i, j) .

The adjacency matrix of a weighted graph, denoted by $\mathbf{W}$, can be written as:

$$W_{ij} = \begin{cases} W_{E(i,j)}, & \text{if } (i, j) \in E, \\ 0, & \text{otherwise} \end{cases}, \quad (5.7)$$

Notice that an unweighted graph $G = (V, E)$ is a particular case of a weighted graph with *unit edge weights*, i.e, $W_E(i, j) = 1$ for all edges $(i, j) \in E$. The graph properties presented above can trivially be redefined for weighted graphs. The vertex degree of a weighted graph is computed as follows:

$$k_i = \sum_{j \in V}^n W_{ij}, \quad (5.8)$$

The edge cardinality $m \in E$ for a weighted graph is defined as:

$$m = \sum_{(i,j) \in E} W_{ij}, \quad (5.9)$$

and the Laplacian matrix can be written as:

$$\mathbf{L} = \mathbf{D} - \mathbf{W}. \quad (5.10)$$

Example

For the undirected and unweighted graph $G = (V, E)$ of Figure 29, with vertex cardinality $n = 7$, edge cardinality $m = 9$ and average degree $\langle k \rangle = 2.57$, we can define its adjacency matrix $\mathbf{A}$, the degree matrix $\mathbf{D}$ and the Laplacian matrix $\mathbf{L}$ are follows:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

$$\mathbf{D} = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix}$$

$$\mathbf{L} = \begin{pmatrix} 3 & -1 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & -1 & 3 \end{pmatrix}$$

5.2 Laplacian Partitioning

The problem of graph partitioning refers to the division of vertices V into g clusters so that the number of edges linking such groups (*cut size*) is minimum. Finding the optimal cut size is an NP-hard problem. However, there are numerous approaches to approximate the solution, although not necessarily optimal (FORTUNATO, 2010). Fiedler (1973) discovered that the solution for dividing a graph into two sets (bi-partition) is closely related to the second eigenvector of the graph Laplacian Matrix. Let us define R as the number of edges (cut size) connecting two groups (1 and 2) of vertices

$$R = \frac{1}{2} \sum_{i,j} A_{ij} \delta(g_i, g_j), \quad (5.11)$$

in which g_i is the group (1 or 2) to which vertex i belongs to. Function $\delta(x, y)$ returns 1, if $x = y$ and zero, otherwise. As each edge is summed twice, we multiply the summation by a $\frac{1}{2}$ factor.

Finding the optimal cut size is equivalent to minimizing R in Equation 5.11. As mentioned before, this is a very hard task. However, a good approximation can be given by the eigenvector corresponding to the second smallest eigenvalue of the Laplacian Matrix, known as the *Fiedler* vector (FORTUNATO, 2010). Let us write Equation 5.11 in a matrix form

$$R = \frac{1}{4} \mathbf{s}^T \mathbf{L} \mathbf{s}, \quad (5.12)$$

where $\mathbf{L}$ is the Laplacian matrix, and $\mathbf{s}$ is a vector with n elements

$$\mathbf{s}_i = \begin{cases} +1, & \text{if } g_i = 1, \\ -1, & \text{if } g_i = 2 \end{cases}, \quad (5.13)$$

Vector $\mathbf{s}$ can be expressed as a linear combination of the normalized eigenvectors $\mathbf{v}_i$ of the Laplacian as:

$$\mathbf{s} = \sum_{i=1}^n a_i \mathbf{v}_i, \quad (5.14)$$

in which $a_i = \mathbf{v}_i^T \mathbf{s}$ and the normalization $\mathbf{s}^T \mathbf{s} = n$. Then, the cut size can be written as

$$R = \sum_i a_i^2 \lambda_i, \quad (5.15)$$

where λ_i is the Laplacian eigenvalue corresponding to the eigenvector $\mathbf{v}_i$ (BRANDES, 2005). Without loss of generality, we assume that the eigenvalues are sorted in increasing order $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. The smallest eigenvalue λ_1 is always zero and its associated eigenvector $\mathbf{v}_1$ is a *ones vector* $(1, 1, 1, \dots)$ (MERRIS, 1994). We can then minimize R by choosing λ_1 and $\mathbf{v}_1$, and

match its signs so that $\mathbf{s} = (1, 1, 1, \dots)$. Hence, the final sum in Equation 5.15 is $R = 0$. Setting up $\mathbf{s} = (1, 1, 1, \dots)$ is equivalent to placing all vertices in the group 1 and none in the group 2. Technically, this is a valid partition of the graph, though useless (NEWMAN, 2006). Nevertheless, the second eigenvalue λ_2 is close enough to zero, and a good approximation to minimum can be achieved by setting up $\mathbf{s}$ as parallel as possible to $\mathbf{v}_2$ (Fiedler vector). To do that, we can set $\mathbf{s}$ as:

$$\mathbf{s}_i = \begin{cases} +1, & \text{if } \mathbf{v}_2^i > 0, \\ -1, & \text{otherwise} \end{cases}, \quad (5.16)$$

Figure 31 shows a graph partitioning solution using the Fiedler vector approach for a graph with nine vertices and fourteen edges.

Figure 31 – Graph Partitioning of a graph with nine vertex and fourteen edges using the Fiedler vector $\mathbf{v}_2 = (0.21, 0.41, 0.28, 0.50, -0.19, -0.37, -0.17, -0.32, -0.35)$ and corresponding $\mathbf{s} = (1, 1, 1, 1, -1, -1, -1, -1, -1)$. Thus $g_1 = (1, 2, 3, 4)$ and $g_2 = (5, 6, 7, 8, 9)$. The red dashed line shows the solution of the minimum cut on the graph.

The method presented above gives a basic overview of a graph partitioning with Laplace matrices. Nevertheless, there exists a number of graph partitioning methods which are also based on the Laplacian matrix combined with other approaches (BICHOT; SIARRY, 2013; DEO, 2017). Besides, most of these methods divide the graph automatically i.e. they do not use external information to guide the cutting process, whereas others use prior knowledge to guide the process. However, applying an automatic approach may be unsuitable for some applications, hence using external information becomes mandatory.

This is exactly the problem presented in this theses: segmentation of the mandibule and skull bones in Cone Beam Computed Tomography (CBCT) with no prior information is unsuitable. In the following section, we present a graph partitioning method based on the Laplacian matrix which uses prior information.

5.2.1 Seed-based Graph Partitioning

Like regular graph partitioning, seed-based methods divide a graph into two clusters (*foreground* and *background*). Unlike the former, the latter include user knowledge by manually labeling a subset of vertices (seeded vertices) either as foreground or background. This idea was first introduced in the image segmentation context by [Boykov and Jolly \(2001\)](#). Then, several adaptations and improvements followed ([BOYKOV; FUNKA-LEA, 2006](#); [JUAN; BOYKOV, 2006](#); [YI; MOON, 2012](#)). The basic idea consists of including prior information (seeds) as additional terms in the energy function and then solving it to minimize the cut size, taking into account the labeled vertices. [Casaca, Nonato and Taubin \(2014\)](#) proposed an energy function that is mathematically simple and guaranties a unique solution. Let us define a set of background B and foreground F seeded vertices with labels x_B and x_F ($x_B > x_F$), respectively. The energy function is defined as:

$$E(\mathbf{x}) = k_1 \sum_{i \in B} \|x_i - x_B\|_2^2 + k_2 \sum_{i \in F} \|x_i - x_F\|_2^2 + k_3 \sum_{i \in V} \left\| k_i x_i - \sum_{j \in m} w_{ij} x_j \right\|_2^2, \quad (5.17)$$

with m being the number of edges in the graph, $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is the vector solution, $w_{ij} \in W_E$ is the weight of the edge $(i, j) \in E$ and k_1, k_2 and k_3 are positive constants. The third term of Equation 5.17, the so called *Laplacian Coordinates term*, in a matrix form, can be written as follows:

$$\sum_{i \in V} \left\| k_i x_i - \sum_{j \in m} w_{ij} x_j \right\|_2^2 = \|\mathbf{L}\mathbf{x}\|_2^2, \quad (5.18)$$

where $\mathbf{L}$ is the Laplacian matrix. The energy function 5.17 can also be written in a more general matrix form:

$$E(\mathbf{x}) = \mathbf{x}^t(\mathbf{I} + \mathbf{L}^2)\mathbf{x} - 2\mathbf{x}^t\mathbf{b} + c, \quad (5.19)$$

where c is a constant, $\mathbf{I}$ is a diagonal matrix with elements:

$$I_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise} \end{cases}, \quad (5.20)$$

and $\mathbf{b}$ is a vector with n elements:

$$b_i = \begin{cases} x_B, & \text{if } i \in B, \\ x_F, & \text{if } i \in F, \\ 0, & \text{otherwise} \end{cases}, \quad (5.21)$$

The quadratic term $\mathbf{I} + \mathbf{L}_2$ of Equation 5.19 is symmetric and positive definite, and it is the unique minimizer of the equation. Thus, minimizing $E(x)$ is equivalent to solving the following linear system:

$$(\mathbf{I} + \mathbf{L}_2)\mathbf{x} = \mathbf{b} \quad (5.22)$$

Since the linear system 5.22 is symmetric, positive definite and sparse it can be solved by Cholesky decomposition. This is more efficient than the general Gaussian elimination, also known as Lower-upper (LU) decomposition (WATKINS, 2004). After solving Equation 5.22 vertices are labeled as follows:

$$l_i = \begin{cases} x_B, & \text{if } x_i \geq \frac{x_B + x_F}{2}, \\ x_F, & \text{otherwise} \end{cases} \quad (5.23)$$

where l_i are the corresponding foreground and background labels of vertex $i \in V$.

5.3 Community Structure Based on the Optimization of Modularity

The goal of graph partitioning is to divide a graph into a pre-defined number (usually two) of clusters, so that the number of edges connecting them is minimum. This minimization process is an NP-hard problem. Nevertheless, there are several methods that approximate the solution (See Section 5.2). However, this is not the only solution to the problem of graph clustering. One of the most studied problems in the complex networks field is that related to community detection (FORTUNATO, 2010). In networks that model real world systems, vertices normally share similar characteristics. These vertices tend to form clusters (*communities*) in which the number internal edges (edges linking vertices belonging to a cluster) is much higher than the number of edges connecting communities (external edges). This behavior, illustrated in Figure 32, is known as *Community structure* (OLIVEIRA; ZHAO, 2008).

From the community structure perspective, the problem of community detection consists in dividing a graph into sets of vertices so that the number of external edges is minimum. And like graph partitioning, optimizing this number is an NP-hard problem (FORTUNATO, 2010). Newman and Girvan (2004) introduced a clustering index called *Modularity* to measure the clustering quality. In the last years several community detection algorithms have been proposed in order to optimize the modularity. These are based on greedy agglomeration, spectral partitioning or multilevel optimization (CLAUSET; NEWMAN; MOORE, 2004; NEWMAN; GIRVAN, 2004; BLONDEL *et al.*, 2008; BRANDES *et al.*, 2008). In the next sections, we explain

Figure 32 – Network with three communities within dashed red circles. The goal of community detection algorithms is to group vertices so that the number of internal connections (black edges) outnumbers external connections (blue edges).

the modularity concept and describe a community detection algorithm based on a multilevel optimization of modularity.

5.3.1 Modularity

Modularity measures clustering quality based on the tradeoff between the number of internal and external edges (NEWMAN, 2006). In other words, the modularity can be defined as

$$Q = (\text{number of edges within communities}) - (\text{expected number of such edges}). \quad (5.24)$$

The first term of Equation 5.24 is straightforward to compute, whereas the second is somehow vague. To provide a robust modularity measure, the latter should be computed accurately. Newman and Girvan (2004) compute this amount based on the assumption that it is not expected to find communities in networks built with random connections, this kind of network is called *null model*. Therefore the second term of Equation 5.24 is given by the probability P_{ij} of an edge to exist between vertices i and j if connections are made randomly, but taking into account the actual vertex degree:

$$P_{ij} = \frac{k_i k_j}{2m}, \quad (5.25)$$

where k_i is the degree of the vertex i and $m \in E$ is the number of edges. Finally, the modularity can be mathematically defined as:

$$Q = \frac{1}{2m} \sum_{i,j} [A_{ij} - P_{ij}] \delta(g_i, g_j), \quad (5.26)$$

where $\mathbf{A}$ is the adjacency matrix, g_i is the community which vertex i belongs to, and function $\delta(x, y)$ returns 1, if $x = y$ and zero, otherwise. If the number of internal edges is similar to the *null model*, then Q will be zero. Nonzero indicates deviations from randomness. In practice, $Q \geq 0.3$ is a good indicator of strong community structure (CLAUSET; NEWMAN; MOORE, 2004).

5.3.2 Multilevel Optimization of Modularity

In real world networks there is usually no prior information about the modeled system. For instance the number of communities in the network and the number of vertices within communities are rarely known. For that reason, community detection algorithms should be capable of determining both the number of communities and the quantity of vertices within each community (FORTUNATO, 2010). These algorithms normally use modularity to measure the quality of network partitions.

As mentioned before, maximizing the modularity is an NP-hard problem (BRANDES *et al.*, 2008). The Multilevel Optimization of Modularity (MOM), introduced by Blondel *et al.* (2008), is faster and achieves higher modularity values than the so called *Fast Greedy* approach proposed by Clauset, Newman and Moore (2004), one of the fastest and most popular algorithm for approximating the maximization of modularity. MOM is an iterative approach divided into two steps:

1. *Modularity Optimization*: each vertex of the network is labeled as a community of one element only. Hence, there are initially as many communities as vertices in the graph. Next, for each vertex i and all its neighbors j , we evaluate the gain of modularity by assigning i to every vertex j 's community. The vertex i is placed in the community for which the gain of modularity is maximum, but only if this gain is positive. If no positive gain is obtained, i stays in its original community. This process is repeated for all vertices until no more improvements can be achieved.
2. *Community Aggregation*: build a new weighted graph whose vertices are the communities found in the first step. In this new graph, the edge weights are given by the sum of external edges. Internal edges lead a self-loop (edge from i to i) with weigh given by the sum of its respective edge weights.

Part of the efficiency of the multilevel optimization algorithm is own to a function ΔQ that computes the gain in modularity obtained by moving an isolated vertex i into a community g

$$\Delta Q = \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right] \quad (5.27)$$

where Σ_{in} is the sum of internal edges of the community g , Σ_{tot} is the sum of edges weights incident to vertices in g , k_i is the degree of vertex i (See Equation 5.8), $k_{i,in}$ is the sum of weights the vertex i to vertices in g , and m is the sum of the weight of all edges in the graph. Figure 33 shows an example of the process.

Figure 33 – Multilevel optimization of modularity algorithm. Each iteration is performed in two steps: 1) modularity optimization by moving vertices from their communities to adjacent ones, 2) creating a new graph where vertices within communities found in the first step are aggregated to form a single vertex. In this example two communities were found after two iterations.

5.3.3 Seed based Multilevel Optimization of Modularity

The MOM algorithm does not use prior knowledge to guide the clustering process. Nevertheless in some applications, for instance medical imaged segmentation, this can be a useful trend to enhance clustering precision. For that reason, in this theses, we propose a seed-based version of MOM algorithm. Prior information can be included in the MOM algorithm by means of labeling (seeding) a set of F foreground and B background vertices with labels x_F and x_B , respectively. As in the original MOM algorithm, the graph is divided into as many communities as there are vertices in the graph. During the community aggregation process, vertex labels are stored into a *membership* vector $\mathbf{x}$ of size n , which gives the current community of each vertex i and also the final vertex distribution. Initially, $\mathbf{x}$ is filled up with elements

$$x_i = i; \quad (5.28)$$

$i \in V$. Notice that labels change at each modularity optimization step.

To include the seeding vertices in the process, we define a seed vector $\mathbf{b}$ of size n as:

$$b_i = \begin{cases} 1, & \text{if } i \in x_F, \\ 2, & \text{if } i \in x_B, \\ 0, & \text{otherwise} \end{cases} \quad (5.29)$$

and we redefine the initially configuration of the membership vector $\mathbf{x}$ as:

$$x_i = \begin{cases} b_i, & \text{if } b_i > 0, \\ i + 2, & \text{otherwise} \end{cases}, \quad (5.30)$$

with this simple redefinition of the membership vector, vertices seeded as either foreground or background remain unchanged throughout the entire process. Recall that our goal is to divide the graph into two sets of vertices. However, the original MOM algorithm automatically establishes the number of communities and this is undesirable in our context. To overcome this, we just force the algorithm to aggregate vertices until only two communities remain. Once the process is finished, the membership vector contains only two labels (1 and 2) for every vertex.

5.4 Chapter Notes

In this chapter we presented an overview of graph theory and complex networks theory. In addition, two algorithms of graph clustering were presented: a) a graph partitioning method based on the Laplacian matrix and the Cholesky decomposition and b) a community detection algorithm based on a multilevel modularity optimization strategy.

The complex networks theory is intimately related to the graph theory, as networks are mathematically represented as graphs. Moreover, methods, concepts, and measurements are applied instinctively in both fields. Besides, community detection algorithms and graph partitioning have quite similar goal: to find or approximate the division of a graph as best as possible.

Graph partitioning methods usually minimize an energy function to optimize the cut size. This energy function is often solved with the eigenvectors of the Laplacian matrix or with a system of linear equations. These methods give a good approximation to the optimal cut size. However, for large graphs (more than a million of vertices) this strategy is not feasible as computing eigenvectors or solving linear equation systems in such situations is far too costly.

In this thesis we deal with 3D images of average size $400 \times 400 \times 400$ or 64 millions of voxels. A graph build from those images (if every voxel is mapped as a vertex) will contain the same number of vertices $n = 64000000$ and the adjacency matrix will be of dimension $n \times n$.

These are intractable amounts of memory and processing times for computing eigenvectors or solving systems of linear equations. In Section 4.2 we have presented two super-voxels methods whose main goal is to reduce the quantity of voxels in a 3D image so that graph partitioning methods can be considered.

On the other hand, community detection algorithms are designed to deal with large networks (more than a million of vertices). The multilevel modularity optimization algorithm was successfully tested in networks with 118 million vertices (it took 152 minutes) (BLONDEL *et al.*, 2008). Despite its success, minimizing the number of vertex is still a desirable task in order to reduce the processing time.

PROPOSED CBCT SEGMENTATION METHODS

As discussed in Chapter 3, segmenting the mandible and the skull in CBCT data is a complex problem that has already been addressed with different strategies. Naturally, those strategies have pros and cons, and selecting one of them will depend on the application. The application that motivated this thesis came from specialists of UNESP-Araraquara, whose aim is the segmentation of the mandible and the skull to evaluate bone changes over time. Their clinical practice involves many types of patients (male, female, child, and teenagers) with considerable anatomical variations. To the present date, the segmentation process is manual and takes in average six hours to get an acceptable result. For that reason, a method capable of segmenting the mandible and the skull in CBCT data irrespective to patients' gender, age, and anatomical variations is desirable. To the best of our knowledge, this is not currently available.

We propose two interactive methods for CBCT segmentation based on super-voxels and graph clustering. Since both fully automatic approaches and atlas-based methods are unsuitable for our application, we include prior knowledge in the process with labels (seeds) placed by the specialist. This strategy allows to segment the mandible and the skull in any type of patient, including those presenting deformities, missing or broken bones.

6.1 Mandible and Skull Segmentation with Super-voxels and Laplacian-based Graph Clustering (MSSL)

The diagram of Figure 34 illustrates the interactive two-stage MSSL method proposed in this thesis.

Figure 34 – Mandible and Skull Segmentation with Super-voxels and Laplacian-based Graph Clustering (MSSL): a) automatic segmentation of bony structures, followed by grouping of similar voxels into super-voxels and a graph representation where each vertex is a super-voxel; b) interactive segmentation with seed placement (brush strokes) on the bone of interest. Segmentation can be improved by either adding or removing seeds interactively.

6.1.1 Stage 1: Automatic Extraction of Bony Structure

The first stage of our approach consists in automatically building a graph from the bony structure detailed as follows:

1. *Bone Enhancement*: we first intensify the gray-level of the entire bony structure with the bone enhancement algorithm (Sheetness score) introduced in Section 4.1.1. This allows an accurate identification of all bones including thin ones such as the maxilla.
2. *Bony Structure Extraction*: we apply a threshold strategy, introduced in Section 4.1.1, to separate bones from tissue, air, and image background. In this method, the extracted bony structure is the ROI.

3. *Super-voxels Generation*: we then apply the SUSV algorithm, proposed in Section 4.2.2.1, to create super-voxels from the bony structure. This drastically reduces the quantity of redundant information (similar voxels).
4. *Graph Building*: a weighted graph is created in which vertices are super-voxels instead of individual voxels. We give more details about the graph-building process below.

Graph Building Strategy

Let us define a weighted graph $G = (V, E, W_E)$ where V is the set of vertices which correspond to super-voxels in the input image Ω , E is the set of edges defining pairs of connected super-voxels, and $w_{IJ} \in W_E$ is the weight assigned to an edge (I, J) . Ideally, edge weights should represent best-possible similarity values between two connected vertices. Commonly, weights are computed according to some image feature, e.g., color, intensity, gradient magnitude or contours (PENG; ZHANG; ZHANG, 2013). Nonetheless, voxels belonging to the same bone can have different intensity values. By contrast, regions of different bones may share similar textures. Thus, weights considering both features could lead to a graph with low edge-weights for voxels of the same bone. Therefore we compute edge weights based on a simplified Laplacian-based function and the spatial location of super-voxels' centroids as follows:

$$w_{IJ} = \exp \left(- \frac{\| \Omega_I(\mathbf{x}) - \Omega_J(\mathbf{x}) \|^2}{\delta} \right), \quad (6.1)$$

where $\Omega_I(\mathbf{x})$ is the spacial centroid of the I^{th} super-voxel, $J \in V_I$, with V_I being the set of adjacent super-voxels of I . Notice that each super-voxel may have ≈ 26 adjacent neighbors (See Figure 35). The Laplacian function leads to low weights for adjacent super-voxels whose centroids are spatially far apart. We compute the parameter δ as follows:

$$\delta = \max_{J \in V_I} \| \Omega_I(\mathbf{x}) - \Omega_J(\mathbf{x}) \|^2. \quad (6.2)$$

Notice that gray intensity is not considered in Equation 6.1, as all super-voxels under investigation belong to a bone. Thus, intensity would not contribute to a better data discrimination. It is worth noting that Equation 6.1 requires no fine-tuning parameters.

6.1.2 Stage 2: Interactive Segmentation.

The second stage is an interactive seed-based segmentation process and consists of 3 major actions:

1. *Seeding Process*: by applying brush strokes on any of the three image planes (axial, sagittal, and coronal), users can select bones of interest and undesired structures (red and green brushstrokes in image (1) of Figure 34.b, respectively).

Figure 35 – Visualization of adjacent super-voxels for MSSL method. We create an edge between vertices I (green) and J (blue) only if super-voxels Ω_I and Ω_J are adjacent (similar to Moore neighborhood). Notice that adjacent background super-voxels (yellow) do not belong to the ROI (bony structure), thus they are excluded from the graph-building process. As a result, any super-voxel Ω_I has ≈ 26 neighbors. However, most of them are eliminated, thus the degree of the vertex I is far lower than 26. In this example, the super-voxel under evaluation (green) is the same in (a), (b), (c).

2. *Graph Clustering*: the laplacian-based graph clustering, presented in Section 5.2.1, is then applied to divide the graph into two clusters (foreground and background). Seeds are used to guide the clustering process so that vertices belonging to the cluster associated with foreground seeds (red) correspond to the bone of interest.
3. *Interaction*: steps 1 and 2 can be executed repeatedly in order to correct possible imperfections.

6.2 Mandible Segmentation with Super-voxels and Modularity Optimization (MSM)

Illustrated in the diagram of Figure 36, this method differs from MSSL not only with regard to the underlying algorithm employed, but also from its ability to include low-density bone regions in the resulting segmentation. In Section 6.3 we give more details about this important feature and also point out other differences between both methods. MSM is divided into two stages as follows:

6.2.1 Stage 1: Super-voxel Generation

We generate super-voxels for the entire input CBCT image previously enhanced with the sheetness score.

1. *Bone Enhancement*: as in Method MSSL, the gray-level of bony regions is enhanced with the sheetness score algorithm presented in Section 4.1.1.

Figure 36 – Mandible segmentation with Super-voxels and Modularity optimization (MSM): a) preprocessing. Similar voxels are grouped into super-voxels; b) Interactive segmentation. User selects the mandible by placing seeds on it (brush strokes). Next, a bounding box is computed to delimit the segmentation area. We then build a graph where each vertex is a super-voxel inside the bounding box. Segmentation can be improved by either adding or removing seeds interactively.

2. *Image Multiplication*: low-density bones must be included in the final segmentation. However, the sheetness score does not enhance voxels belonging to such regions. As a result, we perform a voxel-multiplication of the input CBCT image Ω by the sheetness

Score S as:

$$\Omega' = \Omega \times S. \quad (6.3)$$

3. *Super-voxels*: we apply the SUSV algorithm to create super-voxels from the Ω' image.

6.2.2 Stage 2: Interactive Segmentation:

Unlike Method MSSSL, here we consider ROI all non-background voxels inside a bounding box in Ω' . Details of our ROI extraction process are given bellow:

1. *Seeding Process*: the specialist selects the mandible with brush strokes on any of the three image planes at her convenience.
2. *Bounding Box Filtering*: we compute the minimum bounding box enclosing the mandible using the seeds placed by the specialist. All super-voxels outside the bounding box are filtered out. A ROI consists of all super-voxels within the bounding box.
3. *Graph Building*: a weighted graph is created in which vertices are represented by super-voxels.
4. *Graph Clustering*: the graph is divided into foreground and background clusters using the multilevel optimization algorithm proposed in Section 5.3.3. Seeds are used to guide the modularity optimization process.
5. *Interaction*: steps 1 to 4 can be executed repeatedly. The graph may be rebuilt at each iteration, but only if the dimension of the bounding box changes. Since seeds can be added or removed, the bounding box may also change, thus the graph will have to be rebuilt from super-voxels within the bounding box. Building the graph takes about 5% of the total processing time.

Graph Building Strategy

Let's define a weighted graph $G = (V, E, W_E)$ in which V is the set of vertices defining super-voxels Ω_I , E is the set of edges defining pairs of connected super-voxels, and $w_{IJ} \in W_E$ is the weight assigned to an edge (I, J) . A vertex J is connected to the vertex I if J is inside a cubic region of edge length r whose center is aligned to the Ω_I 's spacial centroid. As shown in Figure 37, the set of super-voxels within this cubic region is defined as the neighborhood V_I .

The goal of this method is to segment the mandible so as to include both low and high-density bone. Therefore, we consider the gray intensity to compute super-voxels similarity

Figure 37 – Visualization of neighboring super-voxels in method MSM. An edge between vertices I (blue) and J (yellow) exists only if their respective super-voxels centroids are inside of a cubic region of length (in voxels) r . The set of super-voxels, whose centroids are inside the cube, is defined as V_I . Notice the number of neighboring super-voxels increases in a cubic form regarding to r . This neighborhood can be seen as an adaptive-size version of the Moore neighborhood.

(edge weight). Edge weights are computed using an adapted version of the Laplace distribution function defined as:

$$w_{IJ} = \frac{1}{2b_I} \exp \left(-\frac{p(|\Omega_I - \Omega_J| + \|\Omega_I(\mathbf{x}) - \Omega_J(\mathbf{x})\|^2)}{b_I} \right), \quad (6.4)$$

where Ω_I is the mean intensity of the I^{th} super-voxel, $\Omega_I(\mathbf{x})$ is the spatial centroid of the super-voxel I , p is a tuning parameter that we have defined as 2.0 to penalize small differences (intensity and position) among super-voxels. Parameter b_I is estimated as follows:

$$b_I = \frac{1}{n} \sum_{J \in V_I} |\Omega_J - \Omega_I|, \quad (6.5)$$

where V_I is a super-voxels neighborhood of I , and n is the size of V_I .

Due to the irregular shape of super-voxels, finding a super-voxels neighborhood is computationally expensive (the equidistant centroid distribution is lost after convergence). Therefore, verifying whether a point is inside a cube (constant time) is computationally faster than looking for adjacent neighbors. Given a super-voxel I , if a super-voxel J is inside a cube aligned to I , we assume that I and J belong to the same neighborhood. In contrast, in method MSSL, the number of super-voxels is much lower than in method MSM. Besides, in method MSSL, we connect only adjacent (degree 1) super-voxels, thus finding a neighborhood is faster.

6.3 Chapter Notes

Both methods we have proposed share a similar strategy to segment CBCT images: a) reduce the image resolution with super-voxels and build a graph from the set of super-voxels; b)

divide the graph into clusters that correspond to objects in the input image. Nonetheless, there exist various differences between them:

Method MSSL extracts the entire bony structure as ROI using the sheetness score and a threshold-based filter. By doing that, both image background and tissue are filtered out prior to segmentation. We highlight two advantages of this strategy:

1. About 99% of useless voxels (background and tissue) is eliminated. This allows us to use a computationally expensive, though accurate, method for graph clustering: minimization of a linear system using the Cholesky factorization (Section 5.2.1).
2. Seed placement becomes a trivial task. Sheet-like bones may have 1-pixel thickness, and placing seeds on such bones is, in fact, a delicate task. However, since we filter out adjacent tissue, there is no need for an accurate placement of strokes as long as undesired bones are not seeded (See Figure 38).

Figure 38 – Axial view of the anterior wall of the maxilla. In method MSSL an accurate placement of seeds is not needed (red strokes). This is useful because the maxilla is mainly composed of sheet-like bones. Notice that red brush strokes surpass thin-bones borders.

In addition, low-density bone regions (interior bone) are not included in the final segmentation. Actually, this is not a drawback in our particular application. As detailed in Section 2.2, CBCT is not a medical imaging modality designed for bone density evaluation. Therefore, orthodontists do not use it to study bone quality. Besides, this is an *ad-hoc* method, i.e. it is designed only for bone segmentation in CBCT data. However, we managed to successfully segment other bones in CT data.

On the other hand, Method MSM uses a bounding box-based filter to extract the ROI which includes high and low-density bones, and tissue (only image background is filtered out). Next, a graph, built from super-voxels, is divided into clusters using a modularity optimization approach. We highlight two advantages of this strategy:

1. Low-density bones are included in the segmentation. Although this is not an actual advantage for our application, this feature gives our method the ability to segment many

other “objects”, especially hollow ones whose interior regions are required as part of the segmentation. For example, segmentation of organs, retina, and even natural-scene images (See Section A).

2. Since tissue and low-density bones are included in the ROI, the number of super-voxels (about 27,000) to be dealt with is much higher than the number of super-voxels in Method MSSL (about 11,000). If a high-complexity graph clustering algorithm is to be employed, it may not be feasible to many practical applications. We have tested the MSSL’s Laplace-based algorithm with no success (hours). To overcome this problem, we proposed a new version of the multilevel modularity optimization algorithm, introduced in Section 5.3.3, which is capable of detecting communities in very large graphs. The average processing time takes about 5 minutes only.

Unlike method MSSL, seeds must be correctly placed in this version i.e. brush strokes must not drawn beyond object’s boundaries. Unfortunately, due to this constraint, it is hard to place seeds on sheet-like bones, which turns method MSM unsuitable for segmenting bones such as maxilla or sinus in 3D images.

Important Remarks on the Edge-weight Functions Equation 6.1 and 6.4

It is important to note that the Laplace-based graph clustering algorithm used in Method MSSL and the modularity optimization-based graph clustering algorithm used in Method MSM, in their own way, verify the sum of edge-weights in order to group vertices. (Equation 5.17 and 5.27). Ideally, “different”, though connected vertices, should have low edge-weights so that they do not contribute “negatively” to the sums. In contrast, similar vertices should have high edge-weights. We can generate edge-weights with this property using the probability density function of the Laplace distribution defined as:

$$f(x|\mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right), \quad (6.6)$$

in which very high values are obtained when the variable x is near location parameter μ , and very low if x get slightly farther. In other words, the Laplace curve decays exponentially to zero (See Figure 39).

Figure 39 – Laplace probability density function generated with Equation 6.6, we set $x = [-10, +10]$, $\mu = 0$ and $b = 1$.

The Laplace probability distribution fits our application, because it produces high values for similar super-voxels and very low values, otherwise. Besides, the parameter b gives information about the “diversity” of the distribution, and tunes a probability function for each super-voxel neighborhood. The parameter b can be estimated trivially (6.4). For that reason, edge-weight functions defined in Methods MSSL and MSM are based on the Laplace probability distribution.

RESULTS

We employed an empirical discrepancy approach to quantitatively evaluate the results of our methods: Mandible and Skull segmentation with Super-voxels and Laplacian-based graph clustering (MSSL), and Mandible segmentation with Super-voxels and Modularity optimization (MSM). We compared our results with the best-expected segmentation, known as Ground Truth (GT) (ZHANG, 1996). Our GT data were manually created by experienced orthodontists with the ITK-Snap tool, which includes both high- and low-density bones. Since method MSSL does not include low-density regions, for comparison purposes, we computed a mesh representation for the MSSL's output and its respective GT. To that end, we chose the marching cubes algorithm (LORENSEN; CLINE, 1987). We measured the similarity between two meshes using the Jaccard (J) index and Dice (D) coefficient. Both similarity metrics are widely used in medical applications for comparing segmentation outputs against ground truth data (LI; MAGNOTTA, 2011). Jaccard index and Dice coefficient are defined as follows:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}, \quad (7.1)$$

$$D(A, B) = \frac{2|A \cap B|}{|A| + |B|}, \quad (7.2)$$

where A and B are images, and the number of pixels is denoted by $|A|$ and $|B|$, respectively. Both Jaccard and Dice are within the interval $[0, 1]$. If A and B are empty, then $\{J|D\} = 1$.

Our CBCT data has a $400 \times 400 \times 554$ resolution with an isotropic voxel size of 0.4 mm of 17-year-old patients (male and female), acquired with an i-Cat scanner. We have performed experiments for both methods MSSL and MSM in different scenarios.

7.1 Method MSSL

Although automatic, the pre-segmentation stage of our proposed methods requires three parameters:

1. Gaussian standard deviation σ used by the sheetness score.
2. Threshold t for bony structure extraction.
3. Initial edge length l of each super-voxel.

In general, setting up these parameters is a time-consuming task imposed on users. In section 7.1.1, we present an analysis where parameters are evaluated in order to find an appropriate setup. We also compared our seed-based interactive segmentation method with a related segmentation approach described by [Zhu *et al.* \(2014\)](#). Results are provided in section 7.1.2. In addition, we tested our approach for publicly available CT data. Results are presented in Section 7.1.3. Finally, we performed a usability test with two orthodontists.

7.1.1 Parameter Optimization

We segmented the mandible and skull of eight CBCT samples, for which GT data were available (four mandible GTs and four skull GTs). Again, the pre-segmentation stage demands three parameters: σ , t and l . In Table 3 we present the range of parameters investigated.

Table 3 – Parameter ranges evaluated

Parameter	Range	Step Size
Sigma	$\sigma = [0.5 - 1.0]$	0.1
Threshold	$t = [100 - 300]$	50
Length	$l = [10 - 15]$	1

Taking all three parameters and their respective step sizes, 180 segmented meshes were produced for each bone. We compared all results with their corresponding GTs. The best segmentation accuracy, for both skull and mandible, was obtained with the parameter value settings present in Table 4.

Table 4 – Parameter values for most accurate segmentation

Parameter intervals
$0.7 \leq \sigma \leq 1.0$
$150 \leq t \leq 300$
$10 \leq l \leq 12$

Figure 40 shows an example of the seed placement process and the resulting 3D segmentation for two CBCT samples. In the following subsections, we provide more details about the segmentation of the skull and mandible, and corroborate the parameter analysis.

Figure 40 – Upper row: segmentation of skull of a male patient (including maxilla) with method MSSL. Lower row: segmentation of mandible of a female patient. From left to right: axial slices, seed placement process and 3D view of final segmentation. Red brush strokes indicate the bone of interest (foreground), whereas green strokes label undesired bones (background). Brush strokes do not have to be precise and may be drawn beyond bone borders, as long as they do not touch other bones. More example results are exhibit in Section A.

7.1.1.1 Skull Segmentation

The plots shown in Figure 41 represent segmentation accuracy (Jaccard curves) for different values of the three parameters. Best segmentation was observed for threshold (t) values within the range $150 \leq t \leq 300$. However, in the presence of thin bones, which are the most challenging structures to segment, the sheetness score (σ) plays a crucial role. Figures 42 and 43 show how different values of σ and t affect the segmentation of sheet-like bones. Regarding initial super-voxel edge length l , best results were obtained with $10 \leq l \leq 12$. For this range of l values, super-voxels are well-adjusted to borders after only 5 iterations. Larger super-voxel edge lengths demand more iterations until convergence. The mean and standard deviation values for all datasets are listed in Table 5.

Table 5 – Mean and standard (std) deviations of Jaccard index and Dice coefficient for skull segmentation of all datasets.

	Skull 1	Skull 2	Skull 3	Skull 4
Jaccard mean	0.828	0.874	0.865	0.827
Jaccard std	0.057	0.049	0.047	0.051
Dice mean	0.905	0.932	0.921	0.904
Dice std	0.034	0.028	0.025	0.030

Figure 41 – Influence of parameters t , σ and l on segmentation quality. The curves represent the mean Jaccard index of the four samples (720 segmentation results). We measured the segmentation quality by evaluating several combinations of the parameters mentioned, for instance: $t = 100$, $\sigma = 0.5$ and $t = [100 - 300]$. The most influential parameter is the threshold t (a) with best quality obtained for $t > 150$. In a lesser scale, variations in σ and l also affect the segmentation quality, seen as minor fluctuations in the curves.

Figure 42 – Axial view of skull segmentation of the maxilla for different values of σ , with $t = 200$ and $l = 10$. Sheet-like bones, such as the anterior and posterior walls of the sinus (yellow arrows in (a)), are accurately segmented irrespectively to changes in σ values. J is the Jaccard index.

Figure 43 – Axial view for skull segmentation of the maxilla for different values of t , with $\sigma = 0.8$ and $l = 10$. Changes in threshold t affect segmentation quality, unlike variations in σ , see Fig. 42. The retroantral fat pad (yellow arrow in (e)) is not entirely segmented for $t \geq 250$.

7.1.1.2 Mandible Segmentation

Unlike the skull, the mandible does not comprise several thin bones, albeit it has some structures, such as coronoid process and condyle, with prevailing low contrast causing diffuse borders in images. Segmentation of such bone-parts is difficult. Nonetheless, the sheetness score enhances sheet-like structures as well as low-contrast borders, leading to correct segmentation of such bone regions. Plots (Jaccard curves) of segmentation accuracy for different values of t , σ and l are shown in Figure 44. Figures 45 and 46 illustrate the segmentation of coronoid process and condyle for different parameter configurations. The mean and standard deviation values for all datasets are listed in Table 6.

Table 6 – Mean and standard (std) deviations of Jaccard index and Dice coefficient of mandible segmentation.

	Mandible 1	Mandible 2	Mandible 3	Mandible 4
Jaccard mean	0.853	0.851	0.882	0.854
Jaccard std	0.108	0.093	0.051	0.104
Dice mean	0.916	0.929	0.936	0.917
Dice std	0.076	0.059	0.029	0.073

Figure 44 – Influence of parameters t , σ , and l on mandible segmentation quality. We produced 180 segmentation of the mandible for four CBCTs (hence, a total of 720 segmentations). Curves show the mean Jaccard index for the four datasets evaluated. We measured segmentation quality by evaluating several combinations of the parameters, for instance: $t = 100$, $\sigma = 0.5$ and $t = [100 - 300]$, and so on. Like skull segmentation (Fig. 41), the most influential parameter is the threshold t (a), with best results obtained for $150 < t < 300$. However, σ and l also affect, marginally, the segmentation quality, seen as fluctuations in the curves. Local extrema, is noticeable for $t < 150$ (a); these are explained by the presence of noise in segmentation (Fig. 46.a). Noise can be interactively eliminated with extra brush strokes and, hence, improve segmentation quality.

Figure 45 – Axial view of the mandible segmentation (condyle and coronoid process), with $t = 200$ and $l = 10$. Results obtained with parameter σ at different scales and best results for $\sigma \geq 0.7$. Some noise is still present (yellow arrows in (e)), but can be interactively eliminated with additional brush strokes.

Figure 46 – Axial view of mandible segmentation (condyle and coronoid process), using $\sigma = 0.8$ and $l = 10$. Results shown for different values of threshold t . Best quality is achieved for $100 < t < 300$. Higher values of t do not yield proper segmentations of thin-bone regions, e.g., the coronoid process (yellow arrow in (e)). Noise can still be seen in the final segmentation (yellow arrows in (a)) for $t < 100$, but can be interactively eliminated with brush strokes.

7.1.2 Comparison with a Related Segmentation Method

We compared our approach with a related segmentation method described by Zhu et al., based on an “adaptive Dijkstra algorithm” (Section 3.2.3). It is also interactive and uses manually placed seeds for bone of interest selection. However, the method is computationally expensive and segmentation must be carried out locally: an axis-aligned bounding box, encompassing all brush strokes, is created and segmentation is performed in this region only.

We employed the same seeding setup conditions and same GT data. We first segmented the mandible of our CBCT 1 for two different seed placements, as shown in Figure 47.a and 47.d. Our results are shown in Figures 47.b and 47.e, whereas Figures 47.c and 47.f show the results produced by Zhu et al.’s method. In both cases, our method produced a more accurate segmentation of the mandible. In some situations, our method does not even require a user to specify background seeds (green strokes). Moreover, the similarity between both segmented bones produced by our method using different seeding setups confirms that it does not require a user to precisely place seeds to obtain accurate segmentation. The Jaccard indices for our method and Zhu et al.’s were 0.92 and 0.75, respectively. Segmentation quality of Zhu et al.’s method was improved by adding extra seeds, yielding a Jaccard similarity index of 0.88. Nevertheless, visual inspection revealed that the bone was still not properly segmented. Table 7 lists the Jaccard

index obtained for our CBCTs in a mandible segmentation scenario.

Table 7 – Comparing our method with Zhu et al.’s in a mandible segmentation scenario. Values are the Jaccard similarity with respect to ground truth.

	Mandible 1	Mandible 2	Mandible 3	Mandible 4
Our method	0.92	0.93	0.94	0.93
Zhu et al.	0.88	0.87	0.88	0.89

Figure 47 – Axial view of mandible segmentation for Zhu et al.’s and our method. Two seeding setups, (a) and (d), with red strokes indicating the desired bone and green strokes the undesired regions; (b) and (e) our results, for seeding conditions as defined in (a) and (d), respectively. Likewise, (c) and (f) are Zhu et al.’s results for the same seeding setups. Our method uses less brush strokes, and Zhu et al.’s results exhibit “leaks” (yellow frames) in low-density areas, even with seed placement 2.

We also evaluated both methods in a thin-bone scenario, with the segmentation of the skull shown in Figure 48. In general, segmentation of thin structures is a challenge for seed-based approaches. It is difficult to constrain the strokes to foreground regions only. Voxels belonging to adjacent tissues or background areas can be included by mistake. Thin bones have low-intensity values, and wrong segmentation is likely to occur, even when seeds are correctly placed. These are the main reasons why Zhu et al.’s method was not able to correctly segment the skull. Our approach, however, is not impaired in the same way. The pre-segmentation stage guarantees robust algorithmic behavior even when seeds are not perfectly placed and bones have low-intensity values, see Figure 48.a.

Figure 48 – Sagittal view of skull segmentation for Zhu et al.’s and our approach. Seeding setup is shown in (a), with red indicating the desired bone and green undesired regions. Our method properly segments thin bones, even when brush strokes cover adjacent tissue types, see yellow frames in (b). Zhu et al.’s method produces “leaks” in soft-tissue regions and air (cranial cavities).

7.1.3 Experiment with Public CT Data

We also evaluate our method on publicly available CT data ([PIXMEO-SARL, 2016](#)). No GT data were available for a quantitative analysis. Nevertheless, experienced specialists empirically confirmed the high quality of our results. Figure 49 shows a skull segmentation of a patient after surgical repair of facial deformity. The extraction of the mandible of a patient with missing teeth is shown in Figure 50. These results demonstrate the potential of our method to segment CT and CBCT for a wide range of patients, including patients with facial deformities, broken or missing bones.

Figure 49 – Skull segmentation. CT of patient with facial deformities after surgical repair. CT sample of $512 \times 512 \times 361$ resolution and anisotropic voxel spacing of 0.42mm, 0.42mm and 0.70mm.

Figure 50 – Segmentation of mandible. Patient with missing teeth. CT sample of resolution $512 \times 512 \times 460$ and anisotropic voxel spacing of 0.48mm, 0.48mm and 0.70mm.

7.1.4 Usability Testing

We have performed a usability test with two specialists in orthodontics, referred to as user A and B. The experiment we carried out consisted of asking them to segment the mandible and the skull with the MSSSL method. Based on their own knowledge, both had to interactively improve the segmentation until the result is considered “good enough” to use in their clinical applications.

In average, only two iterations were needed to reach the segmentation they considered good, and it took them about one minute per iteration. The total time was, in average, ten minutes (seeding and processing). Figure 51 illustrates the final seed placement of both users and their respective results. Table 8 presents the accuracy achieved.

Table 8 – Jaccard value achieved by users A and B in a mandible and skull segmentation test

	Mandible 1	Mandible 2	Skull 1	Skull 2
User A	0.97	0.95	0.94	0.93
User B	0.93	0.94	0.95	0.96

Before starting the test, we showed the users how to place the seeds only once. We noticed they were able to correct imperfections (noise) after the first or second iterations. We also observed they seeded many slices on the three planes (axial, sagittal and coronal) more than actually needed by the proposed method. Nevertheless, placing extra seeds did not affect the segmentation accuracy. We highlight that the simplicity of our seed placement process allowed users to quickly grasp the idea behind the segmentation process under which our method is based.

Figure 51 – Usability testing: skull segmentation of CBCT 1 by user A, seed placement, and the respective result, in the first row. Mandible segmentation of CBCT 1 by user B in the second row. User A finished the segmentation process after three iterations, whereas user B finished it after only one iteration.

7.2 Method MSM

The same parameters used by Method MSSL, except the threshold t , are employed by Method MSM. Therefore, to evaluate accuracy, and following similar procedure adopted in Method MSSL, we setup $\sigma = 0.8$ and $l = 10$ (See Table 4). We compute accuracy with the Jaccard index and the Dice coefficient. Unlike Method MSSL, we do not create meshes. Instead, we compute Jaccard and Dice directly from the volumetric data.

As discussed in Section 6.3, this method is not suitable for segmenting thin anatomical structures observed in the skull, mainly because it is hard to place seeds on such structures. Thus, evaluation has been performed on CBCT data of the mandible. In addition, we tested this method on a biological image dataset. We used a cat-retina segmentation benchmark dataset consisted of 2D cellular-level nuclei images. The dataset contains 50 images, with respective ground truth created by experts. Images were collected using a laser scanning confocal microscope (GELASCA *et al.*, 2008).

7.2.1 Mandible Segmentation

Figure 52 illustrates the process of seed placement and the resulting segmentation of the most challenging regions (teeth and condyle).

Figure 52 – Segmentation of mandible obtained with method MSM on 4 CBCT images. From left to right: seed placement in the first column, our result in the second column, and ground truth in the third column. Seed placement on thin structures is quite complicated, for example the coronoid process in (d). Besides separating touching lower and upper teeth also needs more precise seeds (j).

Table 9 presents the Jaccard index and Dice coefficient obtained for our four CBCT ground truth. As mentioned before, to segment thinner structures with this method, a more precise seed placement is needed. Notice that internal structures have become part of the segmented regions. For every brush stroke, one segmented solid object is created. As mentioned in section 6.3 this behavior is not an actual advantage in our CBCT application. Nevertheless, it turns out to be very useful in some other applications as the one we present next.

Table 9 – Mandible segmentation accuracy obtained with method MSM

CBCT image	Jaccard	Dice
1	0.944	0.971
2	0.894	0.944
3	0.942	0.970
4	0.896	0.945

7.2.2 Retina Cellular-level Segmentation

In this experiment we used a 2D dataset with 50 cellular level nuclei images from cat retina of $[512 \times 512]$ resolution. Figure 53 illustrates 3 sample images, the obtained segmentation and the respective ground truth (GT). Notice that GT data consists of a single solid object which contains many little circular structures. Table 10 contains the average and standard deviation of Jaccard index and Dice coefficient. As we are dealing with 2D data, we employed super-pixels rather than super-voxels.

Table 10 – Segmentation of cat retina accuracy ratios obtained with method MSM.

	Jaccard	Dice
Mean	0.948	0.973
Standard std	0.016	0.008

Notice that the boundaries of the regions of interest (cellular nuclei) in the original images present both irregular contours and circular shapes (Figure 53.a, 53.d and 53.g). Even though super-pixels adjust accurately to contours, occasionally they do not properly fit small circular shapes. To correct this inaccuracy we apply a morphological erosion and dilation filter, as post-process, with a Parabolic Structuring Function (PSF) to turn irregular shapes into circular ones (BEARE, 2008). This filter is applied as the last step in the process, after dividing the network with our modularity optimization method. Figure 54 shows a “before and after” example of the application of this filter. To evaluate how well our segmentation matches the ground truth boundaries with the erosion and dilation filter, we compute the Hausdorff distance. In average, accuracy was incremented by 42.568%

Figure 53 – Retina cellular-level segmentation obtained with method MSM. From left to right: seed placement in the first column; our results in the second column and the ground truth in the third column.

Figure 54 – Retina cellular-level segmentation: before (a) and after (b) application of the morphological erosion and dilation filter. H is the Hausdorff distance between segmentation and ground truth (c). A close-up view of erosion process within the yellow frame.

7.3 Chapter Notes

In this chapter we presented the main results obtained with methods MSSL and MSM. We used the Jaccard index and the Dice coefficient to quantitatively evaluate segmentation accuracy. In general, our best results revealed quality indices higher than 0.9 (1 being the maximum) for both methods. Besides, processing time of both methods for 3D data was in average 5 minutes. Method MSM took in average less than a second to segment 2D images.

Method MSSL is adequate for segmenting the mandible and the skull in CBCT and CT data. As well as its segmentation quality, one of most important advantages of this method is the simplicity of the seeding process. Seeds can be drawn across bone limits without compromising the quality, as long as they do not “touch” undesired bones. In addition, the Laplace-based graph clustering method, guarantees a unique solution (deterministic) which is desirable in medical image segmentation. Nevertheless, the time complexity to solve the Cholesky factorization is $O(n^3)$. However there exist “fast” algorithms to solve it, which can be used in special cases, for example, the Simplicial LDLT algorithm recommended for very sparse, though not too large, problems (GUENNEBAUD *et al.*, 2010). The linear system of Equation 5.22, used in method MSSL, is sparse and not too large.

In contrast, the most outstanding feature of method MSM is its scalability. By using the modularity optimization strategy it is possible to process huge graphs (million of vertices). Its computational complexity is in average near linear $\simeq O(n)$ on sparse graphs. However, this is a non-deterministic method i.e. for the same input and distinct runs, it may return different results. Nonetheless, quality remains very similar. Another important feature of method MSM is that it incorporates interior parts of the region of interest into the final segmented region. This feature makes the method suitable for segmenting other types of images, as we have shown in Section 7.2.2.

We have conducted a usability test of the method MSSL with specialists in orthodontics. They were able to learn quickly how to interact with the method in order to generate an accurate segmentation of both the mandible and the skull. It took them, in average, 10 minutes (seeding plus processing) to reach good segmentation, according to their own criteria. Besides, the users affirmed that our method was not only easy to understand and handle, but also testified that it produced good segmentation in a very short processing time.

Apart from the segmentation results discussed in section 7.1.1, all other results were obtained with the following parameters: $l = 10$, $\sigma = 0.8$, $t = 200$. We performed our experiments on a Linux workstation (Intel Core i7-2600 CPU 3.40GHz x 4 with 16GB Memory). The C++ source code included the National Library of Medicine Insight Segmentation and Registration Toolkit (ITK) (JOHNSON *et al.*, 2013) and the C++ template library of the linear algebra Eigen (GUENNEBAUD *et al.*, 2010). We have used a “painting tool”, available on 3D-Slicer, to place the seeds. All results were rendered with ITK Snap.

CONCLUSIONS

In this thesis, we addressed the problem of mandible and skull segmentation in CBCT data. We proposed two interactive methods (MSSL and MSM) both based on super-voxels and graph clustering. In general, both methods are divided into two stages. First, the image redundancy is reduced by grouping voxels into super-voxels. Next, a graph whose vertices are super-voxels, is built. Second, user-placed seeds are used to guide the graph clustering algorithm, yielding the final segmentation. The method MSSL is designed for sheet-like structures as the maxilla, whereas the method MSM performs best on mandible including its inner regions.

It was hypothesized that approaches based on super-voxels and graph clustering produce an accurate segmentation of the mandible and the skull in CBCT data, to be used in real clinical applications. We have confirmed this hypothesis experimentally as follows:

1. Detailed evaluation of the parameter space.
2. Segmentation of CT data of patients with missing and broken bones.
3. Comparison with a related work method.
4. Segmentation of a biological image dataset.
5. Usability testing.

Results indicated that method MSSL produced accurate segmentation of both mandible and skull. Not only contours affected by low-contrast (condyle or coronoid process), but also thin bones (maxilla), were precisely segmented. We have shown that the method exhibits low sensitivity to changes in parameter values and does not require precise seed placement. Moreover, method MSSL robustly handles low-contrast, noisy and inhomogeneous data.

The robustness of method MSSL with respect to imprecise seed placement is a consequence of the pre-segmentation stage which also enhances bones, especially thin ones. Once

bones are separated from other tissue types and structures, strokes drawn outside bone regions, i.e., strokes wrongly placed in non-bone regions, have no negative effect on segmentation quality.

On the other hand, method MSM produces accurate segmentation of the mandible, internal region included. Although this is not an actual requirement of our application, this feature turns method MSM suitable for segmenting other medical data for which inner regions are required. However, seeds have to be placed more precisely, i.e. they cannot exceed the limits or contours of the object of interest. It is worth noting that precision in seed placement is also a hard constraint of other seed-based methods (ZHU *et al.*, 2014; CASACA; NONATO; TAUBIN, 2014).

The use of super-voxels made possible a friendly user-interaction scenario. Image segmentation methods based on graph clustering, often creates a graph so that every pixel is mapped as a vertex. Consequently, the adjacency matrix of a graph created from an image with n pixels has n rows and n columns. Graphs built from small to medium size images $\approx [500 \times 500]$ have an “equivalent” cardinality and many algorithms can be applied without considerable restrictions in terms of memory and processing times. Nevertheless, a typical 3D CBCT data may have more than 64 million voxels, yielding an adjacency matrix with 64^2 million of elements. Such a quantity is unsuitable for interactive-based image segmentation applications. Nevertheless, with super-voxels the graph cardinality was reduced by 99%, so that we can apply a precise, though computationally expensive $O(n^3)$, linear system solver as the Cholesky factorization. As a result our methods produced accurate segmentation in only about five minutes, which is suitable for segmenting 3D data. Moreover, it is clearly much faster than manual segmentation.

8.1 Limitations

Separating upper and lower teeth, in a closed-mouth situation, is still a challenging problem. Our method is already capable of separating them by using a few additional brush-strokes, see Figure 40.c. However, visual inspection reveals that accuracy is lower for teeth regions than one would have liked. Due to the complexity of this task, most methods, in general, simply ignore it or handle it separately (LE *et al.*, 2009).

Segmenting sheet-like anatomical structures with method MSM requires seeds to be placed more precisely. This is doable in 2D space where users can spend a little extra effort in this task. However, this is harder in the 3D domain. Therefore, method MSM is not suitable for segmenting thin structures in 3D domain.

8.2 Future Work

Many ideas have emerged during the doctoral period and may be explored in future work.

- Both proposed methods may be used to segment other bones in CT data, such as hip, vertebrae, fingers or ribs. Experiments should be carried out in order to evaluate the performance of our methods in this scenario.
- Teeth segmentation is still an open problem. It will be worth investigating teeth segmentation methods in order to either separate upper from lower teeth or segmenting a single tooth.
- Even though super-pixels(-voxels) boundaries fit accurately image contours for most cases, there are still some irregular shapes or diffuse contours for which super-pixels are not capable of producing accurate contours. Including gradient information, texture or shape into the super-pixels generation process would be an interesting future work.
- The proposed super-voxels algorithm (SUTV) is capable of starting the process with different grid setups e.g. Octree. We have already implemented this feature. However, a quantitative evaluation must be carried out. Experiments with super-pixels and Quadtree have already been performed with natural-scene images and published in ([LINARES *et al.*, 2017a](#)).
- In some cases, it is possible to automatically identify the mandible prior to the segmentation process, as some related work do. We can use this information instead of the user-placed seeds and deliver a fully automatic mandible segmentation method.
- Some tools for medical image processing support plug-ins or extensions¹. This allows developers to integrate their source code as part of such tools and taking advantage of their interfaces. We have partially implemented our methods as 3D-Slicer extensions. A fully implementation will allow releasing our methods not only to our collaborators at UNESP - Araraquara, but also to the general public.

¹ Extensions are software programs created to modify and enhance the functionality of computer software tools.

BIBLIOGRAPHY

ACHANTA, R.; SHAJI, A.; SMITH, K.; LUCCHI, A.; FUA, P.; SÜSSTRUNK, S. **Slic super-pixels**. [S.l.], 2010. Citations on pages [60](#) and [73](#).

ACHANTA, R.; SHAJI, A.; SMITH, K.; LUCCHI, A.; FUA, P.; SÜSSTRUNK, S. Slic super-pixels compared to state-of-the-art superpixel methods. **IEEE Trans. on Pattern Analysis and Machine Intelligence**, v. 34, p. 2274–2282, 2012. Citation on page [73](#).

ANTIGA, L. Generalizing vesselness with respect to dimensionality and shape. **The Insight Journal**, v. 3, p. 1–14, 2007. Citation on page [70](#).

BARRAT, A.; BARTHELEMY, M.; VESPIGNANI, A. **Dynamical processes on complex networks**. [S.l.]: Cambridge university press, 2008. Citation on page [80](#).

BEARE, R. Morphology with parabolic structuring elements. **Insight J**, v. 228, 2008. Citation on page [115](#).

BERTRAM, M.; DUCHAINEAU, M. A.; HAMANN, B.; JOY, K. I. Generalized b-spline subdivision-surface wavelets for geometry compression. **IEEE transactions on visualization and computer graphics**, IEEE, v. 10, n. 3, p. 326–338, 2004. Citation on page [59](#).

BICHOT, C.-E.; SIARRY, P. **Graph partitioning**. [S.l.]: John Wiley & Sons, 2013. Citation on page [85](#).

BIGGS, N. **Algebraic graph theory**. [S.l.]: Cambridge university press, 1993. Citation on page [81](#).

BLONDEL, V. D.; GUILLAUME, J.-L.; LAMBIOTTE, R.; LEFEBVRE, E. Fast unfolding of communities in large networks. **Journal of statistical mechanics: theory and experiment**, IOP Publishing, n. 10, p. P10008, 2008. Citations on pages [33](#), [87](#), [89](#), and [92](#).

BOAS, F. E.; FLEISCHMANN, D. Ct artifacts: causes and reduction techniques. **Imaging Med**, v. 4, n. 2, p. 229–240, 2012. Citation on page [40](#).

BOYKOV, Y.; FUNKA-LEA, G. Graph cuts and efficient nd image segmentation. **International journal of computer vision**, Springer, v. 70, n. 2, p. 109–131, 2006. Citation on page [86](#).

BOYKOV, Y.; VEKSLER, O.; ZABIH, R. Fast approximate energy minimization via graph cuts. **IEEE Transactions on pattern analysis and machine intelligence**, IEEE, v. 23, n. 11, p. 1222–1239, 2001. Citation on page [62](#).

BOYKOV, Y. Y.; JOLLY, M.-P. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. **8th IEEE Int. Conf. on Computer Vision (ICCV)**, v. 1, p. 105–112, 2001. Citations on pages [48](#) and [86](#).

BRANDARIZ, M.; BARREIRA, N.; PENEDO, M. G.; SUÁREZ-CUNQUEIRO, M. Automatic segmentation of the mandible in cone-beam computer tomography images. **27th IEEE Int. Symp. on Computer-Based Medical Systems**, p. 467–468, 2014. Citations on pages [30](#) and [55](#).

BRANDES, U. **Network analysis: methodological foundations**. [S.l.]: Springer Science & Business Media, 2005. Citations on pages 79 and 84.

BRANDES, U.; DELLING, D.; GAERTLER, M.; GORKE, R.; HOEFER, M.; NIKOLOSKI, Z.; WAGNER, D. On modularity clustering. **IEEE transactions on knowledge and data engineering**, IEEE, v. 20, n. 2, p. 172–188, 2008. Citations on pages 87 and 89.

BRENNER, D. J.; HALL, E. J. Computed tomography an increasing source of radiation exposure. **New England Journal of Medicine**, Mass Medical Soc, v. 357, n. 22, p. 2277–2284, 2007. Citation on page 42.

BUZUG, T. M. **Computed tomography: from photon statistics to modern cone-beam CT**. [S.l.]: Springer Science & Business Media, 2008. Citations on pages 39 and 40.

CABEZAS, M.; OLIVER, A.; LLADÓ, X.; FREIXENET, J.; CUADRA, M. B. A review of atlas-based segmentation for magnetic resonance brain images. **Computer methods and programs in biomedicine**, Elsevier, v. 104, n. 3, p. e158–e177, 2011. Citation on page 48.

CANNY, J. A computational approach to edge detection. **IEEE Transactions on pattern analysis and machine intelligence**, n. 6, p. 679–698, 1986. Citation on page 60.

CASACA, W.; NONATO, L. G.; TAUBIN, G. Laplacian coordinates for seeded image segmentation. **IEEE Conf. on Computer Vision and Pattern Recognition**, p. 384–391, 2014. Citations on pages 86 and 120.

CATMULL, E.; CLARK, J. Recursively generated b-spline surfaces on arbitrary topological meshes. **Computer-aided design**, Elsevier, v. 10, n. 6, p. 350–355, 1978. Citation on page 58.

CHANG, Y.-B.; XIA, J. J.; YUAN, P.; KUO, T.-H.; XIONG, Z.; GATENO, J.; ZHOU, X. 3d segmentation of maxilla in cone-beam computed tomography imaging using base invariant wavelet active shape model on customized two-manifold topology. **Journal of X-ray science and technology**, IOS Press, v. 21, n. 2, p. 251–282, 2013. Citations on pages 16, 29, 30, 58, and 59.

ÇİĞLA, C.; ALATAN, A. A. Efficient graph-based image segmentation via speeded-up turbo pixels. **IEEE Int. Conf. on Image Processing**, p. 3013–3016, 2010. Citation on page 75.

CLAUSET, A.; NEWMAN, M. E.; MOORE, C. Finding community structure in very large networks. **Physical review E**, APS, v. 70, n. 6, p. 066111, 2004. Citations on pages 87 and 89.

CONSTINE, L.; KONSKI, A.; EKHOLM, S.; MCDONALD, S.; RUBIN, P. Adverse effects of brain irradiation correlated with mr and ct imaging. **International Journal of Radiation Oncology* Biology* Physics**, Elsevier, v. 15, n. 2, p. 319–330, 1988. Citation on page 42.

COSTA, L. d. F.; RODRIGUES, F. A.; TRAVIESO, G.; BOAS, P. R. V. Characterization of complex networks: A survey of measurements. **Advances in physics**, Taylor & Francis, v. 56, n. 1, p. 167–242, 2007. Citation on page 81.

DEO, N. **Graph theory with applications to engineering and computer science**. [S.l.]: Courier Dover Publications, 2017. Citation on page 85.

DESCOTEAUX, M.; AUDETTE, M.; CHINZEI, K.; SIDDIQI, K. Bone enhancement filtering: application to sinus bone segmentation and simulation of pituitary surgery. **Computer Aided Surgery**, p. 247–255, 2006. Citations on pages 30, 68, and 70.

DOEL, T.; GAVAGHAN, D. J.; GRAU, V. Review of automatic pulmonary lobe segmentation methods from ct. **Computerized Medical Imaging and Graphics**, Elsevier, v. 40, p. 13–29, 2015. Citation on page 68.

EBERHART, R.; KENNEDY, J. A new optimizer using particle swarm theory. In: IEEE. **Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on**. [S.l.], 1995. p. 39–43. Citation on page 55.

ELANGO VAN, A.; JEYASEELAN, T. Medical imaging modalities: A survey. In: IEEE. **Emerging Trends in Engineering, Technology and Science (ICETETS), International Conference on**. [S.l.], 2016. p. 1–4. Citations on pages 37 and 38.

FEDOROV, A.; BEICHEL, R.; KALPATHY-CRAMER, J.; FINET, J.; FILLION-ROBIN, J.-C.; PUJOL, S.; BAUER, C.; JENNINGS, D.; FENNESSY, F.; SONKA, M. *et al.* 3d slicer as an image computing platform for the quantitative imaging network. **Magnetic Resonance Imaging**, p. 1323–1341, 2012. <https://www.slicer.org>, (accessed 08.08.17). Citations on pages 30 and 38.

FIEDLER, M. Algebraic connectivity of graphs. **Czechoslovak mathematical journal**, Institute of Mathematics, Academy of Sciences of the Czech Republic, v. 23, n. 2, p. 298–305, 1973. Citation on page 84.

FORTUNATO, S. Community detection in graphs. **Physics reports**, Elsevier, v. 486, n. 3, p. 75–174, 2010. Citations on pages 79, 81, 84, 87, and 89.

FRANGI, A. F.; NIESSEN, W. J.; VINCKEN, K. L.; VIERGEVER, M. A. Multiscale vessel enhancement filtering. In: SPRINGER. **International Conference on Medical Image Computing and Computer-Assisted Intervention**. [S.l.], 1998. p. 130–137. Citation on page 70.

GELASCA, E. D.; BYUN, J.; OBARA, B.; MANJUNATH, B. Evaluation and benchmark for biological image segmentation. In: IEEE. **Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on**. [S.l.], 2008. p. 1816–1819. Citation on page 113.

GRONE, R.; MERRIS, R.; SUNDER, V. S. The laplacian spectrum of a graph. **SIAM Journal on Matrix Analysis and Applications**, SIAM, v. 11, n. 2, p. 218–238, 1990. Citation on page 81.

GUENNEBAUD, G.; JACOB, B.; AVERY, P.; BACHRACH, A.; BARTHELEMY, S. **Eigen v3**. 2010. <http://eigen.tuxfamily.org>, (accessed 08.08.17). Citation on page 117.

GUIMOND, A.; MEUNIER, J.; THIRION, J.-P. Average brain models: A convergence study. **Computer vision and image understanding**, Elsevier, v. 77, n. 2, p. 192–210, 2000. Citations on pages 16 and 49.

GUO, Z.; ZHANG, L.; ZHANG, D. A completed modeling of local binary pattern operator for texture classification. **IEEE Transactions on Image Processing**, IEEE, v. 19, n. 6, p. 1657–1663, 2010. Citation on page 60.

GUTIERREZ, M. S. **A oferta de tomógrafo computadorizado para o tratamento do acidente vascular cerebral agudo, no Brasil, sob o ponto de vista das desigualdades geográficas e sociais**. Phd Thesis (PhD Thesis), 2009. Citation on page 38.

HARDING, G. X-ray scatter tomography for explosives detection. **Radiation Physics and Chemistry**, Elsevier, v. 71, n. 3, p. 869–881, 2004. Citation on page 37.

HO, T. K. Random decision forests. In: IEEE. **Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on**. [S.l.], 1995. v. 1, p. 278–282. Citation on page 48.

HORN, B. K. Closed-form solution of absolute orientation using unit quaternions. **JOSA A**, Optical Society of America, v. 4, n. 4, p. 629–642, 1987. Citation on page 59.

HSIEH, J. **Computed tomography: principles, design, artifacts, and recent advances**. [S.l.]: SPIE press, 2003. Citation on page 41.

HU, G. *et al.* Survey of recent volumetric medical image segmentation techniques. In: **Biomedical Engineering**. [S.l.]: InTech, 2009. Citation on page 47.

JAIN, A. K. **Fundamentals of digital image processing**. [S.l.]: Prentice-Hall, Inc., 1989. Citation on page 73.

JERMAN, T.; PERNUS, F.; LIKAR, B.; SPICLIN, Z. Enhancement of vascular structures in 3d and 2d angiographic images. **IEEE Transactions on Medical Imaging**, v. 35, n. 9, p. 2107–2118, Sept 2016. ISSN 0278-0062. Citation on page 70.

JOHNSON, H. J.; MCCORMICK, M.; IBÁÑEZ, L.; CONSORTIUM, T. I. S. **The ITK Software Guide**. Third. [S.l.], 2013. [Http://www.itk.org/ItkSoftwareGuide.pdf](http://www.itk.org/ItkSoftwareGuide.pdf), (accessed 08.16.17). Citations on pages 53 and 117.

JUAN, O.; BOYKOV, Y. Active graph cuts. In: IEEE. **Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on**. [S.l.], 2006. v. 1, p. 1023–1029. Citation on page 86.

KAINMUELLER, D.; LAMECKER, H.; SEIM, H.; ZINSER, M.; ZACHOW, S. Automatic extraction of mandibular nerve and bone from cone-beam ct data. **Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009**, Springer, p. 76–83, 2009. Citations on pages 16, 56, and 57.

KASS, M.; WITKIN, A.; TERZOPOULOS, D. Snakes: Active contour models. **International journal of computer vision**, Springer, v. 1, n. 4, p. 321–331, 1988. Citations on pages 47 and 56.

KOLMOGOROV, V.; ZABIN, R. What energy functions can be minimized via graph cuts? **IEEE Trans. on Pattern Analysis and Machine Intelligence**, p. 147–159, 2004. Citation on page 30.

KRILLE, L.; DREGER, S.; SCHINDEL, R.; ALBRECHT, T.; ASMUSSEN, M.; BARKHAUSEN, J.; BERTHOLD, J.; CHAVAN, A.; CLAUSSEN, C.; FORSTING, M. *et al.* Risk of cancer incidence before the age of 15 years after exposure to ionising radiation from computed tomography: results from a german cohort study. **Radiation and environmental biophysics**, Springer, v. 54, n. 1, p. 1–12, 2015. Citations on pages 41 and 42.

LAMECKER, H.; LANGE, T.; SEEBASS, M. **Segmentation of the liver using a 3D statistical shape model**. [S.l.]: Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2004. Citation on page 51.

LAMECKER, H.; ZACHOW, S.; WITTMERS, A.; WEBER, B.; HEGE, H.; ISHOLTZ, B.; STILLER, M. Automatic segmentation of mandibles in low-dose ct-data. **International Journal of Computer Assisted Radiology and Surgery**, SPRINGER SCIENCE+ BUSINESS MEDIA, v. 1, p. 393, 2006. Citations on pages 16, 51, and 52.

LE, B. H.; DENG, Z.; XIA, J.; CHANG, Y.-B.; ZHOU, X. An interactive geometric technique for upper and lower teeth segmentation. In: SPRINGER. **International Conference on Medical Image Computing and Computer-Assisted Intervention**. [S.l.], 2009. p. 968–975. Citations on pages 50 and 120.

LESKOVEC, J.; MCAULEY, J. J. Learning to discover social circles in ego networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2012. p. 539–547. Citation on page 80.

LI, W.; MAGNOTTA, V. New functionalities for spherical demons registration. **Kitware Source - Software Developers Quarterly**, p. 08–10, 01 2011. https://blog.kitware.com/wp-content/uploads/2016/01/kitware_quarterly0111.pdf, (accessed 08.08.17). Citation on page 103.

LINARES, O. C.; BOTELHO, G.; RODRIGUES, F.; NETO, J. B. Segmentation of large images with complex networks. **IEEE 25th SIBGRAPI Conf. on Graphics, Patterns and Images**, p. 24–31, 2012. Citations on pages 73 and 75.

LINARES, O. C.; BOTELHO, G.; RODRIGUES, F.; NETO, J. B. An adjustable error measure for image segmentation evaluation. In: **2015 28th SIBGRAPI Conference on Graphics, Patterns and Images**. [S.l.: s.n.], 2015. p. 80–86. ISSN 1530-1834. Citation on page 34.

LINARES, O. C.; BOTELHO, G.; RODRIGUES, F. A.; NETO, J. B. Segmentation of large images based on super-pixels and community detection in graphs. **IET Image Processing**, p. 60–80, 2017. Citations on pages 34, 76, 78, and 121.

LINARES, O. C.; BOTELHO, G. M.; RODRIGUES, F. A.; NETO, J. B. Mandible and skull segmentation in cone beam computed tomography using super-voxels and graph clustering. **Submitted to: The Visual Computer International Journal of Computer Graphics**, 2017. Citations on pages 34 and 35.

LORENSEN, W. E.; CLINE, H. E. Marching cubes: A high resolution 3d surface construction algorithm. **ACM Siggraph Comput. Graph.**, v. 21, p. 163–169, 1987. Citation on page 103.

LOUBELE, M.; MAES, F.; SCHUTYSER, F.; MARCHAL, G.; JACOBS, R.; SUETENS, P. Assessment of bone segmentation quality of cone-beam ct versus multislice spiral ct: a pilot study. **Oral Surgery, Oral Medicine, Oral Pathology, Oral Radiology, and Endodontology**, v. 102, p. 225–234, 2006. Citations on pages 29 and 30.

LUCCHI, A.; SMITH, K.; ACHANTA, R.; KNOTT, G.; FUA, P. Supervoxel-based segmentation of mitochondria in em image stacks with learned shape features. **IEEE Transactions on Medical Imaging**, v. 31, n. 2, p. 474–486, 2012. ISSN 0278-0062. Citations on pages 16, 60, 61, and 75.

MACIA, I. Generalized computation of gaussian derivatives using itk. **The Insight Journal**, p. 1–14, 2007. Citation on page 69.

MAHAPATRA, D.; BUHMANN, J. M. Prostate mri segmentation using learned semantic knowledge and graph cuts. **IEEE Transactions on Biomedical Engineering**, IEEE, v. 61, n. 3, p. 756–764, 2014. Citations on pages 16, 62, and 63.

MALLIAROS, F. D.; VAZIRGIANNIS, M. Clustering and community detection in directed networks: A survey. **Physics Reports**, Elsevier, v. 533, n. 4, p. 95–142, 2013. Citation on page 79.

MARTIN, D.; FOWLKES, C.; TAL, D.; MALIK, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: IEEE. **Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on**. [S.l.], 2001. v. 2, p. 416–423. Citations on pages 20 and 134.

MATHEWS, J. D.; FORSYTHE, A. V.; BRADY, Z.; BUTLER, M. W.; GOERGEN, S. K.; BYRNES, G. B.; GILES, G. G.; WALLACE, A. B.; ANDERSON, P. R.; GUIVER, T. A. *et al.* Cancer risk in 680 000 people exposed to computed tomography scans in childhood or adolescence: data linkage study of 11 million australians. **Bmj**, British Medical Journal Publishing Group, v. 346, p. f2360, 2013. Citation on page 42.

MERRIS, R. Laplacian matrices of graphs: a survey. **Linear Algebra and its Applications**, v. 197, p. 143 – 176, 1994. ISSN 0024-3795. Available: <<http://www.sciencedirect.com/science/article/pii/0024379594904863>>. Citations on pages 82 and 84.

MILES, D. A.; DANFORTH, R. A. A clinician's guide to understanding cone beam volumetric imaging (cbvi). **Academy of Dental Therapeutics and Stomatology**, p. 1–13, 2007. Citations on pages 29 and 68.

MOZZO, P.; PROCACCI, C.; TACCONI, A.; MARTINI, P. T.; ANDREIS, I. B. A new volumetric ct machine for dental imaging based on the cone-beam technique: preliminary results. **European Radiology**, p. 1558–1564, 1998. Citations on pages 29, 42, and 43.

NETTER, F. H. **Atlas of Human Anatomy, Professional Edition**. [S.l.]: Elsevier Health Sciences, 2014. Citations on pages 15 and 44.

NEWMAN, M. Finding community structure in networks using the eigenvectors of matrices. **Physical review E**, v. 74, n. 3, p. 036104, 2006. Citations on pages 85 and 88.

NEWMAN, M.; GIRVAN, M. Finding and evaluating community structure in networks. **Physical review E**, APS, v. 69, n. 2, p. 026113, 2004. Citations on pages 87 and 88.

NEWMAN, M. E. The structure and function of complex networks. **SIAM review**, SIAM, v. 45, n. 2, p. 167–256, 2003. Citation on page 81.

OLIVEIRA, T. B. de; ZHAO, L. Complex network community detection based on swarm aggregation. In: IEEE. **Natural Computation, 2008. ICNC'08. Fourth International Conference on**. [S.l.], 2008. v. 7, p. 604–608. Citation on page 87.

ORANSKY, I. Sir godfrey n hounsfield. **The Lancet**, Elsevier, v. 364, n. 9439, p. 1032, 2004. Citation on page 38.

PENG, B.; ZHANG, L.; ZHANG, D. A survey of graph theoretical approaches to image segmentation. **Pattern Recognition**, v. 46, p. 1020–1038, 2013. Citations on pages 30 and 95.

PHAM, D. L.; XU, C.; PRINCE, J. L. Current methods in medical image segmentation. **Annual review of biomedical engineering**, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 2, n. 1, p. 315–337, 2000. Citation on page 37.

PIXMEO-SARL [dataset]. **Osirix DICOM image sample sets**. 2016. [Http://www.osirix-viewer.com/resources/dicom-image-library](http://www.osirix-viewer.com/resources/dicom-image-library). (accessed 12.01.16). Citations on pages 15, 39, and 111.

RATLIFF, S. T. Webb's physics of medical imaging. **Medical Physics**, Wiley Online Library, v. 40, n. 9, 2013. Citations on pages 38 and 39.

RICHMOND, C. Sir godfrey hounsfield. **BMJ: British Medical Journal**, p. 687–687, 2004. Citation on page 29.

SCARFE, W. C.; FARMAN, A. G.; SUKOVIC, P. Clinical applications of cone-beam computed tomography in dental practice. **Journal-Canadian Dental Association**, p. 75–80, 2006. Citations on pages 29 and 42.

SEERAM, E. **Computed Tomography: Physical Principles, Clinical Applications, and Quality Control**. [S.l.]: Elsevier Health Sciences, 2015. Citations on pages 15, 37, 38, 39, 41, and 42.

SHARP PAOLO ZAFFINO, K. F. G.; RAUDASCHL, P. **A Public Domain Database for Computational Anatomy (PDDCA)**. 2016. [Http://www.imagenglab.com/newsite/pddca/](http://www.imagenglab.com/newsite/pddca/). (accessed 08.09.17). Citations on pages 15 and 40.

SHINTAKU, W. H.; VENTURIN, J. S.; AZEVEDO, B.; NOUJEIM, M. Applications of cone-beam computed tomography in fractures of the maxillofacial complex. **Dental traumatology**, Wiley Online Library, v. 25, n. 4, p. 358–366, 2009. Citations on pages 42 and 43.

SINGH, S.; SINGH, M. Explosives detection systems (eds) for aviation security. **Signal Processing**, Elsevier, v. 83, n. 1, p. 31–55, 2003. Citation on page 37.

SMITH, K.; CARLETON, A.; LEPETIT, V. Fast ray features for learning irregular shapes. In: IEEE. **Computer Vision, 2009 IEEE 12th International Conference on**. [S.l.], 2009. p. 397–404. Citation on page 60.

SOLUTIONS, D. I. . M. **Dolphin Imaging**. 2017. [Http://www.dolphinimaging.com](http://www.dolphinimaging.com). (accessed 08.08.17). Citations on pages 30 and 38.

SOMMER, C.; STRAEHLE, C.; KOETHE, U.; HAMPRECHT, F. A. Ilastik: Interactive learning and segmentation toolkit. In: IEEE. **Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on**. [S.l.], 2011. p. 230–233. Citation on page 61.

SPAMPINATO, C.; PINO, C.; GIORDANO, D.; LEONARDI, R. Automatic 3d segmentation of mandible for assessment of facial asymmetry. **IEEE International Symposium on Medical Measurements and Applications Proceedings (MeMeA)**, p. 1–4, 2012. Citations on pages 16, 29, 30, 53, 54, and 55.

ŠPELIC, D.; MONGUS, D.; ŽALIK, B. Lossless compression of segmented ct medical images according to the hounsfield scale. In: IEEE. **Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on**. [S.l.], 2011. p. 297–301. Citation on page 39.

VEZHNEVETS, V.; KONOUCHE, V. Growcut: Interactive multi-label nd image segmentation by cellular automata. In: **proc. of Graphicon**. [S.l.: s.n.], 2005. v. 1, p. 150–156. Citation on page 63.

VOS, W. D.; CASSELMAN, J.; SWENNEN, G. Cone-beam computerized tomography (cbct) imaging of the oral and maxillofacial region: a systematic review of the literature. **International journal of oral and maxillofacial surgery**, Elsevier, v. 38, n. 6, p. 609–625, 2009. Citations on pages 43 and 68.

WANG, L.; CHEN, K. C.; GAO, Y.; SHI, F.; LIAO, S.; LI, G.; SHEN, S. G.; YAN, J.; LEE, P. K.; CHOW, B. *et al.* Automated bone segmentation from dental cbct images using patch-based sparse representation and convex optimization. **Medical physics**, American Association of Physicists in Medicine, p. 043503–043503, 2014. Citation on page 30.

WANG, L.; GAO, Y.; SHI, F.; LI, G.; CHEN, K.-C.; TANG, Z.; XIA, J. J.; SHEN, D. Automated segmentation of dental cbct image with prior-guided sequential random forests. **Medical Physics**, v. 43, p. 336–346, 2016. Citations on pages 16, 30, 49, and 51.

WATKINS, D. S. **Fundamentals of matrix computations**. [S.l.]: John Wiley & Sons, 2004. Citation on page 87.

XIA, J. J.; GATENO, J.; TEICHGRAEBER, J. F. A new clinical protocol to evaluate cranio-maxillofacial deformity and to plan surgical correction. **Journal of oral and maxillofacial surgery: official journal of the American Association of Oral and Maxillofacial Surgeons**, NIH Public Access, p. 2093–2106, 2009. Citation on page 29.

YI, F.; MOON, I. Image segmentation: A survey of graph-cut methods. **IEEE Int. Conf. on Systems and Informatics (ICSAI)**, p. 1936–1941, 2012. Citations on pages 30, 48, and 86.

YUSHKEVICH, P. A.; PIVEN, J.; HAZLETT, H. C.; SMITH, R. G.; HO, S.; GEE, J. C.; GERIG, G. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. **Neuroimage**, p. 1116–1128, 2006. [Http://www.itksnap.org](http://www.itksnap.org), (accessed 08.08.17). Citations on pages 30, 38, and 47.

ZHANG, J.; YAN, C.-H.; CHUI, C.-K.; ONG, S.-H. Fast segmentation of bone in ct images using 3d adaptive thresholding. **Computers in biology and medicine**, Elsevier, v. 40, n. 2, p. 231–236, 2010. Citation on page 67.

ZHANG, Y. J. A survey on evaluation methods for image segmentation. **Pattern recognition**, v. 29, p. 1335–1346, 1996. Citation on page 103.

ZHENG, Y.; COMANICIU, D. **Marginal Space Learning for Medical Image Analysis: Efficient Detection and Segmentation of Anatomical Structures**. [S.l.]: Springer-Verlag New York, 2014. Citations on pages 30, 47, and 48.

ZHU, L.; KOLESOV, I.; GAO, Y.; KIKINIS, R.; TANNENBAUM, A. An effective interactive medical image segmentation method using fast growcut. **Workshop on Interactive Medical Image Computing (MICCAI)**, p. 1–9, 2014. Citations on pages 16, 63, 64, 104, and 120.

GLOSSARY

Bounding Box: The minimum bounding or enclosing box for a set of points S .

Cephalometric analysis: cephalometry is the study and measurement of the dental and skeletal relationships of a human skull, and cephalometric analysis refers to its clinical application.

Connected Component: in image processing, it refers to groups of pixels so-called *components* where all pixels share similar pixel intensity values and are in some way connected with each other.

Convex hull: given a set x of points in the Euclidean space, it is the smallest convex set that enclosed x . The convex hull can be visually interpreted as a rubber band stretched around x .

Eigenmode: an eigenmode is a normal mode of vibration of an oscillating system where various parts move all together at the same frequency.

Framework: is an abstraction that joins common codes among various software projects providing generic functionality. Frameworks are designed with the intent of facilitating software development, enabling designers and programmers to spend more time determining software requirements than with low-level system details.

Hydroxyl radical: is the neutral form of the hydroxide ion. It is a powerful oxidizing agent, capable of reacting with chemicals, including organic pollutants and inhibitors.

Ionizing radiation: is the radiation that carries sufficient energy to liberate electrons from atoms or molecules.

Moore neighborhood: is a 2D square which is composed of a central cell and eight cells which surround it.

Nonsyndromic: Not part of a syndrome.

Suture: is a fibrous joint only found in the skull. The bones are bound together by these fibres.

Wavelet: a wavelet is an oscillation whose amplitude begins at zero, increases, and then decreases back to zero.

MISCELLANEOUS SEGMENTATION

Figure 55 – From top to bottom: 1) a regular mandible, 2) a skull after surgical repair of facial deformity, and 3) a mandible with missing teeth.

Figure 56 – Segmentation obtained with method MSM, images from Berkeley dataset ([MARTIN *et al.*, 2001](#)).