

**TECHNICAL UNIVERSITY OF CRETE  
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING  
DIGITAL IMAGE SIGNALS AND IMAGE PROCESSING LAB**



# **“Reconstruction and graphical modeling of aorta from MRI images”**

**Panagiotis Koulountzios**

**Thesis Committee**

**Professor Michael Zervakis (ECE)**

**Professor Georgios Stavroulakis (PEM)**

**Assistant Professor Vasilios Samoladas (ECE)**

**Chania, October 2017**

**ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ**  
**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**  
**ΕΡΓΑΣΤΗΡΙΟ ΨΗΦΙΑΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ ΚΑΙ ΕΙΚΟΝΑΣ**



## **“Ανακατασκευή και γραφική μοντελοποίηση αορτής από εικόνες MRI”**

**Παναγιώτης Κουλούντζιος**

**Εξεταστική Επιτροπή**  
**Καθηγητής Μιχάλης Ζερβάκης (ΗΜΜΥ)**  
**Καθηγητής Γεώργιος Σταυρουλάκης (ΜΠΔ)**  
**Επίκουρος Καθηγητής Βασίλειος Σαμολαδάς (ΗΜΜΥ)**

**Χανιά, Οκτώβριος 2017**

# Περίληψη

Τα καρδιαγγειακά νοσήματα σήμερα αποτελούν μεγάλο ιατρικό, αλλά και κοινωνικό πρόβλημα, καθώς κατέχουν την πρώτη θέση στις αιτίες θανάτου τόσο στην Ευρώπη, όσο και σε παγκόσμιο επίπεδο. Μέχρι στιγμής οι απεικονιστικές μέθοδοι και οι ραγδαία ανάπτυξη αυτών την προηγούμενη περίοδο έχουν φέρει πολλαπλά οφέλη για την εντόπιση και τη διάγνωση πολλών καρδιαγγειακών ασθενειών (αθηροσκλήρωση, ανεύρυσμα, θρομβώσεις). Οι εφαρμογές του μαγνητικού τομογράφου αποτελούν παράδειγμα τέτοιων μεθόδων, οι οποίες επεξεργάζονται τις απεικονιστικές ακολουθίες και ανασυνθέτουν την πληροφορία στις τρεις διαστάσεις χρησιμοποιώντας αδόμητα δίκτυα γεωμετρίας καθώς τα μοντέλα που ανακατασκευάζουν προορίζονται μόνο για απεικόνιση. Παρ' όλα αυτά, μετά από ενδελεχή μελέτη της σχετικής βιβλιογραφίας, εντοπίζεται μεγάλη ανάγκη προσομοίωσης της γεωμετρίας με δομημένα πλέγματα ώστε να μπορεί να αξιοποιηθεί από ειδικές μηχανικές αναλύσεις. Σε αυτή την κατεύθυνση γίνονται προσπάθειες να συνδεθούν τα πεδία της ιατρικής τρισδιάστατης απεικόνισης με αυτά της υπολογιστικής σχεδίασης (CAD) και της υπολογιστικής ανάλυσης (CAE) με στόχο την εμβιομηχανική ανάλυση με μεθόδους υπολογιστικής μηχανικής.

Η παρούσα διπλωματική πρόταση πραγματεύεται την αυτοματοποιημένη τρισδιάστατη ανακατασκευή της ανθρώπινης θωρακικής αορτής από τα ιατρικά απεικονιστικά δεδομένα μαγνητικού ή αξονικού τομογράφου και την μοντελοποίηση της με τέτοια γεωμετρία ώστε να μπορεί τα καταστεί δυνατή η εμβιο-μηχανική ανάλυση του οργάνου. Έχει γίνει χρήση εξειδικευμένων μεθόδων επεξεργασίας ιατρικών εικόνων ώστε να μπορέσουμε να εντοπίσουμε και να απομονώσουμε την κοιλότητα της αορτής και να εντοπιστεί το εξωτερικό τοίχωμα της μεγάλης αρτηρίας. Έπειτα κάνοντας χρήση μεθόδων τρισδιάστατης ανακατασκευής από ιατρικές εικόνες δημιουργούμε το αρχικό μοντέλο. Τέλος αξιοποιώντας τεχνικές υπολογιστικής γεωμετρίας στοχεύουμε τη δημιουργία του τελικού αξιόπιστου μοντέλου με δομημένο δίκτυο γεωμετρίας. Χρησιμοποιούμε για την τρισδιάστατη ανακατασκευή NURBS επιφάνειες, οι οποίες μας επιτρέπουν την πιο πιστή αναπαράσταση ως προς το φυσικό μοντέλο.

Το NURBS μοντέλο που κατασκευάζουμε είναι και το κατάλληλο μοντέλο προς αξιοποίηση από τις μεθόδους ανάλυσης, για τις οποίες προορίζεται η εφαρμογή μας (Πεπερασμένα Στοιχεία, Ισογεωμετρική ανάλυση). Στόχος των μεθόδων ανάλυσης είναι ο υπολογισμός των ίδιων μεγεθών και συγκεκριμένα του πεδίου των μετατοπίσεων (displacement field), των ανοιγμένων παραμορφώσεων (strain field) και των τάσεων (stress field) των τοιχωμάτων της αορτής.

# Abstract

This thesis was motivated by the need for both quick and accurate computer-aided measurement of human organs, which is vital for an early medical diagnosis. It deals with the thoracic aorta, whose abnormalities are often related to cardiovascular diseases. Also, it presents an innovative approach for semi-automated reconstruction and NURBS-based simulation, which has been applied on real patient data sets. We developed a basic pipeline, which starts with processing MRI DICOM files from the thoracic region. In the first stage, image processing techniques have been used to enhance the quality of the images and then the tracking of the aortic cross-sections' boundaries ends up to the corresponding point cloud. Based on this cloud, a reverse engineering method generates Non-Uniform Rational B-Spline (NURBS) surfaces. This is accomplished by fitting cubic B-Splines to the cross-sections, which generate the 2D NURBS model, means the outer surface of the aorta. Our methodology was tested and evaluated on real MRI data from three different patients and the results indicate good potential for semi or even automated development of a personalized aortic model.

# Acknowledgements

A good supervisor not only teaches you how to carry out researches, but also helps you to further discover your life; I am very lucky because my supervisors are the persons of this kind. I would like to thank Prof. Michalis Zervakis and Prof. George Stauroulakis for their precious guidance in the past two years. Without them, my thesis cannot be accomplished, and it is truly a pleasure to have them as my supervisors.

I want to show my appreciations to my parents and my twin sister, Aglaia. They are always encouraging me with whatever I decide to do and be at my side whenever I am facing challenges.

Moreover I would like to special thank Eleni, who support me and help me significantly through the difficulties of my work.

A word of thanks goes to all my friends in TUC, with whom I have shared my memorable times and from whom I got a lot of helps during my daily life. I am pleased to meet them and have them as my friends.

Last but not least, I would like to thank Panagiotis Karakitsios, a close friend, who was my first inspiration about the mechanical problems in cardiovascular system and his ideas and personality gave to me the strength to keep going forward and work hard.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Thesis area of interest	1
1.2	State of the art	2
1.3	Thesis contribution	3
1.4	Composites steps of modeling and thesis outline	4
<b>2</b>	<b>Anatomy – Physiology of aorta</b>	<b>6</b>
2.1	Anatomy of the aorta	6
2.2	Histology of the aorta	10
<b>3</b>	<b>Analytic background</b>	<b>12</b>
3.1	Medical imaging modalities	12
3.1.1	Computed tomography angiography	12
3.1.2	Magnetic resonance angiography	13
3.2	DICOM	15
3.3	Image processing and techniques	17
3.3.1	Introduction	17
3.3.2	Image enhancement	19
3.3.3	Image segmentation	22
3.4	Image Registration	29
3.4.1	Medical visualization	32
3.5	NURBS- Non Uniform Rational B-Splines	38
3.5.1	NURBS Basis Function	40
3.5.2	NURBS Shape Functions	41
3.5.3	NURBS curves and surfaces	42
<b>4</b>	<b>Proposed approach of aorta reconstruction</b>	<b>44</b>
4.1	Image Processing, Segmentation and Reconstruction	44
4.1.1	Pre-processing	44
4.1.2	Cross-section detection (Segmentation)	49
4.1.3	Artery tracking (Registration)	52
4.1.4	Artery model (Reconstruction)	57
4.2	NURBS fitting techniques	59

4.2.1	2D curves fitting .....	60
4.2.2	3D Surface Generation.....	68
5	<b>Experimental results</b> .....	71
5.1	Evaluation of image segmentation.....	71
5.2	Evaluation of NURBS model .....	78
6	<b>Conclusions</b> .....	82
7	<b>References</b> .....	83

# List of Figures

<b>Figure 2.1</b> Normal size of thoracic aortic segments and characterization of them. [12] .....	6
<b>Figure 2.2</b> On the left side is represented a graphic depiction of the aorta. On the right side is depicted a segmental division of the aorta: the aortic root (LIGHT BLUE), the sinotubular junction (GREEN), the ascending aorta (YELLOW), the aortic arch (DARK BLUE), the isthmus and descending (thoracic) aorta (RED), and the abdominal aorta (PINK).[13] .....	8
<b>Figure 2.3</b> The thoracic aorta. In this picture are emphasized the three supra-aortic branches - i.e., the brachiocephalic trunk, the left common carotid artery, and left subclavian artery. The figure shows also the coronary arteries (left and right coronary arteries). Those arteries are responsible to carry the oxygenated blood to the heart. ....	9
<b>Figure 2.4</b> Structural organization and composition of the three different layers in the coronary vessel wall [13]. ....	10
<b>Figure 3.1</b> Spiral CT scanning principle. The X-rays source and oppositely mounted detectors in rows perform two motions: 1. rotation around the axis, 2. longitudinal movement along the axis. In this way, a large anatomic volume can be scanned rapidly [16]. ....	13
<b>Figure 3.2</b> Electromagnetic spectrum ( <a href="https://www.miniphysics.com/electromagnetic-spectrum_25.html">https://www.miniphysics.com/electromagnetic-spectrum_25.html</a> ). ....	15
<b>Figure 3.3</b> A representation of DICOM file's information. ....	16
<b>Figure 3.4</b> Data structure of a digital image. ....	18
<b>Figure 3.5</b> Original MRA image of thoracic region with its histogram. ....	19
<b>Figure 3.6</b> Histogram equalization enhanced MRA image of thoracic region with its histogram.....	20
<b>Figure 3.7</b> Histogram equalization enhanced MRA image of thoracic region with its histogram.....	21
<b>Figure 3.8</b> Example of using multiple thresholds for segmentation. Top left: original image. Top right: thresholding result after using a low threshold value to segment the image into head and background pixels. Bottom left: result after using a higher value to segment the bone pixels. Bottom right: result after using both thresholds at once [21]. ....	23
<b>Figure 3.9</b> Edge-linking paradigm. Upper image is the original edge-tracked image and the lower image is the edge-linked image. ....	27
<b>Figure 3.10</b> Example of region growing based on a grey level range. On the left, an original 256×256 image is shown, with a grey level range of 256. On the right, the result of region growing with the seed roughly at the center of the structure, allowing a grey value range of $\pm 30$ around the grey value of the seed [21]. ....	29



<b>Figure 3.11</b> Left: A 2D grid, where all pixels of an image are arranged on the grid points of the grid. Right: In volume datasets, the voxels are arranged on a 3D grid. ....	33
<b>Figure 3.12</b> Parallel (a) and divergent (b) types of 2D slice [28].....	34
<b>Figure 3.13</b> Triangulated cubes [27]. ....	36
<b>Figure 3.14</b> The ray-casting algorithm used in this thesis. For each pixel $P_i$ in the image plane a ray is cast into the data volume until a surface, determined by some function of the intensity profile, is encountered. The distance $d_i$ from the image plane to the surface is stored in a depth-buffer or Z-buffer together with the grey-level gradient computed locally at the surface point. Images are rendered from depth and gradient information in the Z-buffer [30].....	37
<b>Figure 3.15</b> Schematic illustration of NURBS of a surface. Open knot vectors and quadratic C1-continuous basis functions are used. Also depicted are C1-quadratic ( $p = 2$ ) basis functions determined by the knot vectors. The paradigm above is a clear explanation of physical, index and parameter space [32].....	40
<b>Figure 3.16</b> Basis function triangular dependency each higher order basis function is dependent on two basis function of the one lesser order. Lower-order basis functions required for the creation of $N_{5,3}(\xi)$ .42	
<b>Figure 3.17</b> NURBS elliptical Entities. (a) Curve, (b) Surface and (c) Solid.....	43
<b>Figure 4.1</b> Anatomical planes. An anatomical plane is a hypothetical plane used to transect the human body, in order to describe the location of structures. Three principal planes are used: coronal, axial, sagittal.....	45
<b>Figure 4.2</b> Histogram of prior image. The information we can extract from this histogram is that the image should be a “dark” image. We can understand that from its histogram distribution. ....	46
<b>Figure 4.3</b> Histogram equalization. The values of histogram have been equalized. The distribution of this histogram has small deviations.....	47
<b>Figure 4.4</b> Adaptive Histogram Equalization (AHE). Comparing the histogram of prior image with this histogram, we can understand that this technique transforms histogram peaks in order to make the image lighter, and to enhance the contrast. ....	47
<b>Figure 4.5</b> Gamma Correction. Comparing the histogram distribution of processed image to the prior image we can see that gamma correction smooths image’s distribution function in order to have better contrast results. ....	48
<b>Figure 4.6</b> The first stack presents input axial images, the second stack outcomes of “Canny edge detection” algorithm and the third outcomes of the seeded region growing algorithm. Images are sorted from upwards to downwards. ....	49
<b>Figure 4.7</b> Edge linking. The above figures are inputs and outputs of the proposed method. The right stack describes the inputs with the red circles denote the discontinuity of the contours and the left stack the fixed outcomes. ....	50

<b>Figure 4.8</b> GUI for selecting the first inner aortic seed point. ....	50
<b>Figure 4.9</b> The three basic algorithms we use for generating a common pipeline for arteries outer wall segmentation. ....	51
<b>Figure 4.10</b> “Seeded region growing algorithm”. The images above are shown how the seeded region growing algorithm works. Inputs exist in the right stack and outputs in the left one. The order of images is sorted from downwards to upwards.....	51
<b>Figure 4.11</b> A schematic diagram of image registration technique. $\mathbf{F}$ denotes the 3D volumetric matrix which has been generated for our processing. ....	52
<b>Figure 4.12</b> A flowchart diagram of the basic semi-automatic elliptical tracking algorithm. The only inputs that algorithm needs are the starting slice, ending slice and the first seed point for descending aorta.....	53
<b>Figure 4.13</b> This scheme shows an elliptical tracking algorithm that moves upwards for descending aorta and aortic arch and downwards for ascending aorta. Red line depicts the “aortic axe”. ....	54
<b>Figure 4.14</b> Three basic segmentation problems. ....	55
<b>Figure 4.15</b> Aortic arch’s segmentation. We label and separate the contours from the image shown in figure 21. Then the above images come out. We compare the distances from the “aortic axe” to the centroids of every contour. Finally we keep the contour with the closest distance.....	56
<b>Figure 4.16</b> First aortic arch’s slice after edge detection and the same slice after the segmentation. ....	56
<b>Figure 4.17</b> Top-left, contour with previous geometrical center. Top-right, contour with previous and geometrical center. Mid-left, line between these points. Mid-right, computing the symmetric point. Bottom, line between the three points. ....	57
<b>Figure 4.18</b> Polygonal model of a thresholding segmentation procedure and the single aorta after the proposed segmentation method of aorta. ....	58
<b>Figure 4.19</b> Polygon model of Aorta generated by iso-contouring. ....	58
<b>Figure 4.20</b> Left: initial sagittal slice. Middle: Boundaries of aorta and the computed central line. Right: Point cloud model of aorta with cross-sections perpendicular to its centerline.....	60
<b>Figure 4.21</b> Basic pipeline for 2D curves fitting. ....	61
<b>Figure 4.22</b> Primary sorting. ....	62
<b>Figure 4.23</b> Circle-like sorting. ....	62
<b>Figure 4.24</b> Convex hull algorithm. A smoother representation of previous cross-section of Figure 4.22..	63
<b>Figure 4.25</b> Linear Interpolations of aortic perimeter’s data points. ....	63

<b>Figure 4.26</b> White area depicts an aortic cross-section. Red spline is an interpolant to initial data points and blue spline is an interpolant to optimized data points of the same cross-section.....	65
<b>Figure 4.27</b> Registration of 20 points per 18 degrees angle in order to avoid over-fitting using lesser points. ....	65
<b>Figure 4.28</b> Linear interpolation to enhanced data points of model in order to create smooth horizontal and vertical cuts.....	65
<b>Figure 4.29</b> Closed cubic B-Spline curves with $11(n)$ control points interpolating $9(n+2)$ data points. ....	66
<b>Figure 4.30</b> Cubic closed and continuous B-Spline interpolating aortic data points. Blue color points represent curve's control points.....	68
<b>Figure 4.31</b> Control points of descending aorta and a zoomed in area of it. ....	68
<b>Figure 4.32</b> The requested control net of thoracic descending aorta describing by B-Splines on the left and the same on the right describing with linear segments. ....	69
<b>Figure 4.33</b> A quadrilateral NURBS surface of thoracic descending aorta. ....	70
<b>Figure 5.1</b> Capturing the diameters of descending aorta. epresent a red and a blue signal for the horizontal and vertical diameter of descending aorta, respectively. ....	72
<b>Figure 5.2</b> From left to right there is a signal representing the diameter of descending aorta and the diameter of ascending aorta, respectively. ....	73
<b>Figure 5.3</b> Signals of the descending aorta values from all the three volunteer's data that have been simulated. ....	73
<b>Figure 5.4</b> From left to right there is a signal representing the values of geometrical center of coordinate x and the values of geometrical center of coordinate y. ....	74
<b>Figure 5.5</b> This signal depicts the convergence criterion of our segmentation algorithm.....	74
<b>Figure 5.6</b> Initial axial slices. We register the edge detection results with white color in these images...	75
<b>Figure 5.7</b> Polygonal model of aorta.Three simulations of diffirent patient's data input.....	76
<b>Figure 5.8</b> Aorta with back-projected initial slices in order to compare and evaluate model. ....	77
<b>Figure 5.9</b> Single NURBS model of aorta and NURBS model back-projected with initial axial slices. ....	78
<b>Figure 5.10</b> White color the boundary of an aortic cross-section. Blue color is the generated cubic B-Spline curve.....	79

# 1 Introduction

## 1.1 Thesis area of interest

The cardiovascular system is made up of the heart and blood vessels. Cardiovascular disease (CVD) is defined as any serious, abnormal condition of the heart or blood vessels (arteries, veins). This kind of diseases includes coronary heart disease (CHD), stroke, peripheral vascular disease, congenital heart disease, endocarditis, and many other conditions. Risk factors are variables that predict who is most likely to develop CVD. Most of the risk factors for cardiovascular disease and stroke are modifiable or entirely preventable. By modifying risk factors, you decrease the chances of getting diseases. Modifiable risk factors include tobacco use, high blood pressure, physical inactivity, high blood cholesterol, obesity, heavy alcohol consumption, and poor nutrition. Non-modifiable risk factors are age and family history. The more risk factors one has, the higher the risk of developing disease. According to the European Heart Network, each year cardiovascular disease causes 3.9 million deaths in Europe, which makes them the main cause of death in western countries, responsible for hundreds of thousands of early deaths all over the world. CVD is not only a major threat to individuals' lives, but also a major economic cost to all European countries [1].

For all these reasons scientists focused on the simulation of biological structures such as the cardiovascular system in order to enhance all the current methods for detection, prediction and fighting against of CVDs. Computer modeling and simulation, particularly when augmented with graphical methods, are playing an increasingly central role in changing both the way in which medicine is taught, and the way it is practiced. Medical imaging and medical visualization have risen in the last two decades to an essential tool in clinical diagnosis and follow-up. Visualization in medical research of cardiovascular system has emerged as a unique and significant discipline aimed at developing approaches and tools to allow researchers and practitioners to “see” and comprehend the living systems they are studying. Approaches to vascular visualization include generation of realistic displays for representation of images and related information in three or more dimensions, development of interactive and automated methods for manipulation of multidimensional images and associated parametric data and design and validation of models that enhance the decision-making processes in biomedical applications. As these techniques evolve, the number of available modalities increases steadily, offering new ways to detect pathologies and new alternative techniques for treatment. The potential benefits of medical simulations in medical practice have been recognized by the American National Board of Medical Examiners (NBME) are the decreased delay in reporting test results , increased accuracy and more life-like testing situations, compared to the traditional paper-based method of testing [2].

This thesis presents a semi-automated algorithm for NURBS reconstruction of the thoracic aorta giving innovative solutions to analysis problems. A semi-automatic algorithm has been developed that combines digital image processing and computational geometry methods for segmentation and 3D NURBS surface reconstruction, using DICOM files. NURBS permit a smoother representation of curves and surfaces and the exact representation of any conic section. Parameterization by means of NURBS functions allows one to obtain a flexible and versatile modeling of surfaces. Once modeled by NURBS, the curve or surface offers flexibility in the representation of its form via the rearrangement of the control points or reassignment of the weight values. As a consequence, NURBS became the standard functions for description of curves and surfaces in Computer Science and Engineering. Last but not least there are of great need in order to take advantage of the new, enhanced isogeometric analysis. The highly demanding geometry of the thoracic aorta demands the isogeometric method as it vanishes all the main drawbacks of the finite element method [3]. Specifically, it merges the geometric design with the mesh generation into a single procedure, eliminates the geometric error, significantly increases the accuracy of the analysis, and drastically reduces the required computational time and cost [4].

## 1.2 State of the art

Modern diagnostic imaging enables acquisition of angiography data such as magnetic resonance angiography (MRA), computed tomography angiography (CTA), and digital subtraction angiography (DSA) and storing the acquired information in DICOM datasets. Available software packages offer basic methods for volume rendering of DICOM files, they do not however provide automatic, easy methods for creating and exporting reusable 3D model of a specific segmented part, which is the first step for many further computer graphics and simulation applications. In current literature great work has been done by Grayhart et al in this work [5] in the automation of segmentation of aorta. An accurate 3D reconstruction of arteries can provide models that allow comprehensive visualization of the vessel geometry and assessment of the characterization of different abnormalities such as aortic calcification, atherosclerosis and aortic aneurysm. The scope of studying and analyzing these diseases is of great value because this gives to us great advantages in the process of prediction. A model of atheromatous plaque and its composition has been studied by Fotiadis et al in order to examine a stenting method in arteries with plaque [6]. Processing vessels geometry and producing three dimensional meshed, can permit blood flow simulation and evaluation of the role of the local hemodynamic forces on any vessel's abnormality with finite element analysis or isogeometric analysis. A novel work has been done by Hughes et al describing an automatic

method for reconstructing aorta with NURBS geometries for isogeometric analysis of the blood flow [7].

Various treatment options are nowadays available for many CVD, such as the percutaneous minimally-invasive techniques such as stenting, grafting and angioplasty procedure, which have to be guided by the numerical results of the respective simulations according to computational methods. Patient cardiovascular modeling is a recently developed and even-growing medical imaging field which focuses on detecting Abnormalities in cardiovascular system and characterizing them as we see in [6], [7], [10]. Furthermore, the quantitative measurements of human cardiovascular parts are crucial for the diagnosis of cardiovascular diseases, such as aortic calcification, atherosclerosis and aortic aneurysm. Thus, medical visualization and geometry editing is a notable task for techniques, which focus on aiding medical treatments, such as 3D printing, rapid prototyping in medicine and virtual surgery.

## 1.3 Thesis contribution

As we see from the relative bibliography the issue of automatic or semi-automatic reconstruction of thoracic aorta has been studied in the past years. At first, medical visualization was developed for visualization purposes. As a consequence the geometry that has been used was of polygonal form with unstructured grid. Unstructured geometry grid intended only for visualization because of its great geometry description of an object and its low computational complexity as a model. Nowadays biomechanical applications have been developed enough to simulate such complicated systems as cardiovascular and they demand visual objects with such a structured grid that can be processed by the analysis methods. Even though the available software packages offer basic methods for volume rendering, they do not provide automated creation of the respective 3D NURBS model. Thorough investigation proved that medical industry doesn't take advantage of NURBS, so far.

This thesis' contribution is the combination of novel methods both of semi-automatic segmentation and reconstruction of aorta and generation of NURBS geometry and its requested control mesh. As we stated before there are not such automatic ways in the medical practice to model and simulate the behavior of single portions of human's body.

Our proposed algorithm has two main stages. The first one is the segmentation of aortic cross-sections from consecutive axial images and the computation of the centerline, the point cloud and the polygon mesh from the aortic contours. The second one is the NURBS fitting, which take advantage of the centerline and the point cloud to form B-Spline curves perpendicular to the centerline and define the NURBS surfaces according to lofting techniques [11]. Crucial points of our pipeline are the matching of centerline based both on axial and

sagittal slices in order to be precise. Moreover in the curves fitting stage we developed a basic method which is focused on decreasing the complexity model and increasing the accuracy of our final NURBS model. At first we translate image's information (pixels) into points in 3D space, subsequently we applied a circular spline regression to the data points and choose and register specific amount of uniformly distributed points of the generated spline. Finally these points are the new optimized point cloud of our model. This technique gives us lesser and more appropriate points describing each aortic cross-section, furthermore, the harmonization of points in a polar coordinate system and more specifically at each specific axial angle give us a smoother surface, voiding overfitting problems.

## 1.4 Composites steps of modeling and thesis outline

Regardless the specific imaging modality, medical images contain information in the form of multidimensional arrays of numeric values, which are correlated with one or more physical properties of the imaged anatomical structures. Modeling the geometry of a vessel from medical images consists in extracting the location of the vessel wall from the values contained in the image volume. Since the image formation process in general alters the underlying anatomical information, and since the resolution with which the latter is acquired is bound to be finite, medical images are coarse representations of the underlying anatomy. In our proposed method we used DICOM 3D images in order to simulate the vascular structures. Vascular modeling pipeline that we follow consists of 5 steps:

- (1) Preprocessing. In scanned computed tomography (CT) or magnetic resonance imaging (MRI) data, the intensity contrast may not be clear enough, noise exists, and sometimes the blood vessel boundary is blurred. Therefore, we use image processing techniques to improve the quality of CT/MRI data, such as contrast enhancement, filtering. Moreover in this stage we used image registration techniques in order to compute a 3D volumetric dataset from the multiple slices of MRI/ CT images.
- (2) Segmentation and 3D-Reconstruction. In this stage we proposed a well-defined algorithm which process all the axial slices of the 3D dataset. The only input that algorithm needs is one inner point of aortic cross-section. Then our procedure is capable to segment all the cross-sections across the thoracic aorta. For this stage we relied on the anatomy of thoracic aorta and composed a procedure that is based on the curvature of aortic arch. Then we applied an isosurface algorithm to the 3D segmented dataset in order to compute the surface from the aortic contours. Vascular surface models can be constructed from the preprocessed imaging data via isocontouring.

- (3) Path extraction. The goal is to find arterial paths. In the Segmentation stage we have computed a closed elliptical perimeter for each aortic cross-section, as well the approximate center of each perimeter. Storing all these centers and producing an interpolant B-Spline curve from to these, we can compute the centerline of the artery.
- (4) B-spline planar fitting. At first we have registered the specific planar cross-sections perpendicular to the centerline of aorta. After this step our main target is to compute a cubic B-Spline curve to the data points of these cross-sections.
- (5) NURBS surface generation– – a skeleton-based lofting method is developed to construct quad NURBS control meshes by sweeping the vessel’s cross-sections (cubic B-Splines) across the arterial path.

In chapter 2 we state the anatomy and physiology of the aorta. We discuss the characteristics of the organ and its normal behavior. We also present the corresponding size of diameters of organ in some specific location, which numbers will aid to evaluate the accuracy of our simulations. In chapter 3 we present all the theoretical background needed for this thesis as we split it into meaningful subchapters. We give an overview on medical imaging modalities and their specialities. Moreover we introduce the importance of DICOM datasets as the international standard for medical images and related information. Continuing with the theoretical background we present some basic methods on digital image processing and 3D reconstruction techniques. Last, but no least, we provide basic information about NURBS theory and applications. In chapter 4 we describe our proposed automated algorithm, presenting our novel methods in the two basic subchapter that we present. First of all medical image processing techniques are combined in order our automatic algorithm to come of. Then the second stage of our work is the NURBS fitting techniques which are presented in the second subchapter. Interesting points of our approach is the matching of extracted vessel’s centerline matching from axial and sagittal slices, as well the compression of point cloud with an optimized manner giving us low complexity for the final model and avoiding overfitting with lesser points. In other words we construct the control net of our model with a novel and enhanced way which is very important step for further simulation applications in biomechanics. In chapter 5 we evaluate the results of our method with some pixel-based binary test metrics. We compute the accuracy sensitivity and precision of our total enclosed pixels by the NURBS surface and compare them with the amount of the total pixels which are enclosed in early stage segmentation. With this metrics we can compare our processing in two different stages and examine the divergence. Last but not least, chapter 6 acts as an epilogue for this thesis, presenting our conclusions along with future improvements.

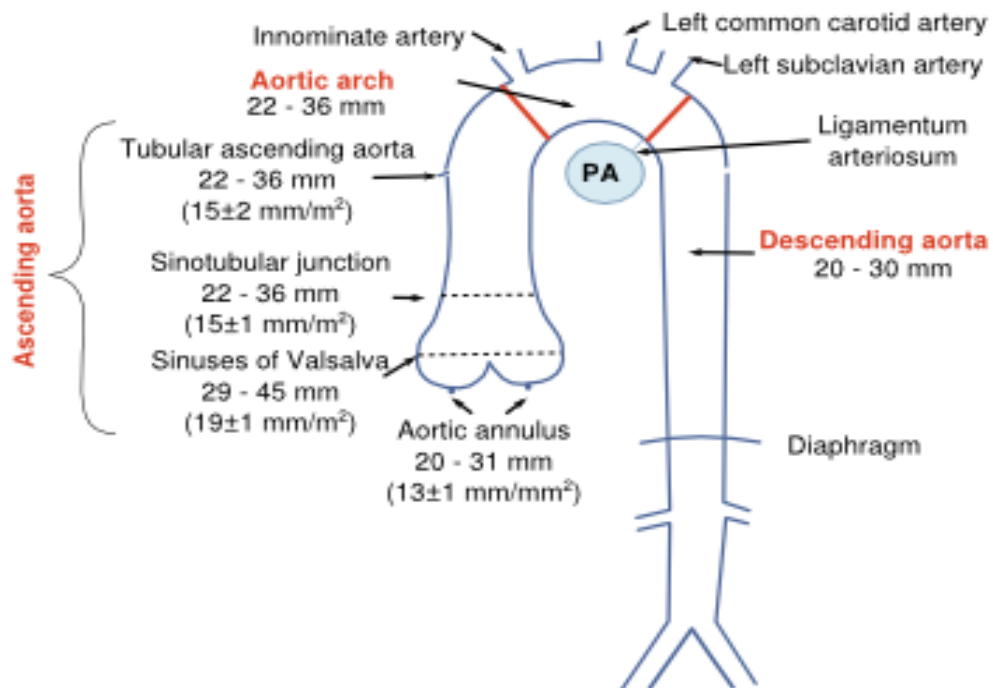


## 2 Anatomy – Physiology of aorta

### 2.1 Anatomy of the aorta

The aorta is the main artery of the human body, often called the greatest artery, normally reaching 70 cm in length and 3.5 cm in diameter. It receives the cardiac output from the left ventricle and supplies the body with oxygenated blood via the systemic circulation. So, it arises from the left ventricle of the heart and extending down to the abdomen, where it splits into the two iliac arteries. The aorta is divided into six segments:

- Aortic root
- Sinotubular junction
- Ascending aorta
- Aortic arch
- Descending thoracic aorta
- Abdominal aorta



**Figure 2.1** Normal size of thoracic aortic segments and characterization of them. [12]

### **Aortic root**

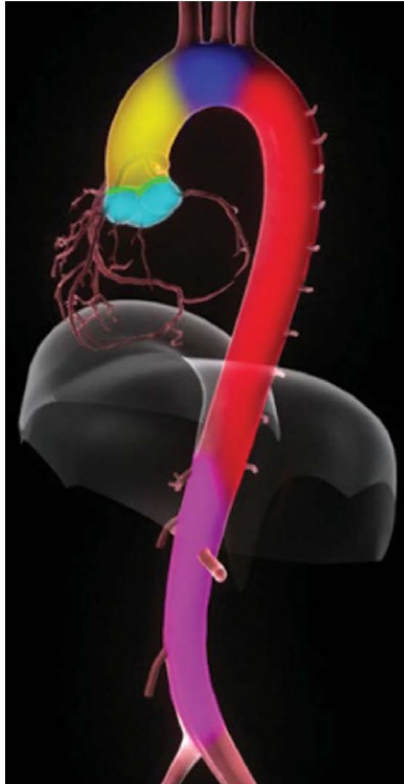
The aortic root begins at the aortic valve annulus level and extends to the sinotubular junction. It is the section of the aorta closest to and attached to the heart as shown in Figure 2.2. The aortic root consists of the aortic valve, the openings for the coronary arteries and the three aortic valve leaflets (Sinuses of Valsalva) surrounded by a fibrous ring (the annulus). Normal diameters of this segment are between 29 and 45 mm, with a variation depending on body size. The most common consequence of disease of the aortic root is impairment of the valve. In this case, dilation of the root reduces coaptation and an event like dissection reduces the suspension of the leaflets and lead to collapse of the aortic valve.

### **Sinotubular junction**

The sinotubular junction is normally a well-defined region at which the rounded and wider sinuses of Valsalva join the narrower tubular shaped ascending aorta, see Figure 2.1. Normally, the sinotubular junction has the same dimension as the aortic annulus and constitutes a critical support to the superior part of the aortic valve commissures. Normal diameter of the sinotubular junction is 22 up to 36mm, with variation depending on body size. Effacement (marked dilation) of this junction suggests annuloaortic ectasia and often is seen in patients with Marfan syndrome. Dilation of the sinotubular junction confounds the correct spatial arrangement of the commissures and typically results in central maladaptation and aortic insufficiency. Replacement of an aneurysmal ascending aorta and sinotubular junction with a tube graft of correct diameter (stent) often corrects aortic insufficiency by reestablishing the correct spatial suspension of the aortic valve. Similarly, reestablishing integrity of a dissected sinotubular junction by sewing the dissected root components to a tube graft that supplies support to the repaired sinotubular junction may also correct aortic insufficiency from an aortic dissection.

### **Ascending aorta**

The ascending aorta is the segment of the aorta from the sinotubular junction to the first great arch vessel, the brachiocephalic artery (innominate artery), see Figure 2.2. Identifying the distinction of the ascending aorta from the arch on the outside region is far easier than it is on the inside part of the aorta. The ascending aorta is often affected by dissections and aneurysms, it is intrapericardial, as the pericardium reaches distally the first supra-aortic vessel, see Figure 2.1. The ascending aorta measures approximately 5cm in length and up to  $30\pm 6$ mm in diameter, with variation appropriate for body size.



**Figure 2.2** Segmental division of the aorta: the aortic root (LIGHT BLUE), the sinotubular junction (GREEN), the ascending aorta (YELLOW), the aortic arch (DARK BLUE), the isthmus and descending (thoracic) aorta (RED), and the abdominal aorta (PINK).[13]

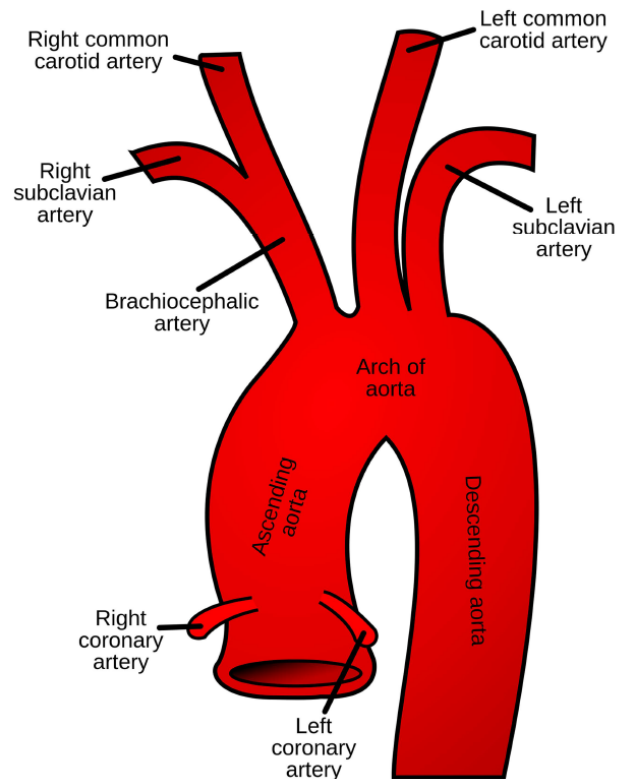
### **Aortic arch**

The aortic arch is the portion of the aorta from the first supra-aortic vessel (i.e., the brachiocephalic artery) to the left subclavian artery. See Figure 2.2 where the aortic arch is highlighted in a dark blue color. The aortic arch is mostly extrapericardial; normal diameter of the aortic arch is  $30 \pm 6\text{mm}$ , with variation appropriate for body size. The arch arcs over the right pulmonary artery. Branch vessels include right brachiocephalic artery (right subclavian artery and right common carotid artery), left carotid artery, and left subclavian artery as shown in Figure 2.3. Aortic arch branch variations are common.

### **Descending aorta**

The isthmus is the narrower portion of the aorta (by approximately 3mm) between the left subclavian artery and ligamentum arteriosus, a remnant of the ductus arteriosus. Blunt traumatic deceleration injury, resulting in transection to the aorta often occurs at this site. The descending thoracic aorta begins at the ligamentum arteriosus and continues to the level of the

diaphragm. In Figure 2.2 the descending aorta is highlighted in red color. The esophagus runs alongside (within 0.5cm) the descending aorta. Normal diameter of the descending proximal aorta is 30mm. At the eleventh rib level, it is inferior to 23 mm. Branch vessels of the isthmus include the ductus arteriosus. Branch vessels of the descending aorta include intercostal arteries, spinal arteries, and bronchial arteries.



**Figure 2.3** The thoracic aorta. In this picture are emphasized the three supra-aortic branches - i.e., the brachiocephalic trunk, the left common carotid artery, and left subclavian artery. The figure shows also the coronary arteries (left and right coronary arteries). Those arteries are responsible to carry the oxygenated blood to the heart.

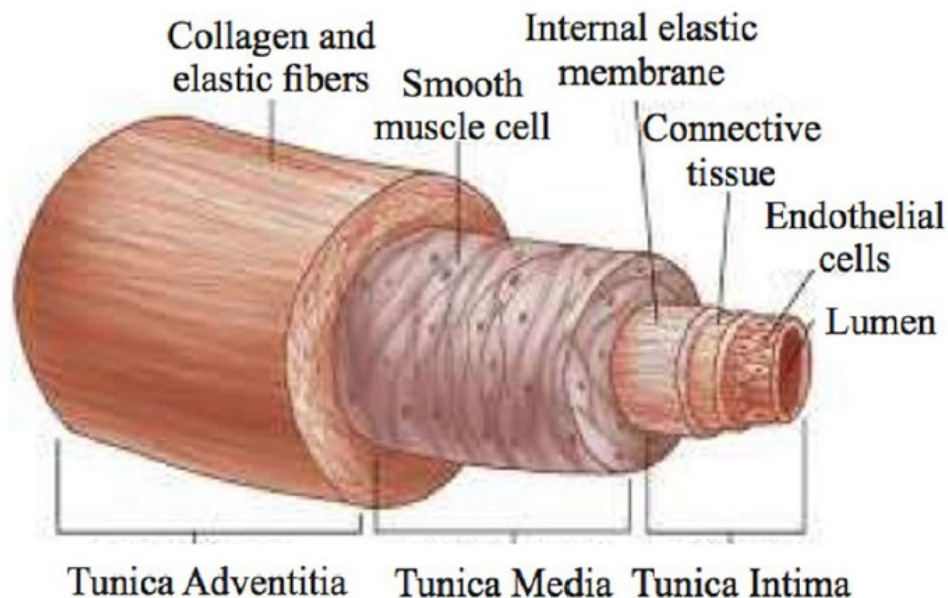
### **Abdominal aorta**

The abdominal aorta starts at the hiatus of the diaphragm and courses retroperitoneal to its bifurcation, see Figure 2.2 pink color highlight the abdominal aorta. Major branch vessels include inferior phrenic arteries, celiac artery branches, renal arteries, superior mesenteric artery, inferior mesenteric artery, lumbar and spinal arteries, and iliac arteries. Normal diameter of the suprarenal abdominal aorta is 20mm, and normal diameter of the infrarenal abdominal aorta is inferior to 20mm. From a practical point of view the abdominal aorta is reasonably equal to the size of the patient's thumb.

## 2.2 Histology of the aorta

### The intima

Tunica intima is the layer at contact with the blood and it is made of a single layer of endothelial cells embedded in extracellular matrix and an underlying thin basal lamina, which is also referred to as basement membrane, providing structural support to the arterial wall. Such a layer provides a non-thrombogenic surface so that the blood can flow through the artery without forming thrombus. The basal lamina contains non-fibrillar collagen types, adhesion molecules laminin, fibronectin, and other extracellular matrix molecules. It has been observed that the orientation of the collagen fibers in the subendothelial layer through the thickness it is not uniform but dispersed. The intima is very thin, i.e., from 0.05 to 0.1mm, and its contribution to the mechanical properties in healthy young human arteries is not significant. Whereas, ageing and in the case of pathological conditions, the intima becomes thicker (from 0.2 to 0.4mm) and stiffer, and develops a more complex and heterogeneous structure. These pathological changes are associated with alterations in the mechanical properties, which differ significantly from those of healthy arteries. Finally, it is separated from the media by the internal elastic lamina, which is often considered to be part of it.



**Figure 2.4** Structural organization and composition of the three different layers in the coronary vessel wall [13].

### **The media**

Tunica media is the middle and thickest layer of the artery, i.e, from 0.1 to 0.5mm, made of smooth muscle cells, elastin and collagen immersed into an aqueous ground substance containing proteoglycans, known as matrix. Such fibers are arranged in repetitive lamellar units separated by thin fenestrated sheets of elastin, forming concentric medial layers. The thickness of such units is nearly independent of the radial location across the wall, but their number decrease as the distance from the heart increases, so that the lamellar units found absent in small muscular arteries. The laminated structure confers high strength to the media and explains how such a layer determines the mechanical properties of the whole vessel wall. In the media, collagen fibers are aligned along the circumferential direction with a very little dispersion. This structural arrangement gives the media the ability to carry loads in the circumferential direction. Apparently, the distribution of collagen does not show changes in atherosclerotic arteries. The media is separated from the adventitia by the external elastic lamina.

### **The adventitia**

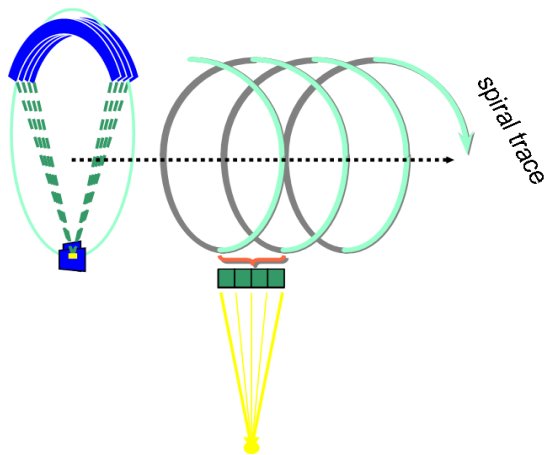
Tunica adventitia is the outermost layer of the artery and consists primarily of collagen fibers, elastin, nerves, fibroblasts, fibrocytes. The thickness size of the adventitia is thinner than the media, i.e., from 0.25 to 0.40mm. The adventitia is surrounded continuously by loose connective tissue, which often provides additional structural support. As for the intima, histological evidence has proven the dispersion of collagen fibers in the adventitial layer, which remains also in atherosclerotic arteries. Collagen fibers tend to maintain an axial orientation and remain slack at low pressures, but as the pressure increases they straightened, reinforcing the arterial wall and preventing the over-stretching and the rupture of the artery. The cardiovascular diseases are being addressed, as appropriate, through the prevention of risk factors, drug therapy and finally surgery.

## 3 Analytic background

### 3.1 Medical imaging modalities

#### 3.1.1 Computed tomography angiography

Computed Tomography Angiography has evolved from the traditional X-rays computed tomography. Its name comes from Greek words tomos - 'slice' and graphien - 'to write or record', because the result of CT imaging are (axial) slices from the object. This is achieved by rotating the X-rays source and detectors around the object, recording the individual cross-attenuations. The slice images are then reconstructed from raw X-rays attenuation measurements by so called tomographic reconstruction, mathematically based on Radon's transformation [14], [15]. With the development of fast CT scanners, the scan times have decreased substantially, which has allowed to scan a particular anatomic region of the body while contrast medium is injected intravenously at the same time. Figure 3.1 depicts the functionality principles of CT. The resulting bright vascular opacification combined with high-spatial resolution CT acquisition gave rise to so called CT-Angiography, or CTA. Modern CTA has evolved within the last years into an accurate, robust, and cost effective non-invasive imaging technique in patients with coronary or arterial diseases, thanks to the advent of helical multi-slice CT scanners which allow fast and precise 3D medical imaging of a human body. Current state-of-the-art 16- and 64-channel CT scanners allow acquisition of a set of transverse images representing the whole area of interest in less than 30 seconds. With a simultaneous use of contrast medium, the produced images are well-suited for angiographic purposes and due to the better contrast resolution of CT when compared to conventional angiography, contrast medium needs not be injected directly into the arterial system, but only into an arm vein, intravenously. CTA is thus much less invasive and less harmful for the patient. A CTA dataset of the lower extremity arterial tree (peripheral CTA) consists of approximately 1200–2000 images  $512 \times 512$  pixels each, with 4096 levels of grey and resolution below  $1 \text{ mm}^3$  is produced. Within such dataset, vessels manifest densities that are different to those of the surrounding tissue, due to the contrast-medium enhancement of blood. As the number of transverse slices is very high, the radiologic interpretation of CTA dataset is laborious and lengthy and dedicated post-processing tools that generate a small set of easily interpretable images are typically used.



**Figure 3.1** Spiral CT scanning principle. The X-rays source and oppositely mounted detectors in rows perform two motions: 1. rotation around the axis, 2. longitudinal movement along the axis. In this way, a large anatomic volume can be scanned rapidly [16].

### 3.1.2 Magnetic resonance angiography

Magnetic resonance imaging (MRI) is a convenient medical imaging technique for visualization of soft tissues and functional imaging, for example of blood flow. Magnetic resonance use for imaging is within a frequency range between 10 and 200 MHz, i.e. in the radio frequency (RF) domain. The RF waves used for MRI have a wavelength between one and 100 meters. For comparison, diagnostic x-rays have wavelengths between  $10^{-12}$  and  $10^{-9}$  m and ultrasound pulses for imaging result in a wavelength of  $10^{-4}$  m in blood. A general electromagnetic spectrum is shown in Figure 3.2. The primary source of the MRI signal used in medical imaging originates from protons. Most of the protons in the human body are present in the form of water since the human body is approximately 70% water. The protons are magnetic and they rotate around their own axis, a motion referred to as spin. Because the protons are magnetic they act as tiny compass needles with the axis of rotation being the direction of magnetization. The scanner magnet produces a strong magnetic field in the longitudinal direction which aligns the spins according to the applied field. Due to the huge number of protons the effect of the spin alignment becomes measurable.

Magnetic resonance angiography is a group of techniques based on magnetic resonance imaging (MRI) to image blood vessels. Magnetic resonance angiography is used to generate images of arteries (and less commonly veins) in order to evaluate them for stenosis (abnormal narrowing), occlusions, aneurysms (vessel wall dilatations, at risk of rupture) or other abnormalities. MRA is often used to evaluate the arteries of the neck and brain, the thoracic and abdominal aorta, the renal arteries, and the iliac arteries. There are special MRA



techniques that maximize vascular contrast by enhancing signal from flowing blood or suppressing signal from stationary tissues. MRA techniques include:

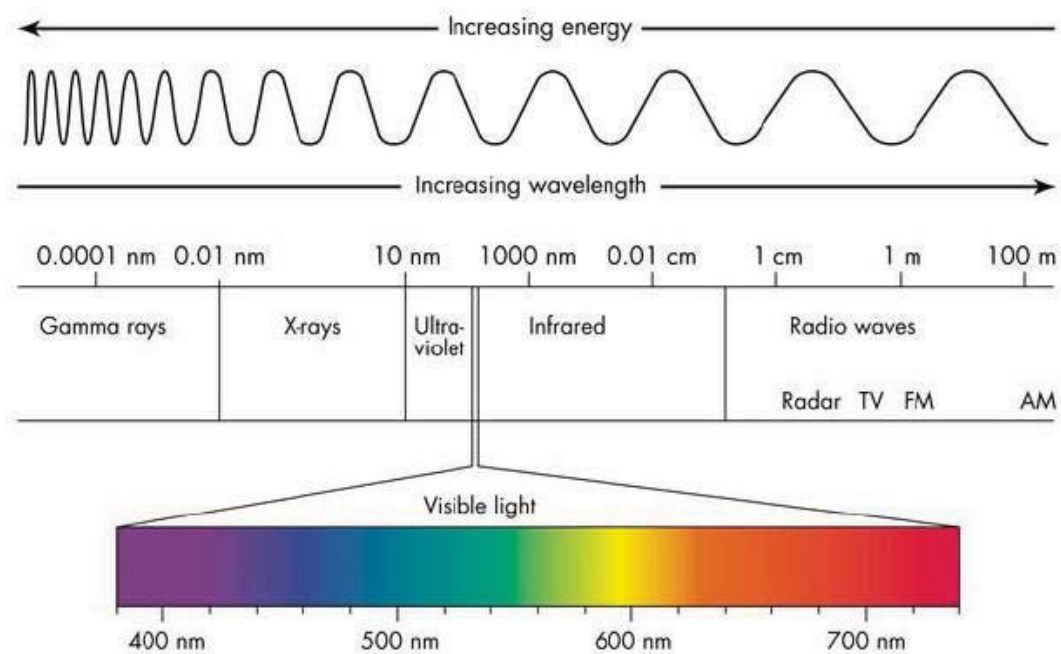
### **Time of flight MRA (TOF-MRA)**

With this technique, signal depends on flow-related enhancement providing information about flowing blood perpendicular to the slice plane. It applies an incoherent gradient echo pulse sequence and gradient moment rephrasing to saturate stationary spins and fully magnetize inflow spins, producing a high vascular signal with proper flip angles and TR values. TOF-MRA can be acquired with 2D or 3D acquisitions. 3D acquisitions are optimal for obtaining high-resolution images and for evaluating smaller vessels with high-velocity blood flow (intracranial vessels). The advantages of TOF-MRA include sensitivity to slow flow, reduced sensitivity to intravoxel dephasing, and reasonable scan time. Its disadvantages include sensitivity to T1 effects and saturation of in-plane flow and enhancement are limited to flow entering the field of view or very high-velocity flow. 2D-TOF offers a large area of coverage yet has lower resolution than the resolution of 3D inflow studies. 2D-TOF also shows saturation of in-plane flow, and patient motion can cause data mis-registration between the individually acquired slices last but not least vessel borders may appear serrated. 3D-TOF offers high resolution for small vessels and is more tolerant of patient motion, demonstrating high SNR where vessels appear less serrated on the reformatted image. However, 3D-TOF also shows saturation of in-plane flow and has a small area of coverage.

### **Phase-contrast MRA**

Phase-contrast MRA is a sequence related to in-phase change of flowing blood. This variation relies on blood velocity, flow direction, and the type of scan used. It provides information about vascular anatomy, flow velocity, and flow direction. Blood flow velocity differs based on physiologic status, type and size of vessel, and pathology within the vessel. An oscillator in the scanner electronics is used as a reference to detect phase shift and is also used for position encoding in MRI. Phase has a 360° range, beyond which the same values repeat, causing aliasing. The PC sequence has high sensitivity to fluid motion, so it can be used in clinical blood flow analysis. It can also be used to evaluate blood flow with multiple directions and multiple velocities. This technique can create two types of images: magnitude and phase images. In magnitude images, vessels show high signal and a suppressed background because of short TR, allowing incomplete recovery of magnetization. In phase images, the signal is linearly proportional to the velocity of the blood. The direction of flow appears bright when flow is in the same direction as the velocity encoding gradient, whereas flow in the opposite direction appears black. This technique can therefore be used to localize vessels that can then be more thoroughly evaluated with other techniques. 3D techniques offer an SNR and spatial

resolution better than those of 2D techniques and also offer the ability of reformatting. 3D PC-MRA images are acquired for smaller vessels, when multidirectional vascular information (flow velocity and flow direction) is required. 2D PC-MRA techniques provide information about direction and velocity of flow in addition to information about multidirectional flow in a reasonable scan time. Precise lumen area delineation demands high in-plane resolution, which is necessary to accurately estimate the mean vessel velocity. The signal level of adjacent tissues controls the partial volume effects at vessel edges, and thicker slices will misalign; therefore, thin slices and high spatial resolution are important to overcome these effects.



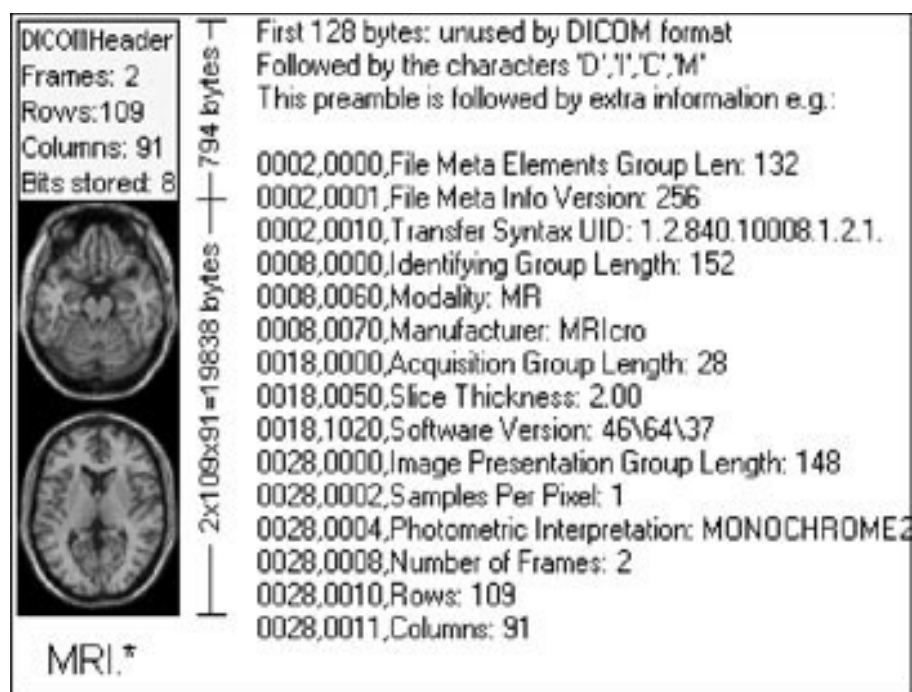
**Figure 3.2** Electromagnetic spectrum ([https://www.miniphysics.com/electromagnetic-spectrum\\_25.html](https://www.miniphysics.com/electromagnetic-spectrum_25.html)).

## 3.2 DICOM

Digital Imaging and Communications in Medicine (DICOM) is the international standard for medical images and related information. It defines the formats for medical images that can be exchanged with the data and quality necessary for clinical use. DICOM was originally developed by the National Electrical Manufacturers Association (NEMA) and the American College of Radiology for computerized axial tomography (CAT) and magnetic resonance imaging (MRI) scan images and it is now controlled by the DICOM Standards Committee. It is implemented in

almost every radiology, cardiology imaging, and radiotherapy device (X-ray, CT, MRI, ultrasound, etc.). Since its first publication in 1993, DICOM has revolutionized the practice of radiology, allowing the replacement of X-ray film with a fully digital workflow. Much as the Internet has become the platform for new consumer information applications, DICOM has enabled advanced medical imaging applications that have “changed the face of clinical medicine”. From the emergency department, to cardiac stress testing, to breast cancer detection, DICOM is the standard that makes medical imaging work — for doctors and for patients. DICOM files consists of a number of attributes that contain, as shown in Figure 3.3:

- patient related information (name, sex, age, etc.),
- modality and imaging procedure information (type of machinery, acquisition parameters, radiation dose, contrast media, etc.) and
- image related information (image dimension, in-plane resolution or pixel size, slice thickness, slice distance).



**Figure 3.3** A representation of DICOM file's information.

## 3.3 Image processing and techniques

### 3.3.1 Introduction

In physical world, any quantity measurable through time over space or any higher dimension can be taken as a signal. A signal is a mathematical function, and it conveys some information. A signal can be one dimensional or two dimensional or higher dimensional signal. One dimensional signal is a signal that is measured over time. The common example is a voice signal. The two dimensional signals are those that are measured over some other physical quantities. The example of a two dimensional signal is a digital image. Signal processing is a discipline in electrical engineering and in mathematics that deals with analysis and processing of analog and digital signals, and deals with storing, filtering, and other operations on signals. These signals include transmission signals, sound or voice signals, image signals, and other signals. Out of all these signals, the field that deals with the type of signals for which the input is an image and the output is also an image is done in image processing. Digital image  $a[m, n]$  described in a 2-D discrete space is derived from an analog image  $a(x, y)$  in a 2D continuous space through a sampling process that is frequently referred to as digitization. But actually, this image is nothing but a two dimensional array of numbers ranging between 0 and 255. The 2D continuous image  $a(x, y)$  is divided into  $N$  rows and  $M$  columns. The intersection of a row and a column is termed as a pixel. Figure 3.4 shows the data structure of a digital image. It has been divided into  $N$  rows and  $M$  columns. The value assigned to every pixel ( $a(i, j)$ ) is the average brightness in the pixel rounded to the nearest integer value. The process of representing the amplitude of the 2D signal at a given coordinate as an integer value with  $L$  different gray levels is usually referred to as amplitude quantization or simply *quantization*.

- Binary Image. This image has only two luminosity values, black which is zero value (0) and white which is one value (1). This image is also the simplest form of a digital image.
- Gray-scale Image. A gray-scale image with  $N \times M$  dimensions there is nothing more than a 2-D matrix.

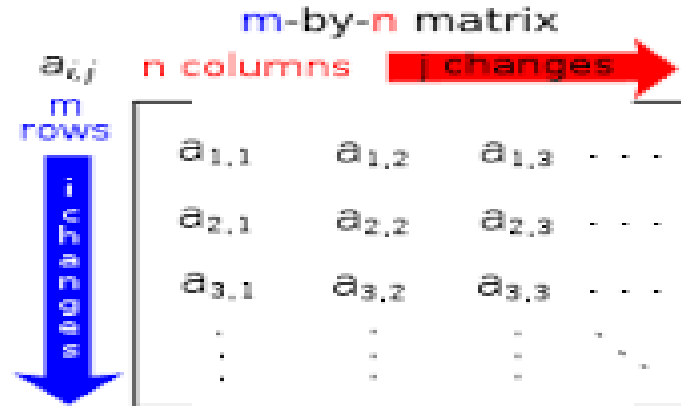
$$I(i, j), i = 1, \dots, N \text{ and } j = 1, \dots, M \text{ where } 0 \leq I(i, j) \leq G - 1 \text{ with } G = 2^m$$

The image's value  $I(i, j)$  is proportionate with corresponding pixel's luminosity value. Commonly variable  $m=8$  and so  $G=256$ . So the commonly range value of a gray-scale image is  $[0 \ 255]$  with 0 correspond to black and 255 to white color.

- Colored Image. This type of image use real colors and it is the most common for the human's vision. It consists of three gray-scale images of  $N \times M$  dimensions.

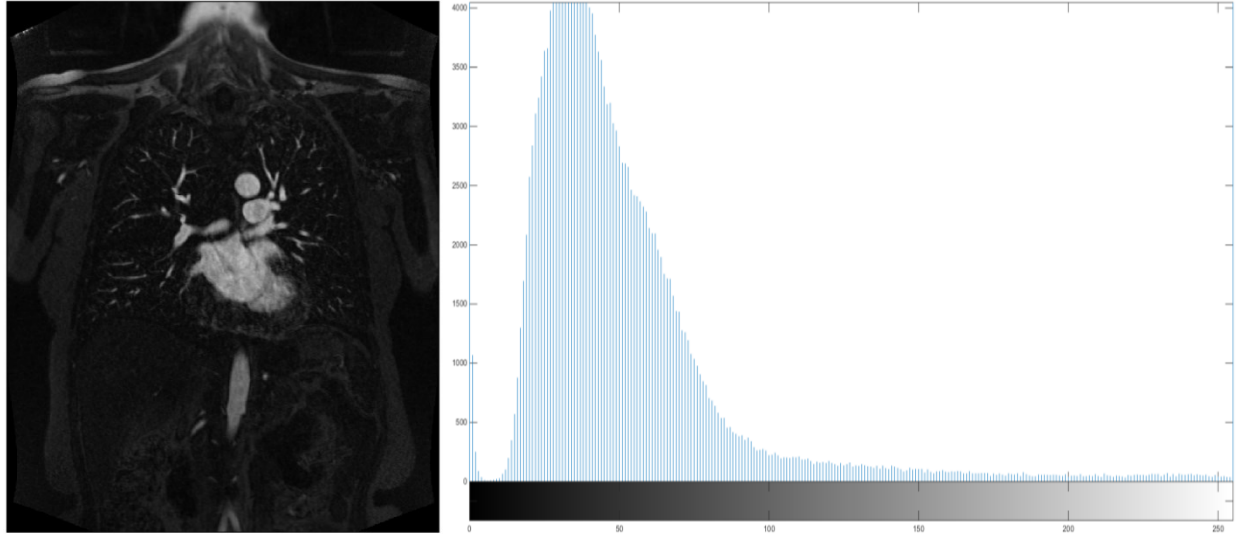
$$I(i, j) = [I_1(i, j), I_2(i, j), I_3(i, j)]$$

$$\text{Where } 0 \leq I_c(i, j) \leq G - 1 \text{ for each } c = 1, 2, 3$$



**Figure 3.4** Data structure of a digital image.

In an image processing context, the histogram of an image normally refers to a histogram of the pixel intensity values. The histogram plots the number of pixels in the image (vertical axis) with a particular brightness value (horizontal axis). For an 8-bit gray-scale image there are 256 different possible intensities, and so the histogram will graphically display 256 numbers showing the distribution of pixels amongst those gray-scale values. In the field of computer vision, image histograms can be useful tools for thresholding. Because the information contained in the graph is a representation of pixel distribution as a function of intensity value variation, image histograms can be analyzed for peaks and/or valleys. This threshold value can then be used for edge detection, image segmentation, and co-occurrence matrices. In the Figure 3.5 there is a medical image taken from a CT that depicts a coronal thoracic slice and its histogram.



**Figure 3.5** Original MRA image of thoracic region with its histogram.

### 3.3.2 Image enhancement

Image filtering allows you to apply various effects on photos. Filtering is a technique for modifying or enhancing an image. For example, you can filter an image to emphasize certain features or remove other features. Image processing operations implemented with filtering include smoothing, sharpening, and edge enhancement.

#### Histogram equalization

Histogram equalization is a technique for adjusting image intensities to enhance contrast. Let  $f$  be a given image represented as a  $m_r$  by  $m_c$  matrix of integer pixel intensities ranging from 0 to 1.  $L$  is the number of possible intensity values, often 256. Let  $p$  denote the normalized histogram of  $f$  with a bin for each possible intensity. So let  $p$  denote the normalized histogram of  $f(m_r, m_c)$ .

$$p_n = \frac{\text{num of pixels with intensity } n}{\text{total number of pixels}} \text{ with } n = 1, 2, \dots, L - 1$$

The histogram equalized image  $g$  will be defined by,

$$g_{i,j} = \text{floor}((L - 1) \sum_{n=0}^{f_{i,j}} p_n)$$

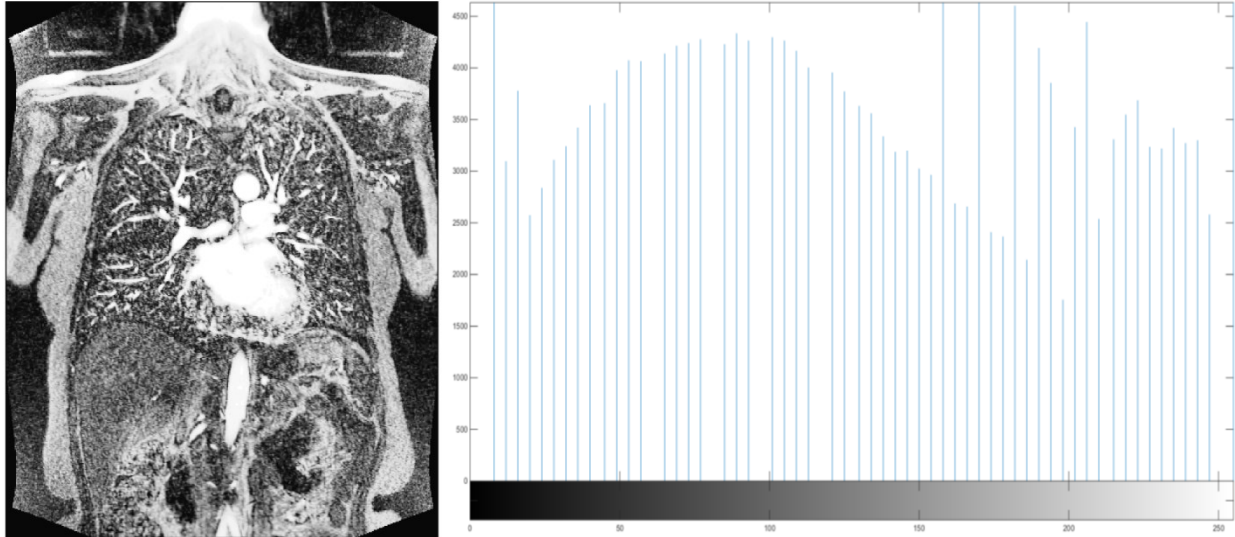
where  $\text{floor}()$  rounds down to the nearest integer. This is equivalent to transforming pixel intensities,  $k$ , of  $f$  by the function

$$T(k) = \text{floor}((L - 1) \sum_{n=0}^k p_n)$$

The motivation for this transformation comes from thinking of the intensities of  $f$  and  $g$  as continuous random variables  $X, Y$  on  $[0, L - 1]$  with  $Y$  defined by

$$Y = T(X) = (L - 1) \int_0^X p_x(x) dx$$

where  $p_x$  is the probability density function of  $f$ .  $T$  is the cumulative distributive function of  $X$  multiplied by  $L - 1$  [17]. In Figure 3.6 we see the histogram equalized prior thoracic slice.

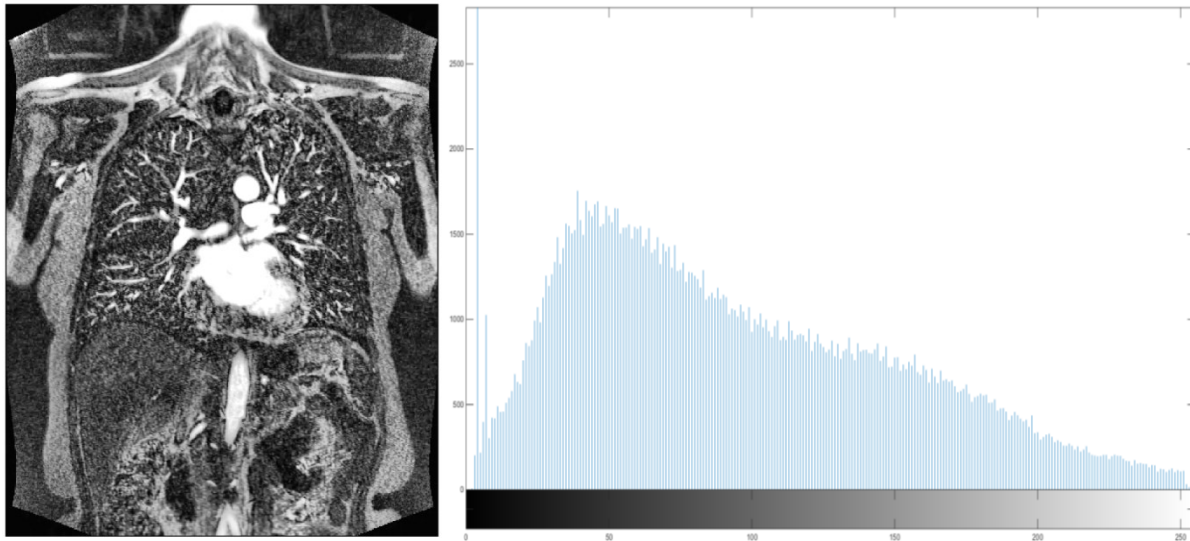


**Figure 3.6** Histogram equalization enhanced MRA image of thoracic region with its histogram.

### **Adaptive Histogram Equalization**

For images which contain local regions of low contrast bright or dark regions, global histogram equalization won't work effectively. Ordinary histogram equalization uses the same transformation derived from the image histogram to transform all pixels. However, when the image contains regions that are significantly lighter or darker than most of the image, the contrast in those regions will not be sufficiently enhanced. A modification of histogram equalization called the Adaptive Histogram Equalization can be used on such images for better results. Adaptive histogram equalization works by considering only small regions and based on

their local CDF (cumulative distribution function), performs contrast enhancement of those regions. It transforms each pixel with a transformation function derived from a neighborhood region. In its simplest form, each pixel is transformed based on the histogram of a square surrounding the pixel, as in the figure below. The derivation of the transformation functions from the histograms is exactly the same as for ordinary histogram equalization: The transformation function is proportional to the cumulative distribution function (CDF) of pixel values in the neighborhood [18]. This method's paradigm is shown in Figure 3.7.



**Figure 3.7** Histogram equalization enhanced MRA image of thoracic region with its histogram.

### **Gamma correction and linear mapping**

Gamma, represented by the letter  $\gamma$ , can be described as the relationship between an input and the resulting output. The relationship in this case between the input and output is that the output is proportional to the input raised to the power of gamma. The formula for calculating the resulting output is as follows:

$$I' = 255 \times \left( \frac{I}{255} \right)^\gamma$$

Note that with a gamma of 1 the input equals the output producing a straight line. For calculating gamma correction the input value is raised to the power of the inverse of gamma. The formula for this is as follows:

$$I' = 255 \times \left( \frac{I}{255} \right)^{1/\gamma}$$



### 3.3.3 Image segmentation

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). Image segmentation is a classification problem and therefore is one of the most important and difficult tasks in image processing. The goal of segmentation is the division of an image into meaningful structures which is often an essential step in image analysis, object representation, visualization, and many other image processing tasks. Image segmentation is typically used to locate objects and boundaries in images [19]. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics. In many cases image segmentation remains an unsolved problem. The simplest form of segmentation is binarization. Binarization is the process of segmenting an image into 2 classes, such as foreground and background, thus distinguishing the object(s) of interest from the rest of the image. The binary image is referred to as a *mask* when it is used to exclude pixel data from calculations.

A great variety of segmentation methods has been proposed in the past decades, and some categorization is necessary to present the methods properly here. The categorization presented in this chapter is therefore rather a categorization regarding the emphasis of an approach than a strict division [20].

- Threshold based segmentation.
- Edge based segmentation.
- Region based segmentation.

#### 3.3.3.1 Threshold based segmentation

Thresholding is probably the most frequently used technique to segment an image. If we have an image which contains bright objects on a dark background, thresholding can be used to segment the image. Since in many types of images the grey values of objects are very different from the background value, thresholding is often a well-suited method to segment an image into objects and background as shown in Figure 3.8. Many methods exist to select a suitable threshold value for a segmentation task. Perhaps the most common method is to set the threshold value interactively. The thresholding operation is a grey value remapping operation  $g$  defined by:

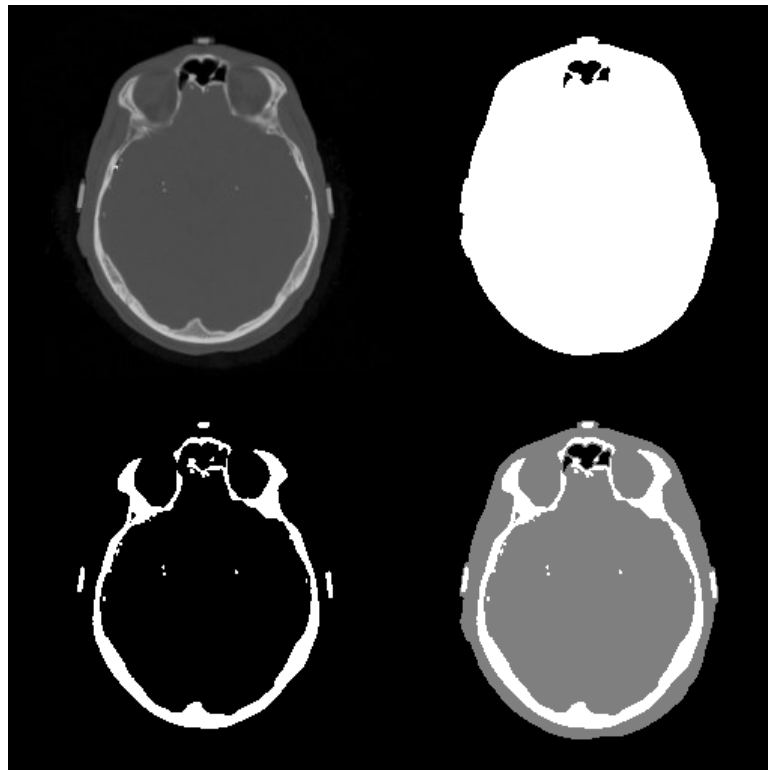
$$\begin{aligned}g(u) &= 0 \text{ for } u < t \\g(u) &= 1 \text{ for } u \geq t\end{aligned}$$

Where  $u$  represents a grey value, and  $t$  is the threshold value. Thresholding maps a grey-valued image to a binary image. After the thresholding operation, the image has been segmented into two segments, identified by the pixel values 0 and 1 respectively.

When several desired segments in an image can be distinguished by their grey values, threshold segmentation can be extended to use multiple thresholds to segment an image into more than two segments as we see in Figure 3.8. All pixels with a value smaller than the first threshold are assigned to segment 0, all pixels with values between the first and second threshold are assigned to segment 1, all pixels with values between the second and third threshold are assigned to segment 2, *etc.* If  $n$  thresholds ( $t_1, t_2, \dots, t_n$ ) are used:

$$\begin{aligned} g(u) &= 0 \text{ if } u < t_1 \\ g(u) &= 1 \text{ if } t_1 < u \leq t_2 \\ g(u) &= 2 \text{ if } t_2 < u \leq t_3 \\ g(u) &= n \text{ if } t_n \leq u \end{aligned}$$

After thresholding, the image has been segmented into  $n+1$  segments identified by the grey values 0 to  $n$  respectively.



**Figure 3.8** Example of using multiple thresholds for segmentation. Top left: original image. Top right: thresholding result after using a low threshold value to segment the image into head and background pixels. Bottom left: result after using a higher value to segment the bone pixels. Bottom right: result after using both thresholds at once [21].

### 3.3.3.2 Edge based segmentation

It was realized early on that human vision is sensitive to contours and there is a duality between simple closed contours and objects. A contour is a frontier separating two objects or an object and its background in the binary case can be defined locally, while an object usually cannot because an object can have an arbitrary extent. A simple closed contour (or surface) is one that is closed and does not self-intersect. By the Jordan theorem [22], in the Euclidean space, any such contour or surface delineates a single object of finite extent. There are some classical difficulties with the Jordan theorem in the discrete setting, but they can be solved by selecting proper object/background connectivities.

A gradient (first derivative) or a Laplacian (second derivative) operator can be used to define an object border in many cases, and gradients are less sensitive to illumination conditions than pixel values. As a result, contour detection through the use of gradient or Laplacian operators became popular, and eventually led to the Marr–Hildreth theory [23]. Given this, it is only natural that most segmentation method use contour information directly in some ways. Early methods used only this information to detect contours and then tried to combine them in some way. By far the most popular and successful version of this approach is the Canny edge detector [24]. In his classical paper, Canny proposed a closed-form optimal 1D edge detector assuming the presence of additive white Gaussian noise, and successfully proposed a 2D extension involving edge tracking using non-maxima suppression with hysteresis. The segmentation of an image into separate objects can be achieved by finding the edges of objects. A typical approach to segmentation using edges is done by following these steps:

- Compute an edge image, containing all (plausible) edges of the original image.
- Process the edge image so that only closed object boundaries remain.
- Transform the result to an ordinary segmented image by filling in the object boundaries.

The difficulty often lies in the middle step when transforming an edge image to closed boundaries often requires the removal of edges that are caused by noise or other artifacts, the bridging of gaps at locations where no edge was detected and intelligent decisions to connect those edge parts that make up a single object.

#### **Canny Edge Detection**

In industry, the Canny edge detection technique is one of the standard edge detection techniques. It was first created by John Canny for his Master's thesis at MIT in 1983, and still outperforms many of the newer algorithms that have been developed. To find edges by separating noise from the image before find edges of image the Canny is a very important method. Canny method is a better method without disturbing the features of the edges in the

image afterwards it applying the tendency to find the edges and the serious value for threshold. The algorithmic steps are as follows:

- Noise reduction

The Canny edge detector uses a filter based on the first derivative of a Gaussian, because it is susceptible to noise present on raw unprocessed image data, so to begin with, the raw image is convolved with a Gaussian filter. The result is a slightly blurred version of the original which is not affected by a single noisy pixel to any significant degree.

- Finding the intensity gradient of the image

An edge in an image may point in a variety of directions, so the Canny algorithm uses four filters to detect horizontal, vertical and diagonal edges in the blurred image. The edge detection operator (Roberts, Prewitt, Sobel) returns a value for the first derivative in the horizontal direction ( $G_x$ ) and the vertical direction ( $G_y$ ). From this the edge gradient and direction can be determined:

$$G = \sqrt{G_x^2 + G_y^2}$$
$$\theta = \tan^{-1} \frac{G_y}{G_x}$$

The edge direction angle is rounded to one of four angles representing vertical, horizontal and the two diagonals (0, 45, 90 and 135 degrees).

- Non-maximum suppression

Given estimates of the image gradients, a search is then carried out to determine if the gradient magnitude assumes a local maximum in the gradient direction.

- If the rounded gradient angle is 0 degrees ( the edge is in the north and south direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes in the west and east directions.
- If the rounded gradient angle is 90 degrees (the edge is in the east-west direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes in the north and south directions.
- If the rounded gradient angle is 135 degrees ( the edge is in the north east-south west direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes in the north west and south east directions.

- If the rounded gradient angle is 45 degrees (the edge is in the north west-south east direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes in the north east and south west directions.

From this stage referred to as non-maximum suppression, a set of edge points, in the form of a binary image is obtained. These are sometimes referred to as "thin edges".

- Hysteresis thresholding

The output of non-maxima suppression still contains the local maxima created by noise. Instead choosing a single threshold, for avoiding the problem of streaking two thresholds  $t_{high}$  and  $t_{low}$  are used. For a pixel  $M(i, j)$  having gradient magnitude  $G$  following conditions exists to detect pixel as edge:

- If  $G < t_{low}$  then discard the edge.
- If  $G > t_{high}$  keep the edge.
- If  $t_{low} < G < t_{high}$  and any of its neighbors in a  $3 \times 3$  region around it have gradient magnitudes greater than  $t_{high}$ , keep the edge.
- If none of pixel  $(x, y)$ 's neighbors have high gradient magnitudes but at least one falls between  $t_{low}$  and  $t_{high}$  search the  $5 \times 5$  region to see if any of these pixels have a magnitude greater than  $t_{high}$ . If so, keep the edge.
- Else, discard the edge.

### Edge Linking

We have seen that the most edge detectors yield information about the magnitude, the gradient and the direction of the specific edge at an edge point. The latter information is obviously useful when deciding which edge points to link together since edge points in a neighborhood which have similar gradients directions are likely to lie on the same edge [25]. We can link adjacent edge pixels by examining each pixel-neighbor pair and seeing if they have similar properties:

- Similar gradient magnitude:  $|\nabla f(x, y) - \nabla f(x, y)| \leq T$  for some magnitude difference threshold  $T$ .
- Similar gradient orientation  $|\phi(\nabla f(x, y)) - \phi(\nabla f(x, y))| \leq A$  for some angular threshold  $A$ .

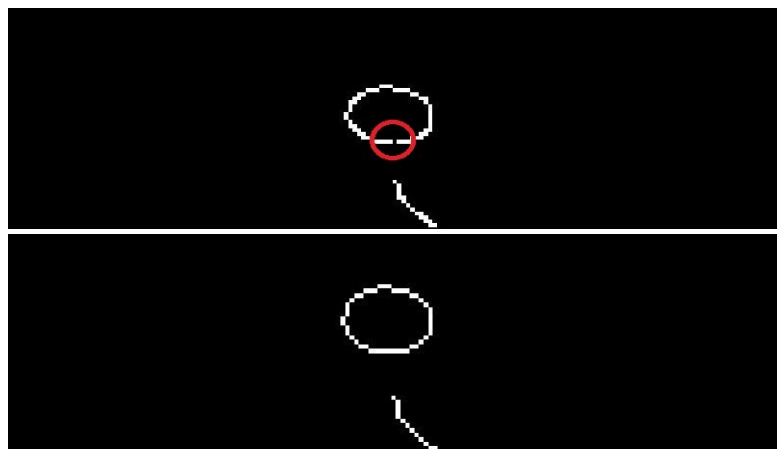
Once the links are established, we take sets of linked pixels and use them as borders. Notice that unless you constrain the linked pixels in some sense (for example, by scanning along

horizontal or vertical lines), these can create clusters of linked pixels rather than long single-pixel thick chains. Edge linking is usually followed by post-processing to find sets of linked pixels that are separated by small gaps— these can then be filled in. Edge detection is always followed by edge linking Local Processing.

- Analyze pixels in small neighborhood of each edge point.
- Pixels that are similar are linked. Edge pixel  $(x_0, y_0)$  is linked with  $(x, y)$  if both criteria are satisfied.
- Principal properties used for establishing similarity:
  - Magnitude of gradient vector.  
Edge pixel with coordinates  $(x_0, y_0)$  in neighborhood of  $(x, y)$  is similar in magnitude to pixel at  $(x, y)$  if  $|\nabla f(x, y) - \nabla f(x_0, y_0)| \leq E$ .
  - Direction of gradient vector.  
Edge pixel with coordinates  $(x_0, y_0)$  in neighborhood of  $(x, y)$  has an angle similar to pixel at  $(x, y)$  if  $|\alpha(x, y) - \alpha(x_0, y_0)| < A$ .

Edge linking, whose results are shown in Figure 3.9, bridges unconnected pixels, that is, sets 0-valued pixels to 1 if they have two nonzero neighbors that are not connected. For example:

1 0 0		1 1 0
1 0 1	becomes	1 1 1
0 0 1		0 1 1



**Figure 3.9** Edge-linking paradigm. Upper image is the original edge-tracked image and the lower image is the edge-linked image.

### 3.3.3.3 Region based segmentation

In theory, finding an object by locating its boundary and finding it by establishing the region it covers will give you exactly the same object; the boundary and the region are just different representations of the same object. In practice, however, taking an edge based approach to segmentation may give radically different results than taking a region based approach. The reason for this is that we are bound to using imperfect images and imperfect methods, hence the practical result of locating an object boundary may be different from locating its region [26]. Region based segmentation methods have only two basic operations: splitting and merging, and many methods even feature only one of these. The basic approach to image segmentation using merging is:

1. obtain an initial (over)segmentation of the image,
2. merge those adjacent segments that are similar to form single segments
3. go to step 2 until no segments that should be merged remain.

The basic form of image segmentation using splitting is:

1. obtain an initial (under)segmentation of the image
2. split each segment that is inhomogeneous in some respect.
3. go to step 2 until all segments are homogeneous.

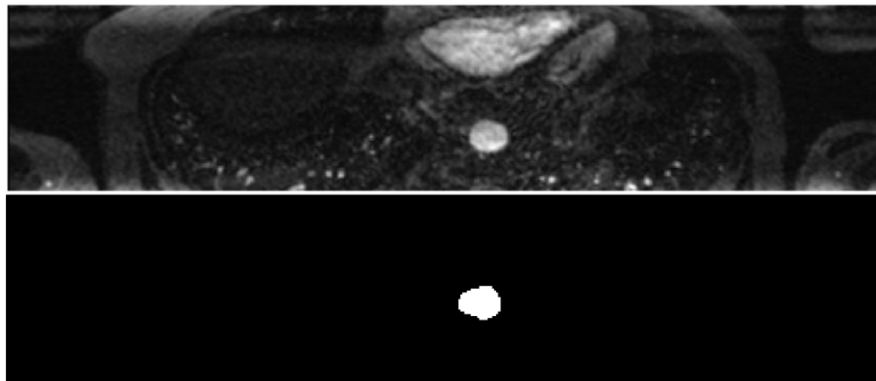
#### **Region Growing**

Many merging methods of segmentation use a method called region growing to merge adjacent single pixel segments into one segment. Region growing needs a set of starting pixels called seeds. The region growing process consists of picking a seed from the set, investigating all 4-connected neighbors of this seed, and merging suitable neighbors to the seed. The seed is then removed from the seed set, and all merged neighbors are added to the seed set. The region growing process continues until the seed set is empty. The algorithm below implements an example with a single seed, where all connected pixels with the same grey value as the seed are merged. Below the algorithm is described [26].

The data structure used to keep track of the set of seeds is usually a stack. Two operations are defined on a stack: push, which puts a pixel (or rather, its coordinates) on the top of the stack, and pop, which takes a pixel from the top of the stack. In the algorithm, the image is called  $f$ , the seed has coordinates  $(x, y)$  and grey value  $g = f(x, y)$ . The region growing is done by setting each merged pixel's grey value to a value  $h$  (which must not equal  $g$ ). The pixel under investigation has coordinates  $(a, b)$ . The final region can be extracted from the image by selecting all pixels with grey value  $h$ . To ensure the correctness of the result, we must select  $h$

to be a value that is not present in the original image prior to running the algorithm. An example of this is shown in Figure 3.10. The algorithm runs:

1. push ( $x, y$ )
2. as long as the stack is not empty do
  - (a) pop ( $a, b$ )
  - (b) if  $f(a, b) = g$  then
    - i. set  $f(a, b)$  to  $h$
    - ii. push ( $a - 1, b$ )
    - iii. push ( $a + 1, b$ )
    - iv. push ( $a, b - 1$ )
    - v. push ( $a, b + 1$ )



**Figure 3.10** Example of region growing based on a grey level range. On the left, an original 256×256 image is shown, with a grey level range of 256. On the right, the result of region growing with the seed roughly at the center of the structure, allowing a grey value range of  $\pm 30$  around the grey value of the seed [21].

## 3.4 Image Registration

Image registration is the process of bringing two or more images into spatial correspondence (aligning them). In the context of medical imaging, image registration allows for the concurrent use of images taken with different modalities (e.g. MRI and CT), at different times or with different patient positions. In surgery, for example, images are acquired before (pre-operative), as well as during (intra-operative) surgery. Because of time constraints, the real-time intra-operative images have a lower resolution than the pre-operative images obtained before surgery. Moreover, deformations which occur naturally during surgery make it difficult to relate



the high-resolution pre-operative image to the lower-resolution intra-operative anatomy of the patient. Image registration attempts to help the surgeon relate the two sets of images.

Registration has an extensive literature. Numerous approaches have been explored ranging from statistics to computational fluid dynamics and various types of warping methodologies. See [45, 46] for a detailed description of the field as well as an extensive set of references, and [47, 48] for recent papers on the subject. Registration typically proceeds in several steps. First, one decides how to measure similarity between images. One may include the similarity among pixel intensity values, as well as the proximity of predefined image features such as anatomical landmarks. Next, one looks for a transformation which maximizes similarity when applied to one of the images. To name a few examples, where registration is required:

- Registration of image data acquired at different points in time. In dynamic imaging, image data are acquired at different time points. A robust mathematic analysis of these data requires the correction for motion artifacts.
- Registration of pre- and intraoperative data. Therapy monitoring is based on intraoperative imaging. It is desirable to relate intraoperative data to analysis results derived from preoperative data. Together with navigation systems, a correct transformation of intraoperative data to preoperative data might be used to support the localization of a pathology.
- Multimodal registration. A wide area of application is multimodal image registration, in which different acquisition techniques, such as CT and MRI, are used complementarily.
- Atlas-based matching. Finally, image registration is often employed to compare the data of a particular patient with an atlas that represents normal anatomical variations.

The goal of image registration is to deform or transform one dataset to optimally match another dataset that provides different information as well as similar information. Usually, single components of the registration process (transformations, similarity measures, optimization, interpolation, image representation) are considered as subclasses in research publications. As in segmentation, there is no standard approach. The registration problem can be formulated as follows:

- Transformation: geometric transformation of voxel coordinates
- Fitting: requires a quantification by means of a similarity measure
- Optimally: The transformation should be accomplished in such a way that the similarity measure is maximized.
- A given aspect: The criteria for optimal matching are chosen such that particular structures are matched as good as possible. As an example, the goal might be to match vascular structures, or skeletal structures, or organs. It is essential to note that optimal

correspondence with respect to a given aspect is often achieved at the expense of other aspects.

**Global Transformation.** Translation and rotation of (all) coordinates are examples for global transformations. These transformations are described by a small set of parameters that is applied to all coordinates. With global transformations, simple movements may be corrected. The modification of one parameter has an influence on all voxels, which is not desirable if more complex transformations have to be represented.

**Local Transformations.** These transformations are described by a large set of parameters that correspond to a mesh of control points. Modifications of a single parameter (control point) affect only a local neighborhood. Examples for local transformations are cubic b-spline transformations and Bezier transformations. For each voxel coordinate, a new set of parameters is employed. Local transformations enable the accounting of more complex movements. Other local approaches utilize elastic models or fluid models to characterize movements. The numeric solution of these models is accomplished by means of partial differential equations, which is computationally extremely expensive.

**Fitting** Similarity measures characterize how similar two images are. Basically, similarity measures based on intensities of voxels and on geometric measures are discriminated:

- Intensity-based similarity. Gray values of voxels in the floating image are compared with voxels in the reference image.
- Geometry-based similarity. Positions of voxels in the floating image are compared to those in the reference image .

There are three categories of intensity-based similarity measures :

- Voxel-based. The intensities of the two images are compared for each voxel in a 1:1 relation. This measure is only applicable if the two images have the same resolution and if both images contain the same content with similar intensities. This assumption is not fulfilled in multimodal registration where different modalities are involved. An application example is time-dependent data with limited motion artifacts, such as cerebral perfusion.
- Statistic. The normalized correlation of intensities in both images is computed. The application of this measure also requires that the same content be represented in both images. Linear transformations of intensity values can be compensated.

- Entropy-based. These methods are based on information theory. The mutual information of the common 2D histogram of both images is computed. Normalized mutual information is a similarity metric, which may be used even if the content of the images is slightly different. The relationship between the intensity values may be statistical. Registration based on normalized mutual information is able to match multimodal image data, such as CT and MRI.

All these similarity measures might be computed automatically (without any interaction). The determination of landmarks in the floating image, as well as in the reference image in general, requires user input. The automatic localization of corresponding landmarks is difficult. In many cases, branching points of vascular trees are appropriate. Intensity-based and similarity-based registration are also combined. Normalized-mutual information is the most frequently used similarity measure in particular for multimodal registration. However, the use of this measure also has considerable drawbacks. Typically, it has many local minima, which make the optimization process difficult.

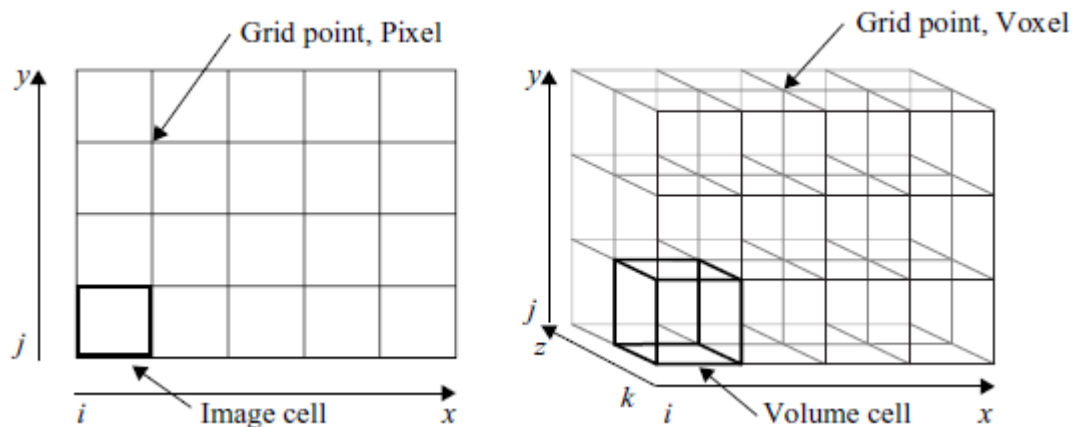
**Optimization Process** For the maximization of the selected similarity measure, a numerical optimizer is used to approximate “optimal” transformation parameters. Numerical optimization is either applied to the similarity values (for example, simplex, Powell, or Hooke approach) or considers derived information. The registration with landmarks is often based on the Levenberg-Marquardt algorithm.

### 3.4.1 Medical visualization

The purpose of scientific visualization is to extract meaningful information from complex datasets through the use of interactive graphics. Volume visualization is concerned with the representation, manipulation, and rendering of 3D volumetric data. In the context of medical imaging this usually involves revealing underlying anatomy and organ morphology for diagnostic or treatment planning purposes [27]. Volume visualization involves three distinguishable aspects, data representation, data classification or surface extraction, and data viewing. The primary source of volume data in medical imaging is empirical. A discrete sampling of an object, usually as a sequence of cross-sectional scans, is generated by a medical scanner. Most viewing algorithms use the 3D voxel array representation. The volumetric dataset resides in a discrete voxel space which is a 3D integer grid of unit volume cells or elements called voxel.

A voxel is the 3D counterpart of the 2D pixel. Each voxel is a quantum unit of volume and has a numeric value associated with it that represents some measurable properties of the sampled

objects (e.g. density, refractive index, acoustic impedance, velocity). Most visualization methods require that the voxels lie on a uniform orthogonal grid. The uniform voxel representation is convenient for the storage and manipulation of volumetric data within a computer, and also for interpolating between voxel centers. A voxel is either represented as a rectangular prism centered at a grid point or, interchangeably, as a zero-dimensional point located at the grid coordinates. The aggregate of voxels tessellating the volume forms the volumetric dataset. CT and MRI datasets are conveniently reconstructed into a regular 3D voxel data by stacking parallel cross-sections. However, non-parallel slices, such as those acquired by a six degrees-of-freedom hand-held ultrasound probe, are not readily suited to this representation. These require the development of new viewing algorithms or alternatively, reconstruction techniques which generate a regular array of voxels from the data and therefore allow conventional methods to be used.



**Figure 3.11** Left: A 2D grid, where all pixels of an image are arranged on the grid points of the grid. Right: In volume datasets, the voxels are arranged on a 3D grid.

Medical 3D image reconstruction is a part of a field called Visualization. All recent medical 3D image reconstruction techniques create 3D images from sets of 2D slices, which can be recorded by various equipments such as CT, MRI, ultrasound etc. Each type of scanner has his own characteristics due to physical principles of image recording (images of CT scanner are often parallel slices with high contrast, images of ultrasound scanner are either parallel or divergent slices with low contrast). Thus there are different 3D reconstruction techniques for each type of data as shown in Figure 3.11 [27]. However, the general principle of 3D reconstruction is composed of following steps:

- Step 1: 2D data slices need to be read and arranged exactly with the real spatial positions, the result is a data volume. This data volume is saved in any memory of computer.

- Step 2: use rendering techniques to visualize data volume as 3D image. Usual rendering techniques for medical image are multi-planar rendering (MPR), surface rendering (SR) and volume rendering (VR).

### **Multi-planar rendering technique**

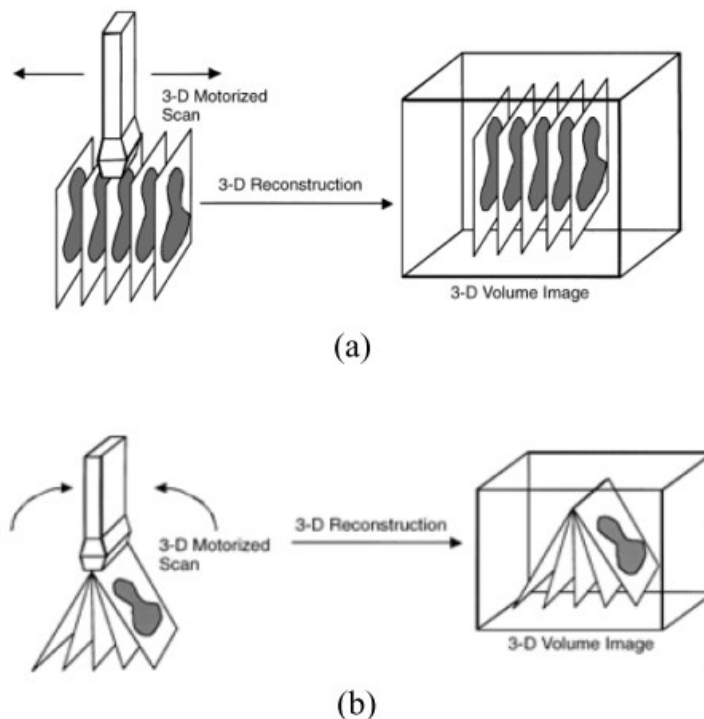
MPR does not require too many calculations, so it is appropriate for low configuration computers. This technique can be used to re-slice structure, i.e. with axial slices we can use MPR technique to re-slice according to different directions such as coronal, sagittal or diverse.

### **Surface rendering technique**

SR technique visualizes a 3D object as a set of surfaces called iso-surfaces. Each surface contains points which have the same intensity (called iso-value) on all slices. This technique is used when we want to see the surfaces of a structure separately from near structure, e.g. skull from slices of head, blood vessel system from slices of body etc. SR technique is often used for high contrast data.

Two main methods for reconstructing iso-surfaces can be considered as follows:

- Contour based reconstruction: Iso-contours, which are extracted from each slice can be connected to create iso-surfaces
- Voxel based reconstruction: Iso-surfaces are built directly from voxels having identical intensity (iso-value). One of the best algorithms is Marching Cubes.



**Figure 3.12** Parallel (a) and divergent (b) types of 2D slice [28].

### **Isosurfacing – Marching Cubes**

Isosurfacing is an approach where a surface is formed from a cross connection of data points, within a volume, of equal value or density. Isosurface methods typically approximate an isosurface with a polygon mesh and shade the mesh in standard graphics pipeline. Since the polygon mesh is constructed in the object space, the graphics display can response rather quickly to a change of the view point. This makes isosurface approach well suited for an interactive environment. Isosuracing methods reduce the data to store. A popular method of constructing an isosurface from a data volume is the marching cubes algorithm.

William E. Lorensen, Harvey E. Cline, has presented a basic algorithm for surface extraction known as Marching Cubes [29]. It uses the surface configurations of a cube for surface rendering of the volume data. In the marching cubes method, a volume data is first partitioned into cubes. Each cube consists of eight voxels. The fundamental problem is to form a facet approximation to an isosurface through a scalar field sampled on a rectangular 3D grid. Given one grid cell defined by its vertices and scalar values at each vertex, it is necessary to create planar facets that best represent the isosurface through that grid cell. The isosurface may not be pass through the grid cell, it may cut off any one of the vertices, or it may pass through in any one of a number of more complicated ways. Each possibility will be characterised by the number of vertices that have values above or below the isosurface. If one vertex is above the isosurface say and an adjacent vertex is below the isosurface then we know the isosurface cuts the edge between these two vertices. The position that it cuts the edge will be linearly interpolated, the ratio of the length between the two vertices will be the same as the ratio of the isosurface value to the values at the vertices of the grid cell (Figure 3.12). After determining the surface configuration of cube, the surfaces of every two adjacent neighbor cubes are combined to form the surface of an object. The surface of an object is then projected to a plane to form the final image. While generating triangles, Marching cubes sometimes results in false positive and negative triangles in the iso-surface.

### **Volume rendering technique**

VR technique is used to visualize the entire volume transparency of the object. Images will be performed by projecting rays through volume data. Along each ray, opacity and color need to be calculated at every voxel. Then information calculated along each ray will to be aggregated to a pixel on image plane. This technique helps us to see comprehensively an entire compact structure of the object. One of disadvantages of this technique is enormous amount of calculations, which requires strong configuration computers. This technique is appropriate for low contrast data. Two main methods for rays projecting can be considered as follows:

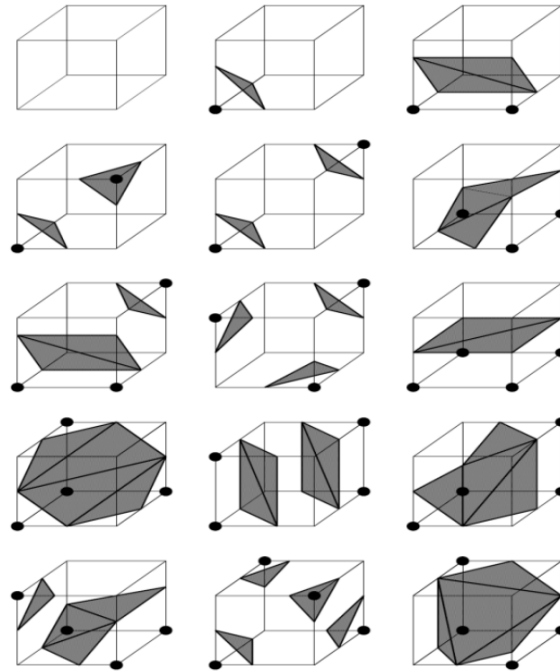
- Object-order or ray-tracing method: Projecting rays go through volume from back to front (from volume to image plane).
- Image-order or ray-casting method: Projecting rays go through volume from front to back (from image plane to volume). There exists some other methods to composite

image, appropriate methods depending on the user's purposes. Some usual methods in medical image are MIP (maximum intensity projection), MinIP (minimum intensity projection), AC (alpha compositing) and NPVR (non-photorealistic volume rendering).

### **Ray casting**

Ray-casting is an image-oriented method which can be used to render opaque or transparent surfaces. At first it was applied to the display of binary volumetric data. Subsequent implementations have varied primarily in their use of transparency, shading and in coding details which improve the computational efficiency of the method. This discussion is restricted to rendering opaque, iso-value surfaces from regular voxel array data. Figure 3.8 illustrates the principle of the ray-casting method used to render surfaces in this thesis. This figure suggests a perspective projection, but parallel rays are often used in medical imaging [39].

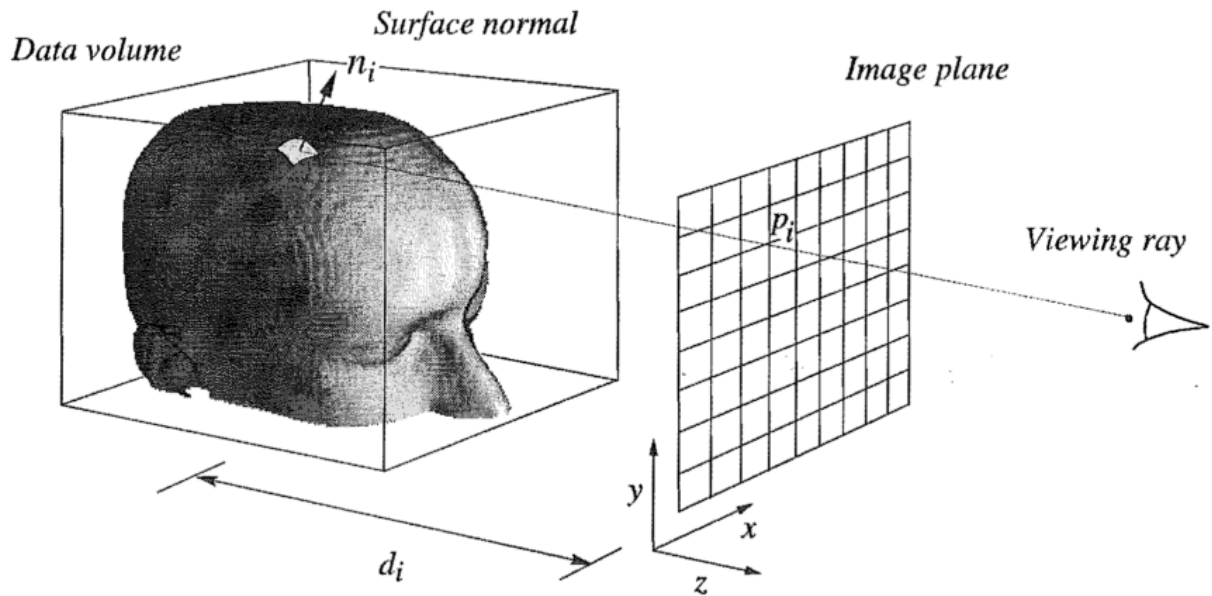
For every pixel in the output image a ray is shot into the data volume. At a predetermined number of evenly spaced locations along the ray the color and opacity values are obtained by interpolation. The interpolated colors and opacities are merged with each other and with the background by *compositing* in back-to-front order to yield the color of the pixel. These compositing calculations are simply linear transformations. Specifically, the color of the ray  $C_{out}$  as it leaves each sample location, is related to the color  $C_{in}$  of the ray, as it enters, and to the color  $c(x_i)$  and the opacity  $a(x)$  at that sample location by the transparency formula :



**Figure 3.13** Triangulated cubes [27].

$$C_{out} = C_{in} * (1 - a(x_i)) + c(x_i)a(x_i)$$

Performing this formula in a back-to-front order, i.e. starting at the background and moving towards the image plane, will produce the pixel color. It is clear from the above formula that the opacity acts as a data selector. For example, sample points with opacity values close to 1 hide almost all the information along the ray between the background and the sample point and opacity values close to zero transfer the information almost unaltered. This way of compositing is equal to the dense-emitter model, where the color indicates the instantaneous emission rate and the opacity indicates the instantaneous absorption rate.



**Figure 3.14** The ray-casting algorithm used in this thesis. For each pixel  $P_i$  in the image plane a ray is cast into the data volume until a surface, determined by some function of the intensity profile, is encountered. The distance  $d_i$  from the image plane to the surface is stored in a depth-buffer or Z-buffer together with the grey-level gradient computed locally at the surface point. Images are rendered from depth and gradient information in the Z-buffer [30].



### 3.5 NURBS- Non Uniform Rational B-Splines

The use of NURBS to properly represent the surface of biological tissues is superior to just using a typical triangular or quadrilateral mesh elements, due to the series of splines that can be generated to capture the surface compared to a series of lines [31]. When working with an ordered point cloud the complications increase. Generating a smooth surface from a point cloud without the use of NURBS is an issue because refining the mesh will not change how smooth the mesh is. However, when generating a NURBS surface which in turn generates a series of B-splines a smooth and continuous surface can be generated with smoother interpolations. The main idea of NURBS is the elimination of the node mesh in the analysis process [40]. The role and properties of the node mesh are inherited by two separate meshes, obtained directly from the geometrical representation:

- The Control Point mesh, which defines geometry and the finite number of degrees of freedom that form the problem equation.
- The Knot mesh, which provides appropriate discretization for numerical integration and boundaries for Shape function influence in the model.

For the scope of this thesis, I have worked exclusively with Non-Uniform Rational B-SPLines (NURBS), as they are the most commonly used computational geometry technology. Despite the fact that quite more advanced SPLines have emerged, CAD industry still invests in NURBS. Both professionals and amateurs still use NURBS despite their disadvantages, such as difficulties in Patch connection and local Refinement [41]. The reason for this is that NURBS are not only much more simple in their definition and use, but also able to represent with accuracy smooth curves and all conic sections [35].

#### **Index, Parameter and Physical Space**

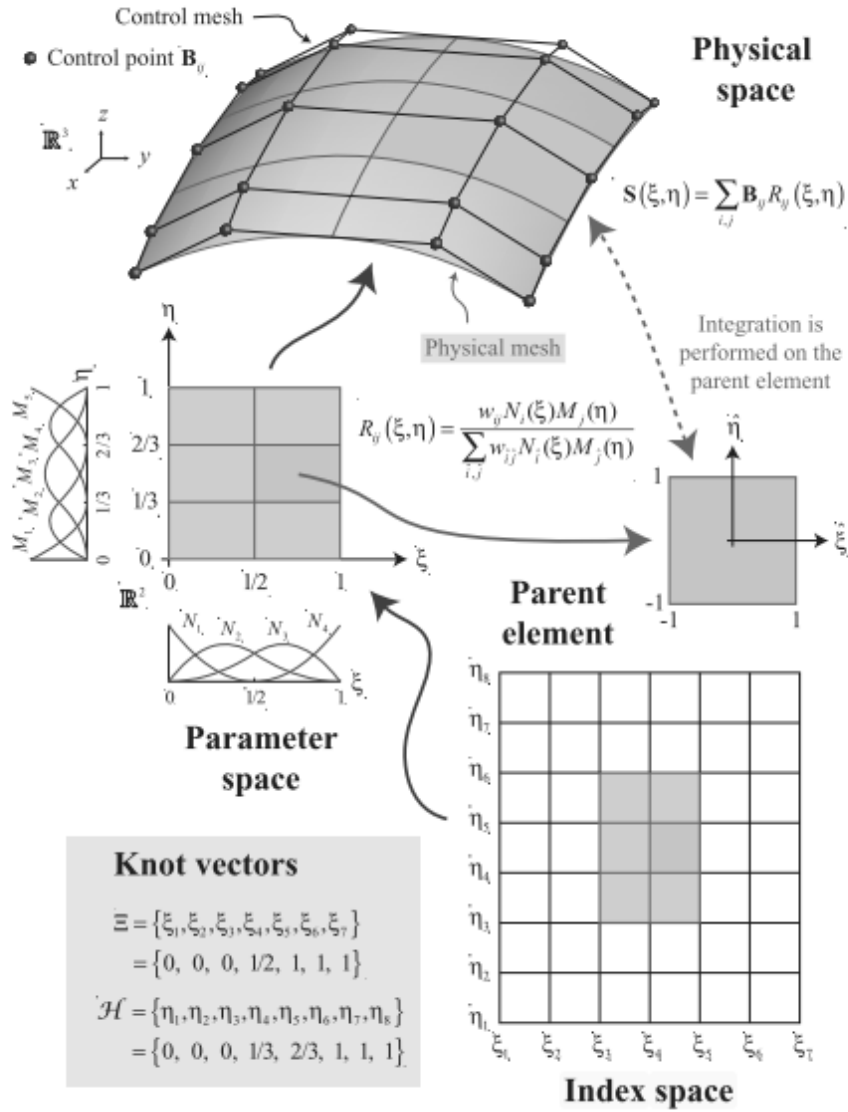
Accurate geometrical representations of the natural model are designed in the familiar Cartesian system, called Physical Space. Additionally, it is very helpful to envision a complex structure in an imaginary, basic space, where all geometries can be represented as lines, rectangles and cuboids. This is Parameter Space. This approach is far from new; it is already known from the isoparametric concept in Finite Element Methods. The Parameter Space utilized in Isogeometric Analysis, however, holds some major differences. Furthermore, Isogeometric Analysis also introduces the Index Space. This additional space plays an important role for some kinds of SPLines, but it is only auxiliary for NURBS.

Index Space is a representation of the model with respect to Knot Values. As shown in Figure 3.15, it is a line in 1D, containing the corresponding Knot Values in equally spaced positions. This space focuses upon the sequence of Knot Values rather than their actual numerical

content. Index Space describes the contribution of each Knot Value to the creation of a certain B-Spline basis function. This helps identify the level of interconnection between basis functions and the Knot Value support of each function. Control Points are also evaluated in the Index Space. In fact, Control Points are defined as the center of the support of Knot Value Spans. Expansion to 2D or 3D leads to the creation of rectangles or cuboids respectively. Due to tensor product properties, everything mentioned about 1D extends and applies to both 2D and 3D. Thus, Index Space provides information that can contribute to the comprehension of a complex representation.

Parameter Space is a representation of the model with respect to Knots as shown in Figure 3.15. Spline entities are always represented as orthogonal shapes in Parameter Space. Only lines, rectangles and cuboids exist here. In order to transform those simple patterns to virtually unlimited, complex geometries, the application of a mapping from Parameter to Physical Space is required. Hence, Parameter Space is a primitive, abstract representation of Physical Space. The mapping between Parameter Space and Physical Space is achieved through the Jacobian Matrix and its inverse. This is something widely utilized in FEM as well. The illustration of basis functions in the Parameter Space allows for a better understanding of concepts such as support, Control Point coordinates and the role of Knots in basis function creation. Each Knot marks the beginning and the end of a basis function domain. By “domain” we mean the area in which the basis function is non-zero, as all basis functions are defined throughout the Parameter Space, but are non-zero only in specific Knot Spans. Basis functions sharing the same domain are overlapping in Parameter Space and controlling a common part of the entity in the Physical Space.

Physical Space is the already known Cartesian Space, where the real model is represented. Simple orthogonal shapes from Parameter Space are transformed into complex entities in the Physical Space. Physical coordinates of the Control Points play a major role in the aforementioned mapping, but an equally drastic role is set upon basis functions. In fact, for a given set of Control Points, only a single set of basis functions can lead to the same geometry. We will examine this thoroughly later. Control Points can often be seen outside the model in Physical Space in contrast to FEM’s nodes which always belong to the mesh. It is one of the reasons NURBS and Spline entities in general can accurately represent multiple types of geometries and the understanding of this peculiarity is one of the many challenges of Isogeometric Analysis.



**Figure 3.15** Schematic illustration of NURBS of a surface. Open knot vectors and quadratic C1-continuous basis functions are used. Also depicted are C1-quadratic ( $p = 2$ ) basis functions determined by the knot vectors. The paradigm above is a clear explanation of physical, index and parameter space [32].

### 3.5.1 NURBS Basis Function

The generation of the basis functions which when pieced together generated the NURBS curve or surface are generated using the knot values. The basis functions are a set of piecewise functions that are generated based on the desired degree (power). The basis function is generated in a way to smoothly fit through a set of points. Each basis function can be represented by  $N_{i,p}$ , where  $i$  is associated with the particular point and  $p$  represents the power.

Because each basis function is associated with a specific knot points the notations becomes  $N_{i,p}(u)$ . This term is defined in the Equations below :

$$N_{i,0}(\xi) = \begin{cases} 1, & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} \cdot N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} \cdot N_{i+1,p-1}(\xi)$$

For degrees greater than zero the basis function is simply an addition of two basis function with one lower degree. The basis functions have a triangular dependency this can be more visually seen in the Figure 3.16.

### 3.5.2 NURBS Shape Functions

In order to evaluate NURBS Shape functions, the Weighting function is defined:

$$W(\xi) = \sum_{i=1}^n \{N_{i,p}(\xi) \cdot w_i\}$$

In most engineering applications, Weights have positive values. Unless otherwise stated, they will be considered positive for the scope of this thesis.  $W(\xi)$  is in fact the Z-coordinate of the projective B-Spline curve. Projective transformation is applied by dividing the other two coordinates of the B-Spline curve with the Z-coordinate. NURBS Shape functions are calculated from

$$R_i^p(\xi) = \frac{N_{i,p}(\xi) \cdot w_i}{W(\xi)} = \frac{N_{i,p}(\xi) \cdot w_i}{\sum_{i'=1}^n \{N_{i',p}(\xi) \cdot w_{i'}\}}$$

$R_i^p(\xi)$  are piecewise rational functions. The expression “the order of NURBS” refers to the order of the projective B-Spline curve. NURBS Shape Functions in multiple directions can be obtained as tensor products of one-directional basis functions:

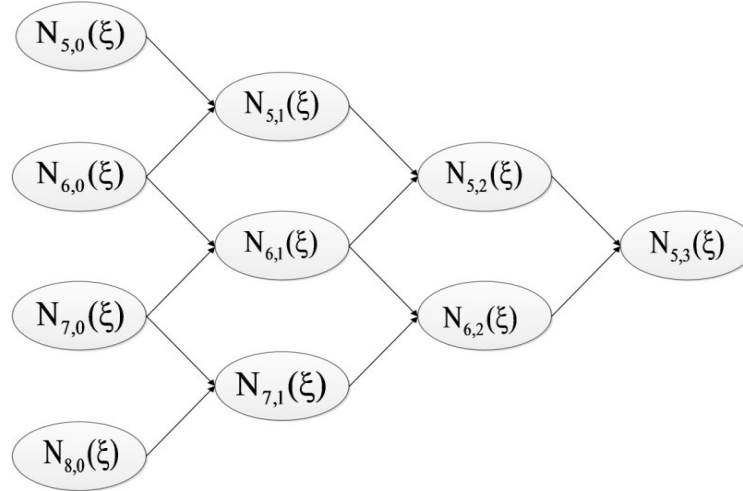
Shape functions for three directions:

$$R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) = \frac{N_{i,p}(\xi) \cdot M_{j,q}(\eta) \cdot L_{k,r}(\zeta) \cdot w_{ijk}}{\sum_{i'=1}^n \sum_{j'=1}^m \sum_{k'=1}^l \{N_{i',p}(\xi) \cdot M_{j',q}(\eta) \cdot L_{k',r}(\zeta) \cdot w_{i'j'k'}\}}$$

The Weighting function is now defined as:

$$W(\xi, \eta, \zeta) = \sum_{i'=1}^n \sum_{j'=1}^m \sum_{k'=1}^l \{N_{i',p}(\xi) \cdot M_{j',q}(\eta) \cdot L_{k',r}(\zeta) \cdot w_{i'j'k'}\}$$

Observe that for  $w_{ijk} = 1$ ,  $\forall i, j, k$ , it applies that NURBS Shape functions downgrade to B-Spline basis functions. Actually, NURBS entities are a generalization of B-Spline entities. All the B-Spline properties examined in this thesis apply for NURBS as well.



**Figure 3.16** Basis function triangular dependency each higher order basis function is dependent on two basis function of the one lesser order. Lower-order basis functions required for the creation of  $N_{5,3}(\xi)$ .

### 3.5.3 NURBS curves and surfaces

In order to generate a NURBS curve, it is important to make sure there are points in Cartesian space that can depict the position and direction the NURBS needs to take. These points are referred to them as control points. The control points are used as guidance for the basis function when generating a NUBRS curve. The NURBS curve may or may not actually lie on a control point. The connectivity between the control points form what is known as the control point polygon, an example of this can be seen in the Figure 3.17. NURBS entities are created as

a linear combination of NURBS Shape functions, exactly the same way as B-Spline entities. The following is the equation for the creation of NURBS Curves:

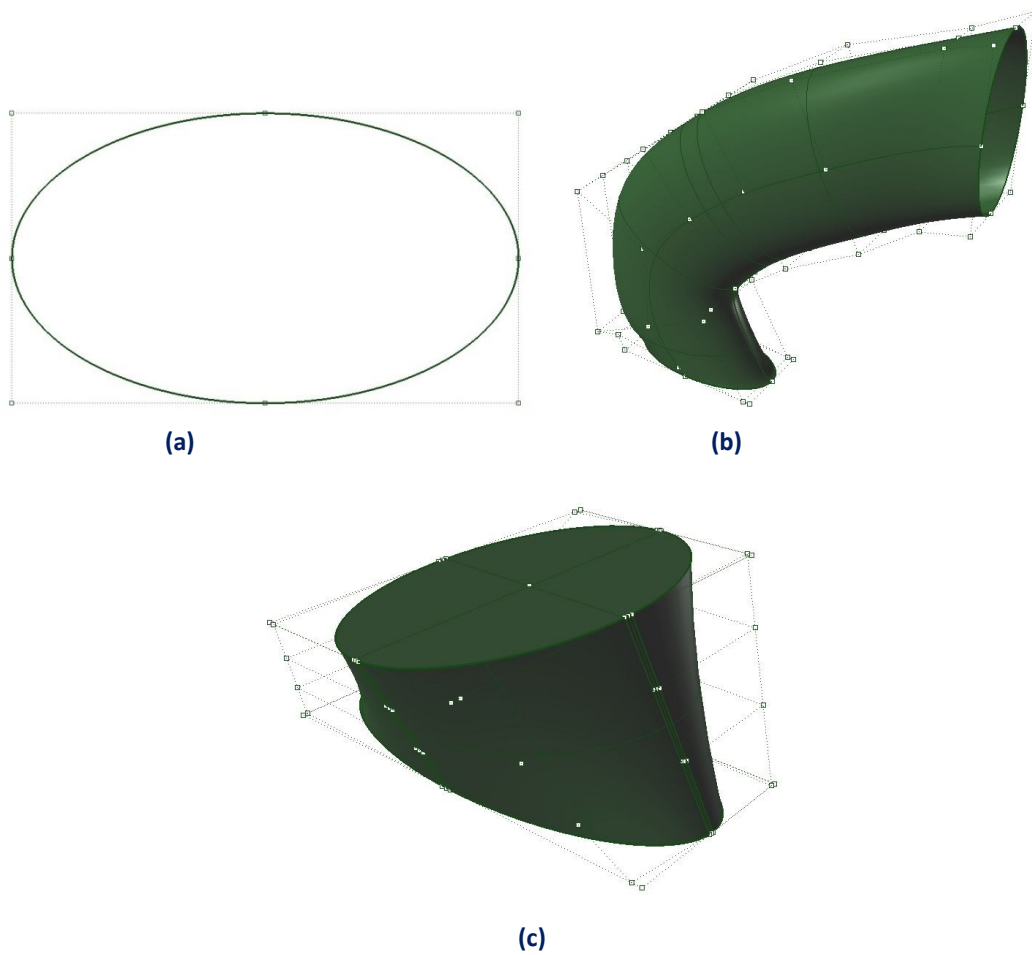
$$C(\xi) = \sum_{i=1}^n \{R_i^p(\xi) \cdot X_i\}$$

Surfaces:

$$S(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m \{R_{i,j}^{p,q}(\xi, \eta) \cdot X_{i,j}\}$$

and Solids:

$$S(\xi, \eta, \zeta) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l \{R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) \cdot X_{i,j,k}\}$$



**Figure 3.17** NURBS elliptical Entities. (a) Curve, (b) Surface and (c) Solid.

## 4 Proposed approach of aorta reconstruction

### 4.1 Image Processing, Segmentation and Reconstruction

In the first main stage of segmentation and reconstruction, we create an automatic iterative algorithm which segments the aortic outer borders from each image of the stack and reconstruct the 3D volume using image registration methods.

#### 4.1.1 Pre-processing

Our procedure begins with a stack of DICOM files, which is the international standard for medical images and related information. It defines the format for medical images that can be exchanged with the data and quality necessary for clinical use. Each of these files except from image information contains also multiple information about the patient, the study description and the specifications of the medical instrument which has been used for imaging. In vascular applications usually are used multi-frame datasets (bunch of single-frame images) for covering the whole abdominal or thoracic region. Single-frame gray-scale images represented as an M-by-N array, on the other hand, single-frame true-color images represented as an M-by-N-by-3 array.

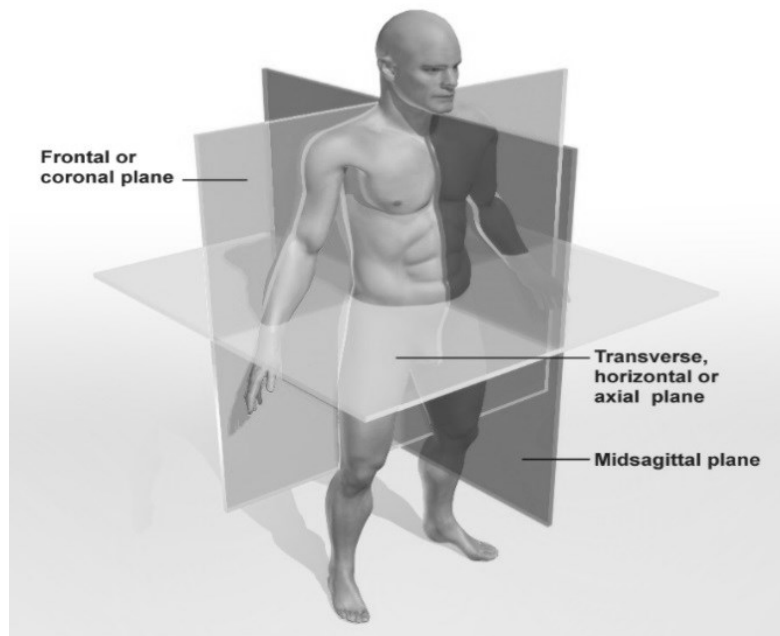
The optimal form to process our data is gray-scale images with 8 bit per pixel and pixel spacing from 0 (black) up to 255 (white). Pixel values represent the intensity values of images. So we use two methods. If the image information of DICOM files is in RGB format we use conversion to gray-scale format method. If the image information of DICOM files is in gray-scale format we use scaling method. There are several algorithms for converting an RGB image to gray-scale. In our case we applied the third technique.

- The lightness method averages the most prominent and least prominent colors.  
$$(\max(R, G, B) + \min(R, G, B)) / 2.$$
- The average method simply averages the values:  $(R + G + B) / 3.$
- The luminosity method is a more sophisticated version of the average method. It also averages the values, but it forms a weighted average to account for human perception. We're more sensitive to green than other colors, so green is weighted most heavily. The formula for luminosity is  $0.3 R + 0.6 G + 0.1 B.$

In many applications are used gray-scale images of 16 bits per pixel coding. In our case we need to be sure that we convert the data into a common coding format before the main procedure starts. So we choose to scale the data of the images to the range of [0 255], which is the 8-bit per pixel coding format. We use the Min-Max scaling method also called “normalization”. In this approach, the data is scaled to a fixed range in our case [0 255]. The cost of having this bounded range - in contrast to standardization - is that we will end up with smaller standard deviations, which can suppress the effect of outliers. A Min-Max scaling is typically done via the following equation:

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

So we convert the primary pixel values into values in the range of 0 (black) to 255 (full intensity or white). Each image that we have is a coronal slice of thoracic and abdominal region. These slices are vertical planar cross-sections of a specific upper human’s body.



**Figure 4.1** Anatomical planes. An anatomical plane is a hypothetical plane used to transect the human body, in order to describe the location of structures. Three principal planes are used: coronal, axial, sagittal.

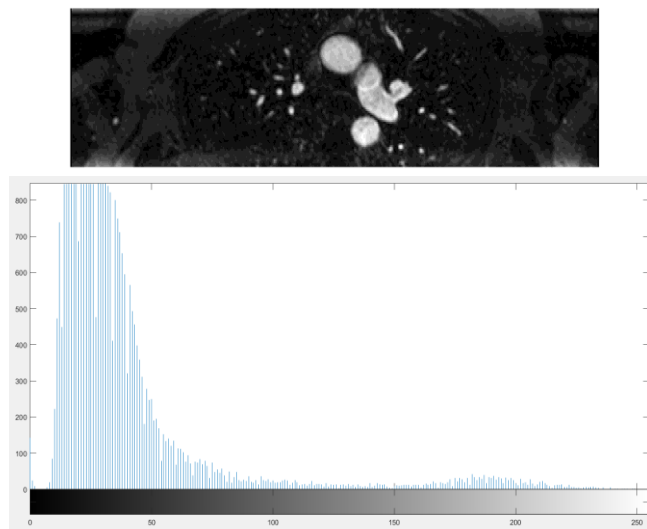


## Image enhancement

In this stage we have already converted the coding format of images and we have to apply image enhancement techniques to improve the interpretability or perception of information in images for human's eye. In other words we need to enhance contrast of images and to remove noise from them. The principle objective of image enhancement techniques is to process an image so that the result is more suitable than the original image for a specific application. Some of these techniques are:

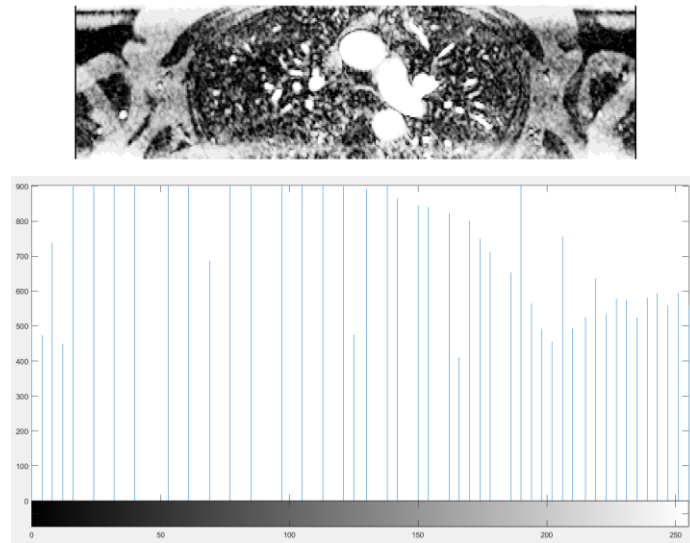
- *Histogram equalization*
- *Adaptive histogram equalization (AHE)*
- *Gamma correction and linear mapping*

In our case we use these three methods in order to evaluate them and select the one with better results. At first we analyze the images generating each image histogram. Histogram of an image shows frequency of pixels intensity values. The x axis shows the gray level intensities and the y axis shows the frequency of these intensities. As shown in Figures 4.2, that most of the bars that have high frequency lies in the first half portion which is the darker portion. That means that the image we have got is darker. And this can be proved from the lightness of image too.



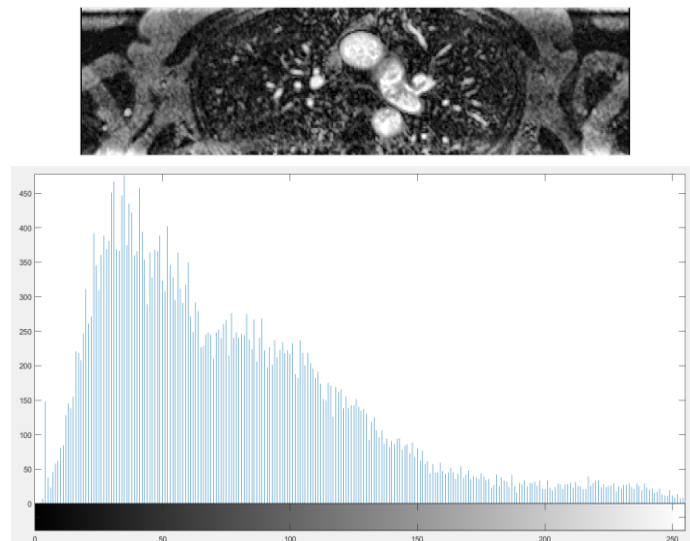
**Figure 4.2** Histogram of prior image. The information we can extract from this histogram is that the image should be a “dark” image. We can understand that from its histogram distribution.

- Firstly we inspect the “Histogram equalization”. The outcomes of this method are shown in Figure 4.3.



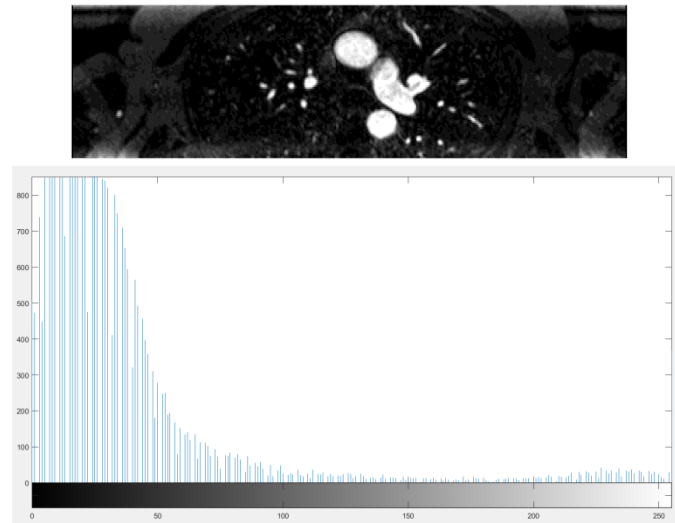
**Figure 4.3** Histogram equalization. The values of histogram have been equalized. The distribution of this histogram has small deviations.

- Secondly we inspect the “Adaptive Histogram Equalization” method . The results are shown in Figure 4.4.



**Figure 4.4** Adaptive Histogram Equalization (AHE). Comparing the histogram of prior image with this histogram, we can understand that this technique transforms histogram peaks in order to make the image lighter, and to enhance the contrast.

- Lastly we inspect the “Gamma correction” method [33]. The results are shown in Figure 4.5.



**Figure 4.5** Gamma Correction. Comparing the histogram distribution of processed image to the prior image we can see that gamma correction smooths image’s distribution function in order to have better contrast results.

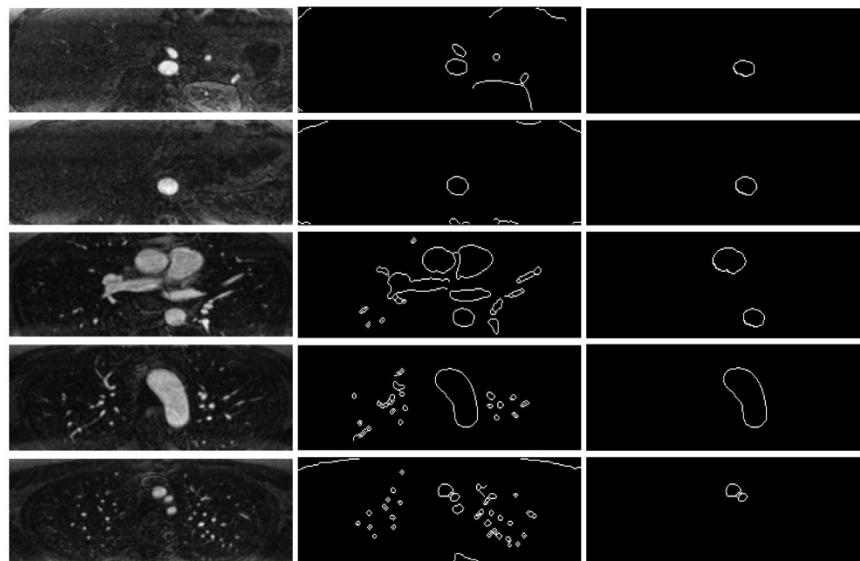
From the results of these three methods we can make some important conclusions. At first it makes sense that the method of “Histogram Equalization” does not fit in our method. This method gives to us much more information than we need. Furthermore it spoils the local contrast in the region of aortic boundaries, so we need to examine the other two methods. “Adaptive Histogram Equalization” is a very trustworthy method for image enhancement because it functions locally to images’ regions, so it enhances the contrast with local characteristics. In our approach we compare this method with the one of “Gamma Correction”. Both of these techniques give to us good results, because each one of them smooth the histogram’s curve, so we take more reliable contrast based results. At last we choose the “Gamma Correction” method because in our operation the only thing we need is to take good results in the region of our interest which is the region that includes aortic boundaries. Furthermore “Gamma Correction” is computationally more effective.

### 4.1.2 Cross-section detection (Segmentation)

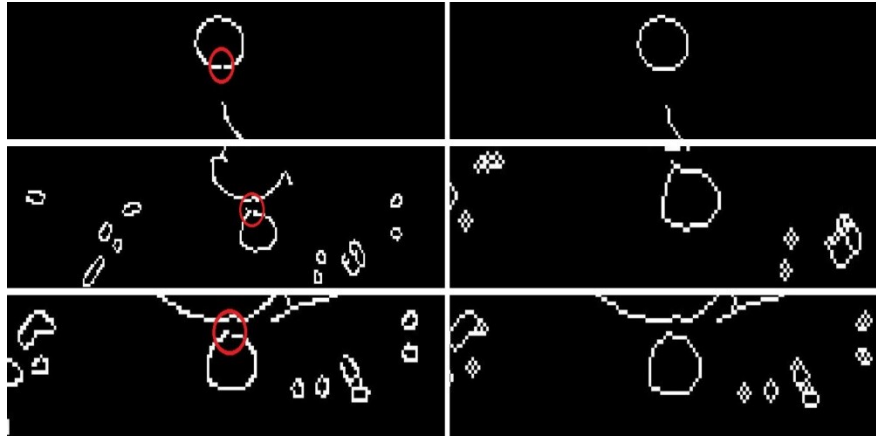
In this stage we process several images from the stack of axial slices in order to segment aortic boundaries' cross-sections. Our goal is to find and locate the thoracic aortic boundaries along the whole aortic path. We apply many useful techniques of image processing to segment the requested part of the body. At first we need to compute credible contours of an image. We use the "Canny Edge Algorithm". Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It is also called the "optimal" edge detector. In this situation, an "optimal" edge detector means:

- *Good detection. The algorithm should mark as many real edges in the image as possible.*
- *Good localization. The edges marked should be as close as possible to the edge in the real image.*
- *Minimal response. A given edge in the image should only be marked once, and where possible, image noise should not create false edges.*

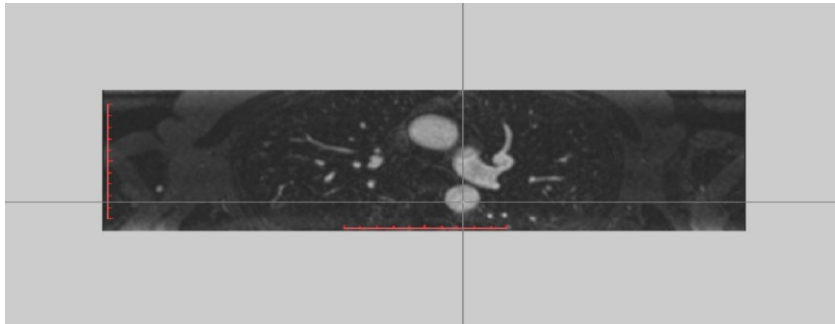
When we have finished with computing images' credible contours, our images are in binary form and the morphological shape of them represents the edges of primary images. These images are shown in the Figure 4.6. These images display the contours of the elements that exist. In order our main algorithm to be functional from the previous contours at least the one that refers to aorta need to be continuous and closed. The second step of cross-section procedure is vital for our pipeline. This step is the edge linking algorithm.



**Figure 4.6** The first stack presents input axial images, the second stack outcomes of "Canny edge detection" algorithm and the third outcomes of the seeded region growing algorithm. Images are sorted from upwards to downwards.



**Figure 4.7** Edge linking. The above figures are inputs and outputs of the proposed method. The right stack describes the inputs with the red circles denote the discontinuity of the contours and the left stack the fixed outcomes.



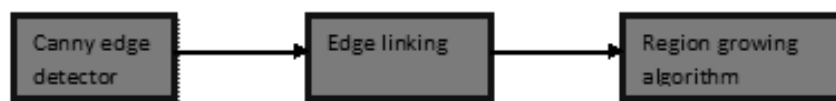
**Figure 4.8** GUI for selecting the first inner aortic seed point.

The common issue that edge detection algorithms have is the incapability of producing closed edges. However this is a very important step to go on with segmentation algorithms and track the edge in need. In our approach we use edge linking algorithms to be sure that the images, which will be processed by any segmentation algorithm, have closed edges. The images in Figure 4.7 are inputs and outputs of the proposed algorithm. The red circle in the input images depicts the discontinuous segments of edges.

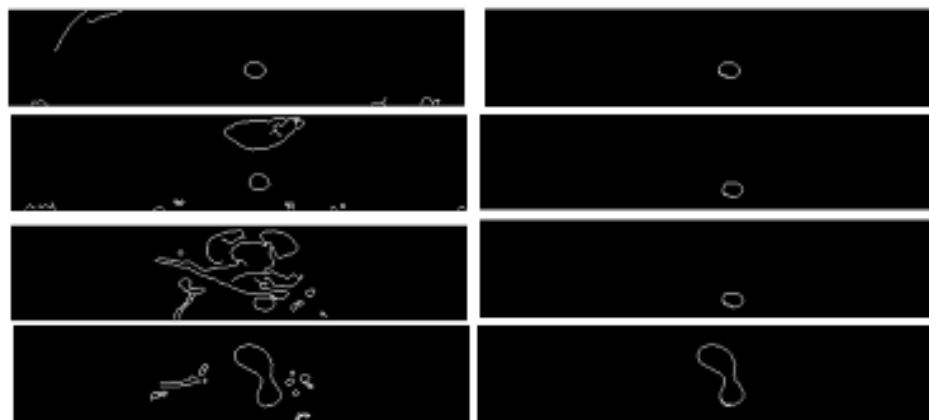
The third and the last step of cross-section procedure is to segment the edges that belong to aortic cross-sections. For this task we decide to create a seeded region growing segmentation algorithm that uses one seed point that belongs to inner area of aortic cross-section and computes the specific edge which refers to aorta. The image in Figure 4.8 shows us the user interaction window for importing the inner seed point to the algorithm. When user specifies with the cursor one point inner the aortic cross-section of primary image algorithm starts and the outcome is a binary image containing the one and only edge of aorta volume. Figure 4.10

shows some inputs and outputs of our algorithm. The Region growing is a simple region-based image segmentation method. It is also classified as a pixel-based image segmentation method since it involves the selection of initial seed points [36]. This approach to segmentation examines neighboring pixels of initial seed points and determines whether the pixel neighbors should be added to the region. Our algorithm takes an initial point as an input and then scans the surrounding area for pixels of the same value. This procedure continues as long as the new pixel's value is same as the old pixel's value. Figure 4.10 depicts the input images and the output of this technique. In the end we create a stack of images containing the aortic boundaries which meets our target's expectations.

So we apply three main procedures for our arteries' boundaries detection. Basic pipeline is shown in Figure 4.9. This procedure is used in every image which has been processed.



**Figure 4.9** The three basic algorithms we use for generating a common pipeline for arteries outer wall segmentation.



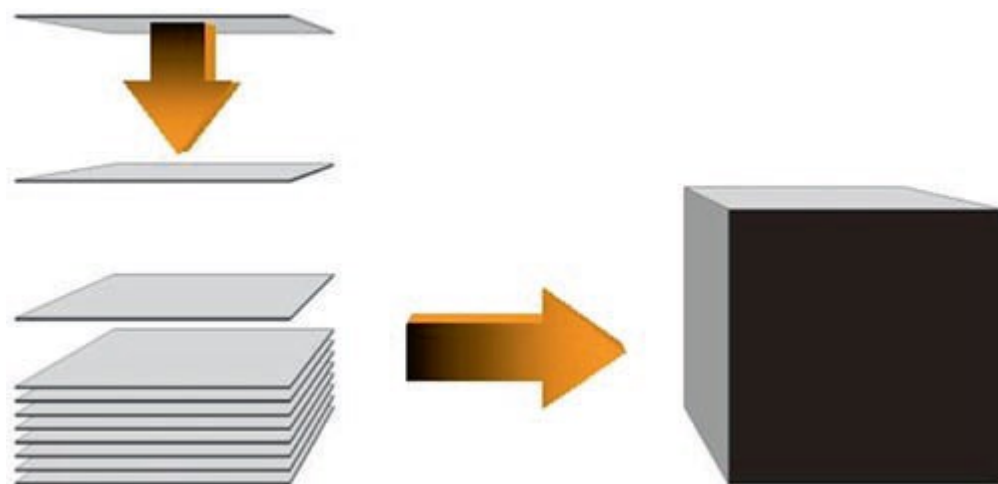
**Figure 4.10** “Seeded region growing algorithm”. The images above are shown how the seeded region growing algorithm works. Inputs exist in the right stack and outputs in the left one. The order of images is sorted from downwards to upwards.

### 4.1.3 Artery tracking (Registration)

#### Image registration

Once we have done with the appropriate formation of images we used image registration techniques to create one 3D solid structure from the dataset of images. Image registration is the process of transforming different images into one coordinate system. This part is vital in order to process our model in space and extract several important information from the dataset. Using image registration techniques we can compose a 3d volume matrix such as the paradigm in Figure 4.11. We align all the coronal slices into one common dimension which is the depth of our space. Provided that every coronal slice is a planar cross-section which moves in the axe of depth in 3D space then we use this attribute to synthesize 3D space form the images' information. So we compute matrix  $f(X,Y,Z)$  in which we store every single coronal slice for every value of variable  $Z$ . In other words  $Z$  represents the amount of coronal slices.  $X$  and  $Y$  represent the pixel resolution of images [37].

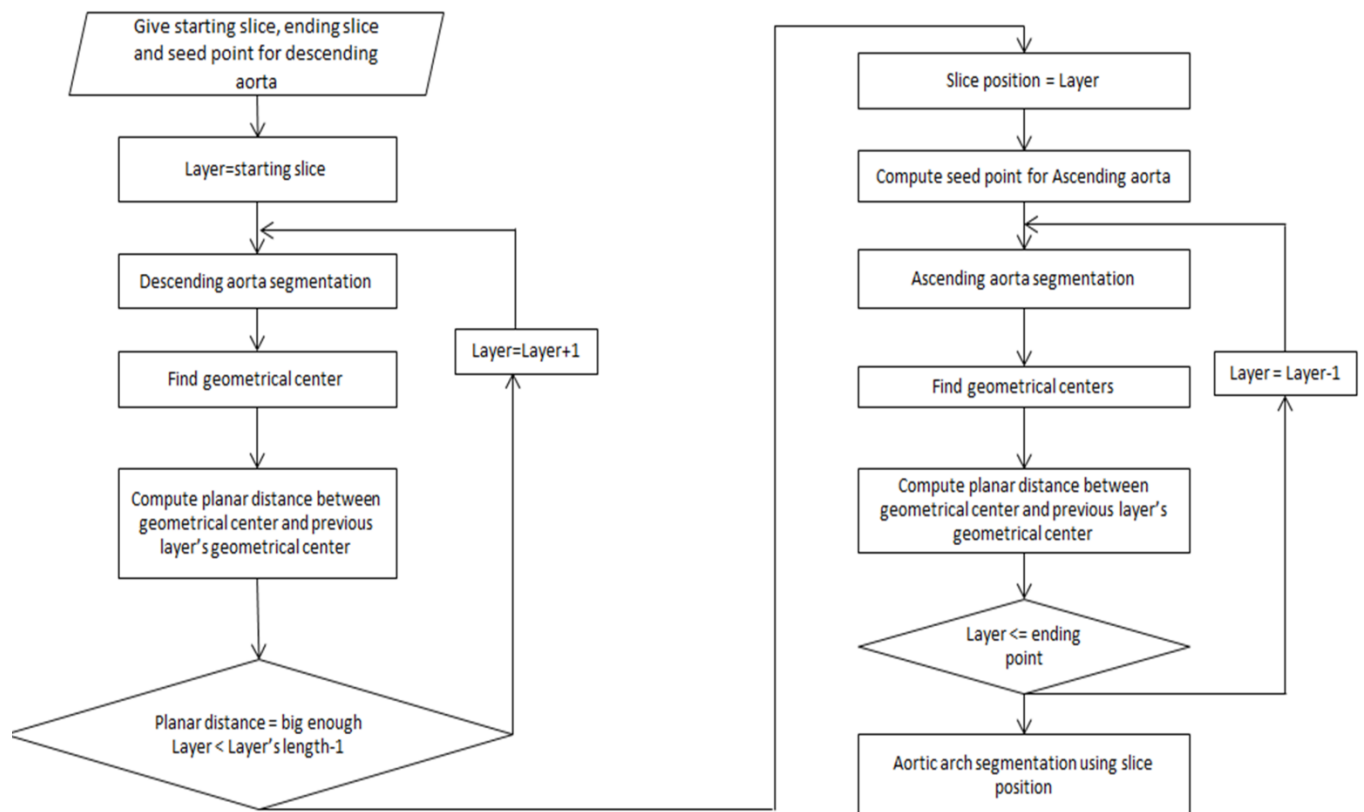
In this way we can synthesize multiple images in 3d space to create a digital object geometrically similar with an existed model. For these operations we shall focus on some specific DICOM image's information such as pixel and voxel spacing, resolution of images and slice's thickness. The outcome of these computations is the exact geometry of a model extracted by consecutives slices over it. The advantage of creating a 3d volume data matrix from medical images is that we can match the information of images and align it in 3D space. This is an important step for calculating and rendering the surface of various human organs. In addition we create other views of slices across the 3D structure such as axial and sagittal slices. We create them transforming the axes of slices that we produce each time form the 3D matrix.



**Figure 4.11** A schematic diagram of image registration technique. **F** denotes the 3D volumetric matrix which has been generated for our processing.

### Semi-Automatic Algorithm

In this stage we will analyze the algorithm we developed for aorta tracking. The procedure is semi-automatic, this means that algorithm needs one inner aortic seed point, the starting and the ending layer to execute all the processing. Our benchmark is axial slices. Axial slices' angle is vertical to 3D space and consequently to the aorta. In Figure 4.12 we present a flowchart of our algorithm. At first we need to examine the morphology of aorta in order to completely understand the generated method. From aortic shape we can conclude that our processing dealing with 3 parts. These three parts are descending, ascending and arch. We separate them because of its dissimilar shape.



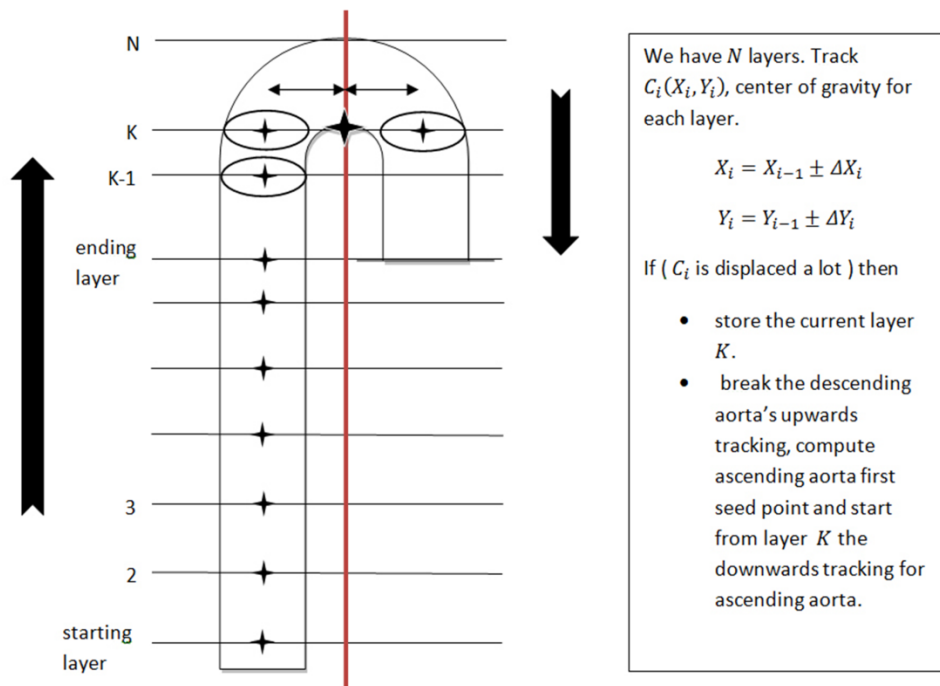
**Figure 4.12** A flowchart diagram of the basic semi-automatic elliptical tracking algorithm. The only inputs that algorithm needs are the starting slice, ending slice and the first seed point for descending aorta.

The main target of this stage is to segment the aortic boundaries from others edges of images. This would be an easy task if the algorithm knew one exact inner aortic point for each image that we process. Examining thoracic aorta's shape from downwards to upwards we can

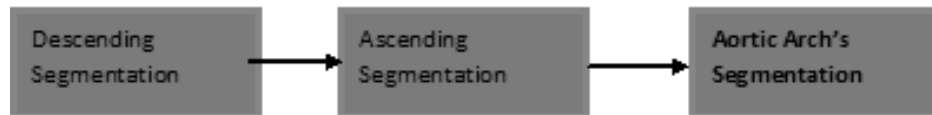


observe that aortic axial contours change position. So in order our effort to be semi-automatic we take advantage of an important property. This property is that consecutives cross-sections of aorta do not change dramatically its positions. Consequently we compute the geometrical center of each cross-section and we know that this point's planar location is certainly an inner aortic point in the next slice we process.

This technique works fine for ascending and descending part of aorta but do not work properly when we pass through the aortic arch. This happens because of the morphology of arch's contours in axial slices. The basic problem is that in this part of aorta we don't deal with elliptical formations, so the centroids that we compute do not satisfy the previous rule. In addition the only seed point that user gives is a seed point from inferior descending aorta layer. For these reasons we developed a method that renews the seed point with the previous aortic cross-section's geometrical center. So when the slices processing goes upwards, the seed point renews for each slice and goes upwards too. The whole procedure describes in Figure 3.14. From the above morphological analysis we decide to split the procedure of semi-automatic segmentation in three parts. The descending's, ascending's and arch's segmentation as shown in the Figure 4.13.



**Figure 4.13** This scheme shows an elliptical tracking algorithm that moves upwards for descending aorta and aortic arch and downwards for ascending aorta. Red line depicts the “aortic axe”.



**Figure 4.14** Three basic segmentation problems.

### **Descending Segmentation**

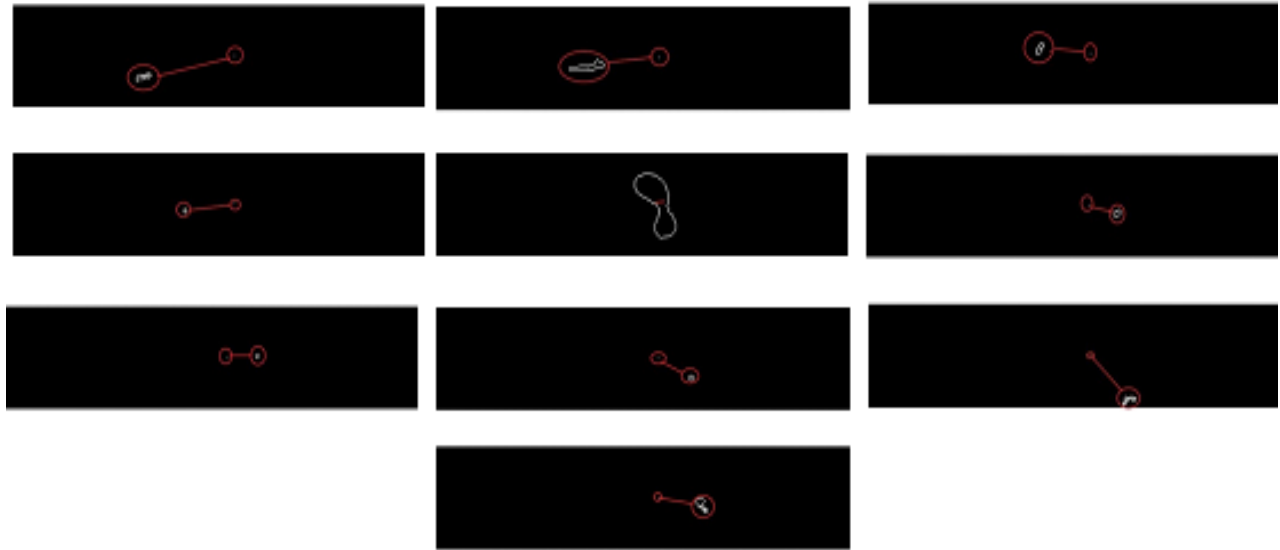
Our algorithm starts with descending segmentation. The slices are processed moving upwards in order to track aortic ellipsoids. The tracking works with a seeded region growing algorithm that we used. In addition algorithm computes and stores the Euclidean distance between consecutive geometrical centers of cross-sections. As we explained before, tracking the descending aorta this Euclidean distance from point to point is approximately equal. When a value of this variable is not approximately equal to the previous value means that algorithm ends with descending aorta segmentation because it finds the K layer which is the first layer that depicts the aortic arch's cross-section. Now the procedure jumps to the ascending segmentation. Figure 4.14 describes this procedure.

### **Ascending Segmentation**

Now, algorithm knowing the K and the ending layer, that user has given, starts the process of ascending aorta's segmentation by moving downwards from K to end. The only thing that needs the whole process is the first inner seed point for the execution of the same region growing algorithm which has been used for descending, too. We developed an effective method for the computation of the ascending aorta's first seed point that we describe below in the same-name subchapter. Finally when this process ends algorithm starts the segmentation of aortic arch.

### **Aortic Arch's Segmentation**

For arch's segmentation we developed a different technique because of morphological dissimilarity. The algorithm goes on from layer K moving upwards for the specific amount of layers that aortic arch's contours exists. This amount of layers is about the 17% of total layers as we compute from several datasets. This number came out from the anatomy of human's body. Instead of using region based segmentation we labeled the contours counting on its discontinuity. Then we compute the red line which is shown in the above figure and transform it to a point in an axial plane. We call that line "aortic axe". We compute each contour's geometrical center and compare it with "aortic axe" point. The closest contour's center to that point describes the right contour. This procedure is repeating for all the layers and has been showed in Figure 4.15.



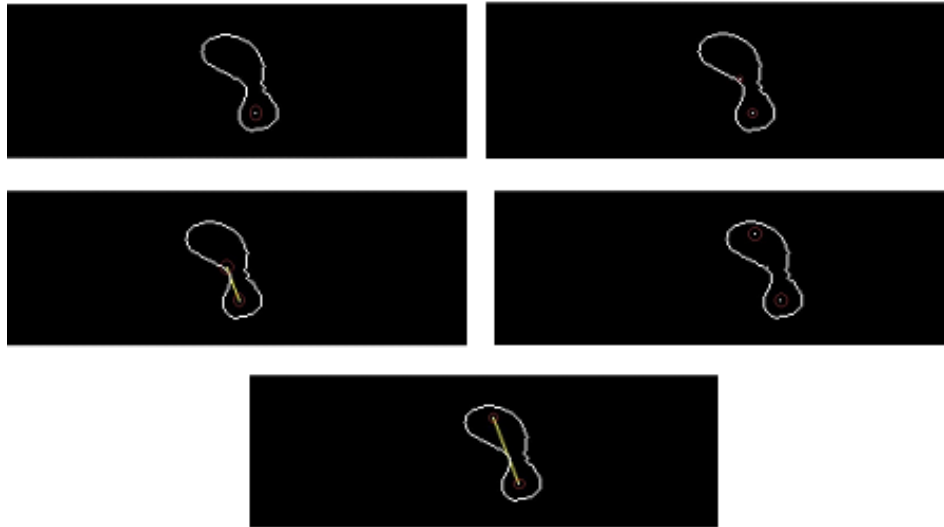
**Figure 4.15** Aortic arch's segmentation. We label and separate the contours from the image shown in figure 21. Then the above images come out. We compare the distances from the "aortic axe" to the centroids of every contour. Finally we keep the contour with the closest distance.



**Figure 4.16** First aortic arch's slice after edge detection and the same slice after the segmentation.

### **Computing ascending aorta's seed point**

When the process comes to the first layer of aortic arch the image that we have to process is shown in the Figure 4.16. This image represents the aortic arch's axial boundaries. We process it and compute the first inner seed point of ascending aorta's perimeter. Our algorithm needs that point in order to segment ascending aorta's boundaries in the same way like descending aorta segmentation works. First we put on this image the geometrical center of previous image. Then we compute the geometrical center of current image. And in this stage we take advantage of aortic symmetry. So we know that if we compute the symmetric point of the previous slice's center from the current slice's geometrical center then the point that we will produce there will be for sure an inner point of ascending aorta's boundaries [5]. The procedure that we described is shown in the Figure 4.17.



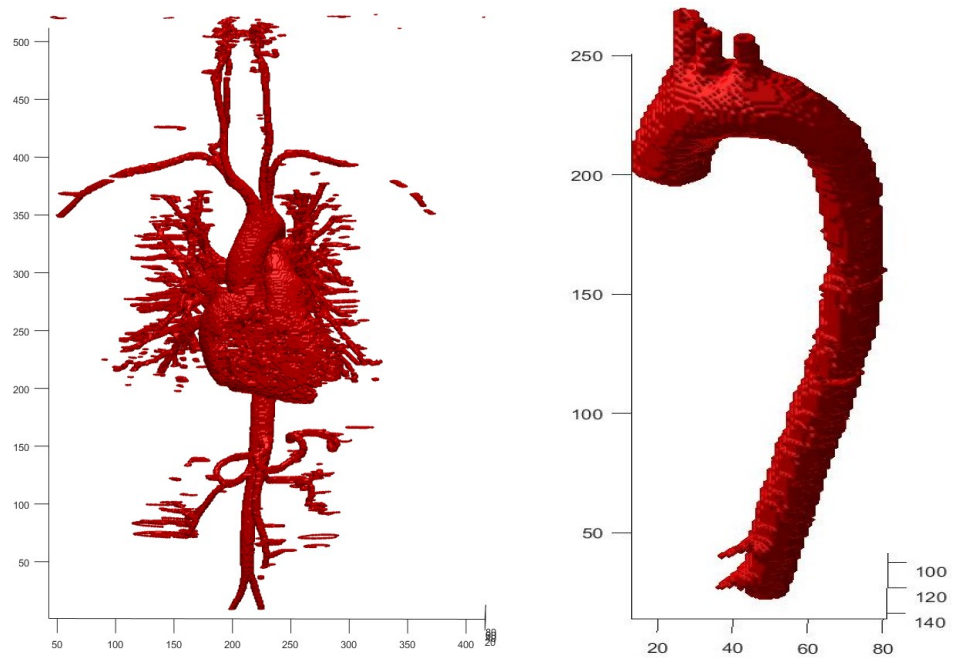
**Figure 4.17** Top-left, contour with previous geometrical center. Top-right, contour with previous and geometrical center. Mid-left, line between these points. Mid-right, computing the symmetric point. Bottom, line between the three points.

#### 4.1.4 Artery model (Reconstruction)

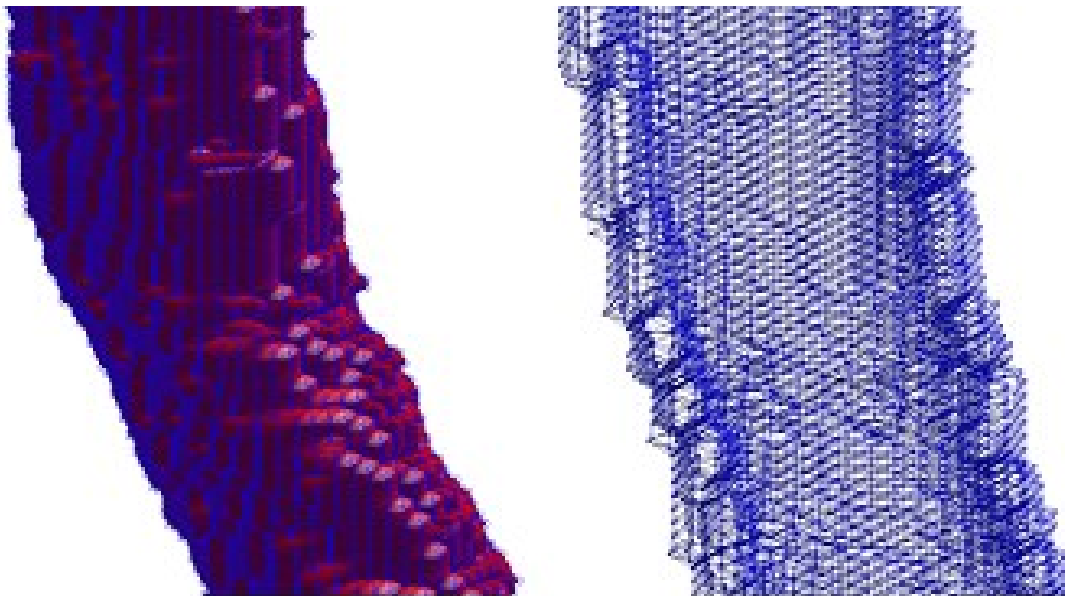
Surface rendering techniques visualize a 3D object as a set of surfaces called iso-surfaces. Each surface has points with the same intensity (called iso-value). It is used when we want to see the separated structures e.g. skull from slices of head, blood vessel system from slices of body etc. Iso-surface refers to a 3D implicit surface defined as  $f(x, y, z) = c$ . So the iso-surface is the set of points for which the function represented by the data taken on a common constant value, that value is the iso-value. So for each iso-value, you get a different iso-surface. This technique is used mostly for high contrast data.

The implementation algorithm we use for visualization is the Marching Cubes algorithm. The algorithm proceeds through the scalar field, taking eight-neighbor locations at a time, then determines the polygons needed to represent the part of the iso-surface that passes through this cube. The individual polygons are then fused into the desired surface.

In our approach in the images that we used to create the 3D volumetric data exists the segmented outer aortic boundaries only, in binary formation [0 1]. In that manner letting iso-value to 1, we can produce the first polygon surface of our model. Figure 4.19 represents the polygonal mesh which has been produced from the marching cubes algorithm for the part of descending aorta. A complete polygonal model of aorta produced by our segmentation proposed method and marching cubes algorithm is shown in Figure 4.18.



**Figure 4.18** Polygonal model of a thresholding segmentation procedure and the single aorta after the proposed segmentation method of aorta.



**Figure 4.19** Polygon model of Aorta generated by iso-contouring.

## 4.2 NURBS fitting techniques

Digital image processing is the preprocessing stage of our work. We are developing these techniques in order to extract the exact point cloud of aorta. Extracting the accurate point cloud is crucial, because there is a need for accurate results which will ensure the accuracy of the final model. The point cloud of aorta is shown in Figure 4.20.

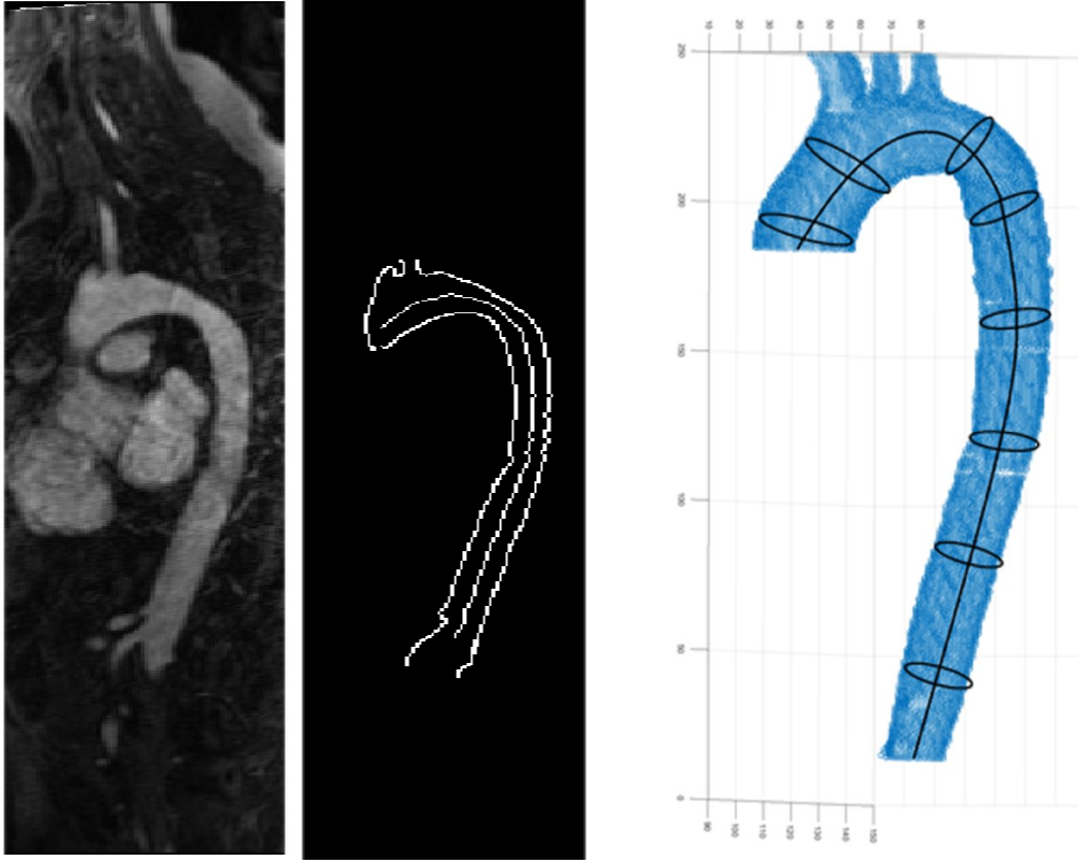
We align all the axial slices into one common dimension in space. We compute a 3D volumetric matrix  $f(X,Y,Z)$  in which we store every single pixel of all axial slices.  $Z$  represents the number of coronal slices, while  $X$  and  $Y$  represent the pixel resolution of the axial images. Transforming our pixel values to points in 3D space, the point cloud is produced. Point cloud is intended to represent the external surface of an object.

Setting up the problem of curve fitting, our target is to transform the crude arrangement of edge points into a smooth curve interpolation in every orthogonal cross section to the aortic centerline. Thus, we developed a pipeline that has four steps. We called first step the “Centerline extraction” the second step “Defining planar cross-sections perpendicular to the centerline”, the third step “NURBS curves fitting method” and the final, fourth step “NURBS surface generation”.

We implemented this differentiation in order to use lofting techniques for generating and rendering the aortic surface. “Lofting” is a drafting technique whereby curved lines are generated to be used in plans for streamlined objects [11], [38]. Lofting approximates a surface from cross-sections along the centerline and is used widely in 3D modeling software packages.

### Centerline extraction

Our first aim is to track the centerline path with a line to apply the NURBS surface fitting method. Centerline is the line across the aorta and the circles along it depicts the pre-selected planar cross-sections. In this section, based on axial slices we compute each aortic cross-section geometric center and construct a cubic B-spline that fits this 3D structure of points. In order to be precise, we use as benchmark sagittal slices, too. We compute the central line of the segmented aorta from a corresponding sagittal slice and matching formation by axial and sagittal slices. In figure 4.20 we can see the original corresponding sagittal slice in the left. In the middle there is the output image of our processing where the central line and the aortic boundaries are depicted. The final aortic centerline, which has been matched from axial and sagittal slices, is shown in same figure at right. Defining the vessel’s path is a very important step to go on. In order to give a smooth curved surface for our object we need to define the path and the perpendicular to it cross-section of our tubular object. We register, on some selected locations across the pre-computed centerline, planes perpendicular it and store the aortic cross-sections on them. Now normal vector of such a cross-section should point in the

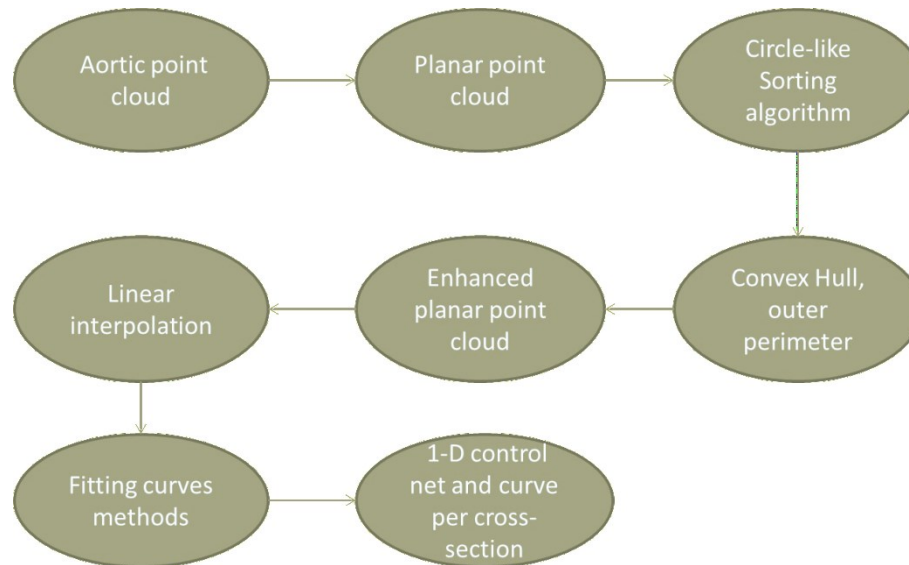


**Figure 4.20** Left: initial sagittal slice. Middle: Boundaries of aorta and the computed central line. Right: Point cloud model of aorta with cross-sections perpendicular to its centerline.

direction tangent to the centerline. This gives the second parametric direction for the NURBS surface representation.

### 4.2.1 2D curves fitting

Computing a curve to approximate data points is a problem encountered frequently in many computer graphics applications. We present a novel and efficient method for computing a planar closed and continuous B-Spline curve. Figure 4.21 below represents the basic procedure that every planar cross-section with data points gets processed by the 2D curve fitting algorithm, in order to generate an interpolant cubic B-Spline curve. Our target is to use these data points in order to compute the control points that describe the B-Spline curves in every pre-selected cross-section.



**Figure 4.21** Basic pipeline for 2D curves fitting.

### **1<sup>st</sup> step (constructing planar point cloud)**

At first we have a point cloud that describes the whole aorta. This point cloud model came of the process of axial slices so points' of the cloud benchmark are the axial slices and as we see in Figure 4.20 these points' positions are of high density in space. This fact is very good for us because it give to us more options to be precise.

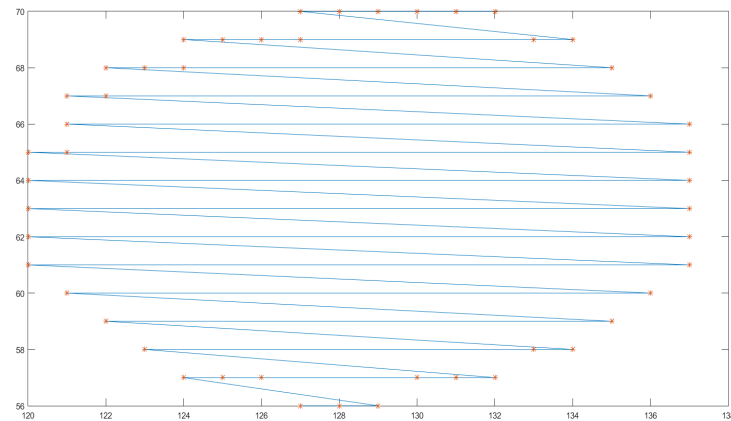
We have to move across the centerline and capture on selected points the ellipsoids that are perpendicular to that centerline. This technique is necessary for generating the NURBS model and giving to it the requested smoothness. We store data points that exist in the same plane in separate matrices. In the end we construct some matrices that contain the geometric information of the cross-sections that describes our tubular shape. In that manner, every matrix consists of points' coordinates (X, Y, Z) and depicts separate planes of the point cloud model. Figure 4.22 represents a specific cross-section of descending's aorta. Data points of aortic outer perimeter are the red points. The blue linear interpolation depicts the sorting order of these points in their matrix.

### **2<sup>nd</sup> step (circle-like sorting)**

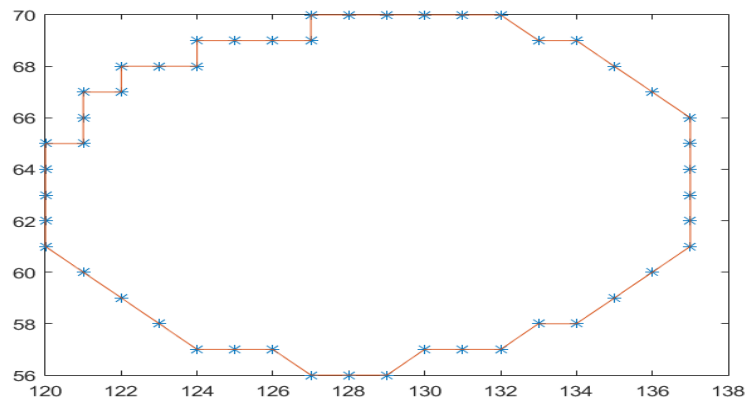
The sorting order that already exists in the matrices is useless because there is not a specific way that consecutives points connected with each other. We need a circle-like connection in order to generate linear interpolation that means something. A graphically circle-like interpolation, in the matrices we got, translates into a circle-like sorting order. So we develop a technique to sort data points in the 2D plane. First of all we compute the mean value of data points' coordinate x and then the mean value of data points' coordinate y. In this way we can



detect the geometrical center of the imaginable circle that data points perform. Then we compute each point's four quadrant inverse tangent. This value represents the direction of each data point in the 2D space related to the circle's center. The four-quadrant inverse tangent ( $\text{atan2}(Y, X)$ ), returns values in the closed interval  $[-\pi, \pi]$  based on the values of  $Y$  and  $X$  as shown Figure 4.22. In this way we construct a direction matrix and every data point has a direction value. Finally we sort the matrix of the data points according to each point's direction value. The outcome of this procedure is shown in Figure 4.23.



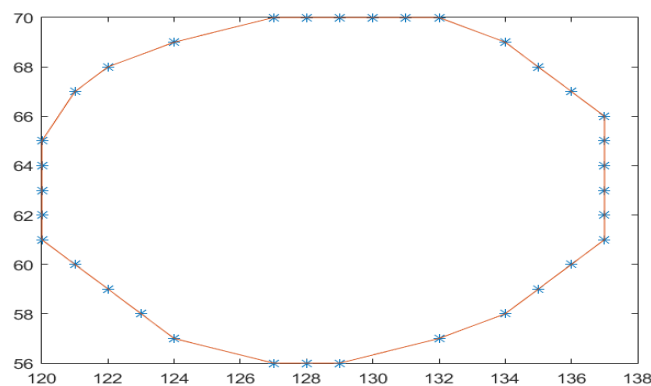
**Figure 4.22** Primary sorting.



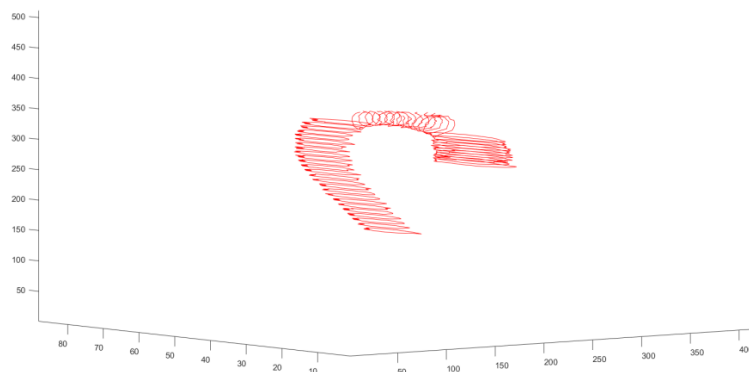
**Figure 4.23** Circle-like sorting.

### **3<sup>rd</sup> step (Convex hull)**

Having stored all the necessary cross-sections, now we have to process these matrices in order to smooth its form. As we see Figure 4.23 the formation of points, we have, is affected by the fact of tessellation. That's the way, cross-sections' formation in many cases is squared. In the 3<sup>rd</sup> step of the geometrical process that we followed is the basic convex hull algorithm. This algorithm returns the convex hull of a set of points in 2-D space. This is accomplished by a procedure that joins with linear parts every data points that exist in the input matrix. So the result of this action is the generation of many triangles. Then algorithm searches for points that do not belong into the area of each triangle that have been formed. The points that satisfy this criterion are the outer points of the point set. The result of the algorithm we used is shown below. The outcome of this procedure is a circle-like formation of only the previous, necessary points. The useless points have been deleted and our shape is smoother as we can see in figure 4.24. So in the end of "2D curves fitting method", we compute all the linear interpolations of the data points per cross-section. In Figure 4.25 we can see the results of that method in 3D space.



**Figure 4.24** Convex hull algorithm. A smoother representation of previous cross-section of Figure 4.22.



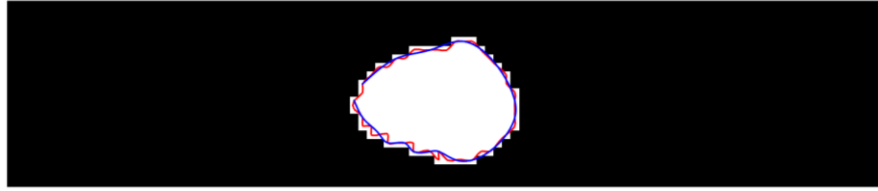
**Figure 4.25** Linear Interpolations of aortic perimeter's data points.

#### **4<sup>th</sup> step (Enhanced planar point cloud)**

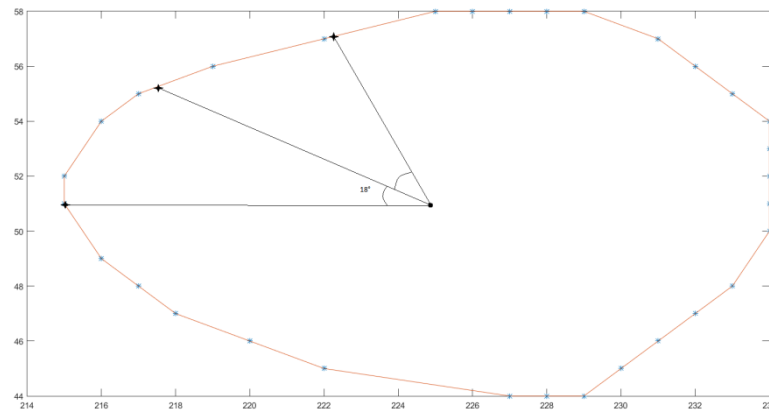
In this step we apply an enhancement technique for each cross-sectional point cloud of aorta. This is a vital part for generating smoothing and optimized NURBS surfaces. NURBS surfaces impose the existence of same amount of control points per cross-section. Furthermore in our case we give an amount of data points of a cross-section as an input and we produce the same number of control points of a curve that is interpolant to the data points. So we need to reduce the amount of every cross-section with a sufficient way to decrease complexity by using lesser data points and to avoid overfitting by finding these points which are optimized and smoothed.

The selected amount of points should represent as precise as possible the aortic geometry. So we applied a novel method for that. In this case the reduction of data points per cross-section is necessary in order to proceed to have same length of data points' matrices per cross-section. We choose 20 points for every cross section and we register each one of them per 18 degrees in the closed linear interpolation. At first we applied a circular spline regression method to the initial data points. This give us a smoothed curve and choose and register specific amount of 20 uniformly distributed points of the generated spline. Finally these points are the new optimized point cloud of our model. This technique gives us lesser and more appropriate points describing each aortic cross-section, furthermore, the harmonization of points in a polar coordinate system and more specifically at each specific axial angle give us a smoother surface, voiding overfitting problems.

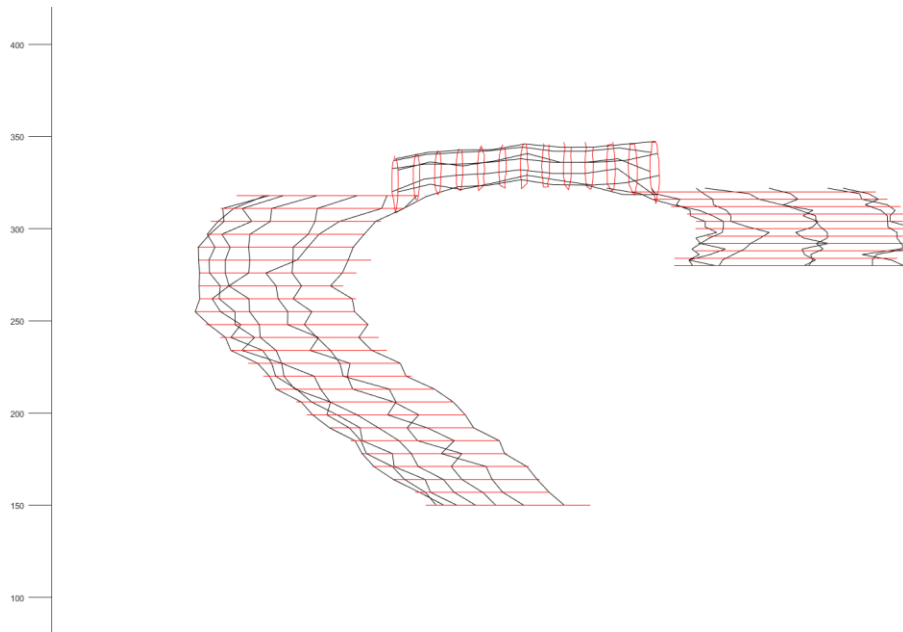
The compression of point cloud that we propose is of great value because it makes our procedure lower-complexity and avoids over-fitting with lesser points. Satisfying results were obtained when cross-section contours were approximated using 20 control points and a B-spline of degree 3 as shown in Figure 4.27. Curves of higher orders generally demand a higher number of control points and produce curves that are prone to loops and wiggles. Results of these methods are shown in Figure 4.26. Red spline is interpolant to initial data points of a cross-section and blue one is interpolant to the new optimized data points. As we see blue spline is smoother and enhanced. The white area is the aortic cross-section area. The great advantage of this method is the selection of the enhanced data points in certain angles across each cross-section of our model. Applying this technique, our target is to align as much as we can the points of each cross section in a vertical connection. Namely, we want the first data point of the first cross-section to be in the closest possible area of the first point of the second cross section and the second point of the first cross section with the second point of the second cross-section respectively. The results of this technique are shown in Figure 4.28 where we interpolate linear the enhanced data points in order to create smooth horizontal and vertical cuts.



**Figure 4.26** White area depicts an aortic cross-section. Red spline is an interpolant to initial data points and blue spline is an interpolant to optimized data points of the same cross-section.



**Figure 4.27** Registration of 20 points per 18 degrees angle in order to avoid over-fitting using lesser points.

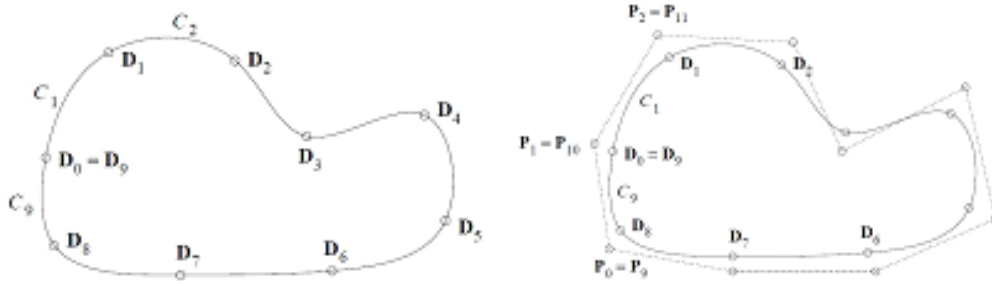


**Figure 4.28** Linear interpolation to enhanced data points of model in order to create smooth horizontal and vertical cuts.

### 5<sup>th</sup> step (Computing control points)

The contour points obtained in previous steps of the algorithm were used to fit a cubic B-spline to each segmented cross-section contour. Given  $n$  data points, when you interpolate, you look for a function that is of some predefined form that has the values in that points exactly as specified. That means given pairs  $(X_i, Y_i)$  you look for  $F$  of some predefined form that satisfies  $F(x_i) = y_i$ .

In this step we use mathematic models to generate smooth, closed interpolating curves using cubic B-Spline curves. Given a set of data points  $D_i = (X_i, Y_i)$ ,  $i = 0, 1, 2, \dots, n$ , ( $n \geq 2$ ) with  $D_0 = D_n$  because our curves should be closed and continuous- $C^2$ .



**Figure 4.29** Closed cubic B-Spline curves with 11(n) control points interpolating 9(n+2) data points.

We notice from the Figure 4.25 above that the closed cubic B-Spline curve has  $n$  segments  $C_1(t), C_2(t), \dots, C_n(t)$  with  $D_{i-1}$  and  $D_i$  being the start and ending points of  $C_i(t)$ . We know such a curve must have  $(n + 3)$  control points, where  $(n + 1)$  is the amount of data points and we need 2 more control points to describe the interpolating curve. So the control points are  $P_0, P_1, \dots, P_{n+2}$ . To guarantee  $C^2$  continuity at  $C_0 = C_n$  control points must satisfy the following conditions:

$$P_n = P_0, P_{n+1} = P_1, P_{n+2} = P_2$$

Such a curve needs  $(n + 7)$  knots, where  $(n + 1)$  is the amount of control points and 3 is the degree of the curve. So the 3 first and the 3 last points of knot value vector must repeat:

$$t_0, t_1, \dots, t_{n+6}$$

To make thing easier, we shall assume that:

$$t_i = i - 3, i = 0, 1, \dots, n + 6$$

Such a cyclic curve can be defined as follows:

$$C(t) = \sum_{i=0}^{n+2} N_{i,3}(t) P_{(i \bmod n)}, t \in [t_3, t_{n+3}] = [0, n]$$

Such that:

$$C(t_{i+3}) = C(i) = \sum_{i=0}^{n+2} N_{i,3}(i) P_{i \bmod n} = D_i \text{ where } i = 0, 1, \dots, n$$

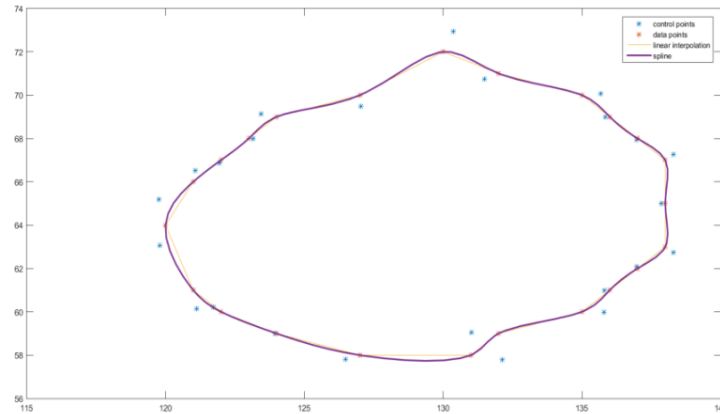
Analyzing the above system linearly:

$$\begin{aligned} \frac{1}{6}P_0 + \frac{2}{3}P_1 + \frac{1}{6}P_2 &= D_0 & \frac{1}{6}P_{n-2} + \frac{2}{3}P_{n-1} + \frac{1}{6}P_n &= D_{n-2} \\ \frac{1}{6}P_1 + \frac{2}{3}P_2 + \frac{1}{6}P_3 &= D_1 & \frac{1}{6}P_{n-1} + \frac{2}{3}P_n + \frac{1}{6}P_{n+1} &= D_{n-1} \\ \dots & & \frac{1}{6}P_n + \frac{2}{3}P_{n+1} + \frac{1}{6}P_{n+2} &= D_n \end{aligned}$$

The last equation is the same as the first equation and, hence, can be ignored. The matrix of the above system is shown below.

$$\begin{bmatrix} 2/3 & 1/6 & & & & \\ 1/6 & 2/3 & 1/6 & & & \\ & 1/6 & 2/3 & 1/6 & & \\ & & & & \ddots & \\ & & & & & \ddots & \\ & & & & 1/6 & 2/3 & 1/6 \\ & & & & & 1/6 & 2/3 & 1/6 \\ 1/6 & & & & & & 1/6 & 2/3 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \vdots \\ \vdots \\ P_{n-1} \\ P_n \end{bmatrix} = \begin{bmatrix} D_0 \\ D_1 \\ D_2 \\ \vdots \\ \vdots \\ D_{n-2} \\ D_{n-1} \end{bmatrix}$$

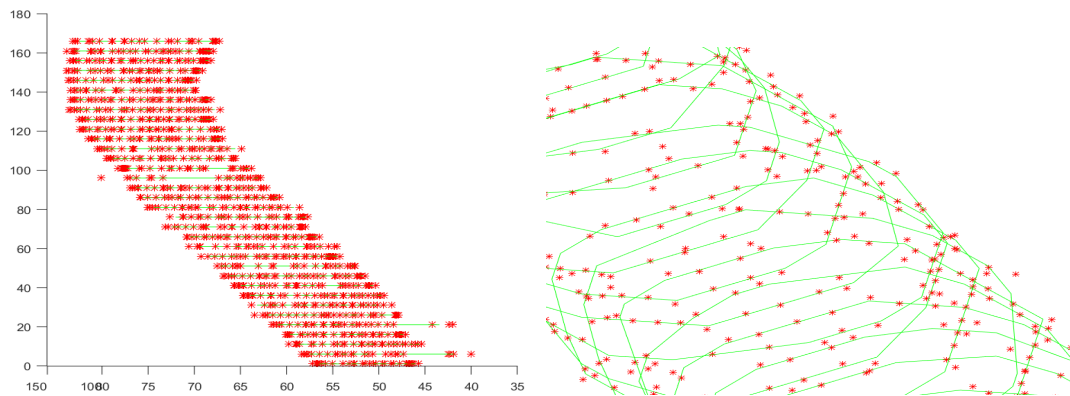
This system of equations can be solved using Gaussian elimination without pivoting as well. We used the above method to generate the control points of the interpolating curve to the specific data points which describes the outer perimeter of aorta. In Figure 4.30 we can see the outcomes of that method. Red colored points represent the data points that belong to the outer aortic perimeter; on the other hand blue points are the computed control points. The purple curve represents a cubic closed and continuous B-spline.



**Figure 4.30** Cubic closed and continuous B-Spline interpolating aortic data points. Blue color points represent curve's control points.

## 4.2.2 3D Surface Generation

Once we have computed the requested cross-sections as we said above we use the “Lofting” method to produce a 3D Nurbs surface from planar cross-sections[5]. Figure 4.31 shows at first all the control points of descending part of aorta and later has been zoomed in one of its areas to see that the control points are not interpolated by the linear interpolation of data points which has to be right because of control points' properties. In the “Lofting” method, a



**Figure 4.31** Control points of descending aorta and a zoomed in area of it.

templated quadrilateral mesh of a closed cubic B-Spline is projected onto each cross-section of the tube, then corresponding vertices in adjacent cross-sections are connected to form a quadrilateral mesh.

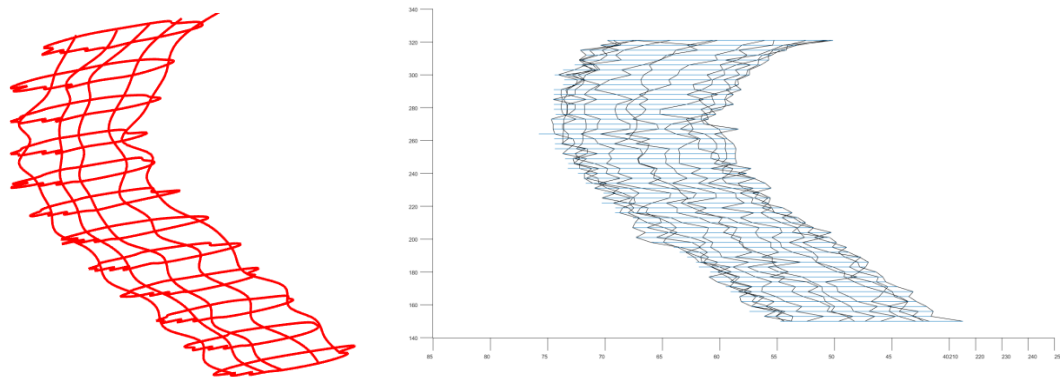
We choose to parameterize the template cross-section as follows. One parametric direction is associated with each closed circular curve, while another parametric direction is associated with a radial coordinate. Rational quadratic basis is used to define the circular curve. Note that

the circular cross-section is unchanged geometrically and parametrically as more control points are chosen for its representation.

In order to produce a smooth surface all approximated cross-section splines have to be made compatible - they need to have an equal degree (3 in our case) and must be defined on the same knot vector which means that we need same amount of control points for each cross-section, as we stated before. Control points of all the cross-section curves (which are illustrated in Figure 4.31) are then taken column after column and a second family of curves is fitted to approximate these columns of control points. This way a second family of B- spline curves is obtained. The two sets of control point vectors form a control net of a tensor surface which forms a quadrilateral control mesh as we can see in Figure 4.32. These types of meshes usually are used for constructing tubular objects like blood vessels [7]. A quadrilateral NURBS control mesh should satisfy the following two requirements:

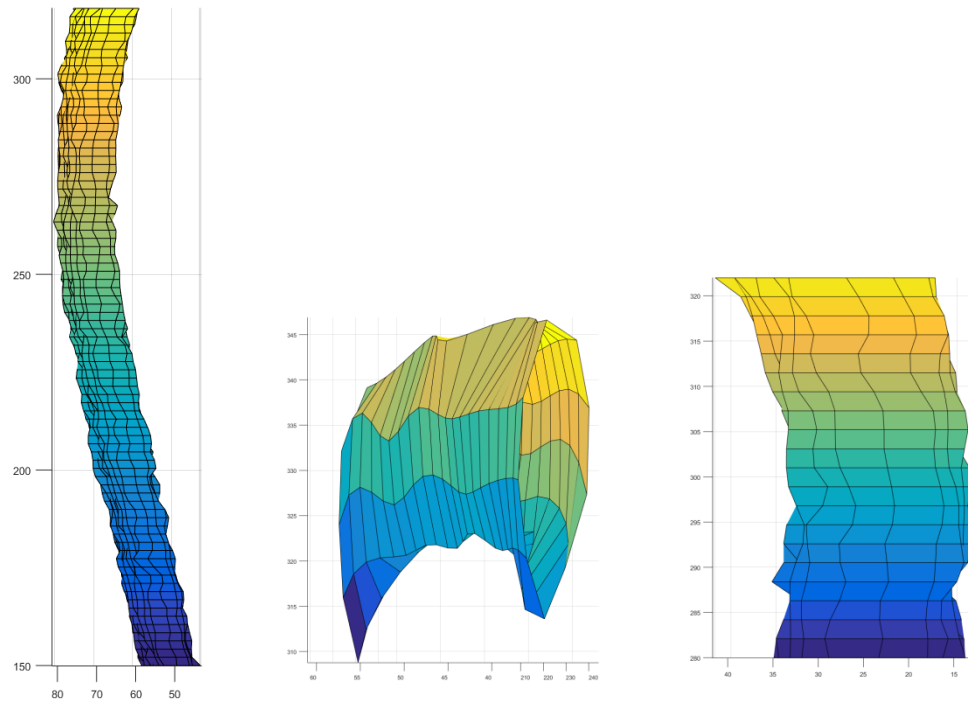
- Any two cross-sections cannot intersect with each other.
- Each cross-section should be perpendicular to the path line.

The control net and 2 knot vectors (one for each family of curves) are enough to completely define the surface. Example of a fragment of the surface of the thoracic descending aorta is shown in the Figure 4.33. We choose 20 and 58 points for U and V directions, respectively.



**Figure 4.32** The requested control net of thoracic descending aorta describing by B-Splines on the left and the same on the right describing with linear segments.





**Figure 4.33** A quadrilateral NURBS surface of thoracic descending aorta.

## 5 Experimental results

In the literature of heart blood vessel segmentation algorithms, most of the algorithms only handle 2D segmentation. Others segment the heart and arteries as a whole in 3D. In common, they do not provide separate segmentation's results of different arteries. Our algorithm is semi-automatic and able to segment and reconstruct the whole thoracic aorta including its three patches, descending, ascending and arch. First of all 2D image segmentation was applied to segment the aortic cross-sections. We used a technique for the identification of aortic arch and the specification of both upwards and downwards tracking of the model and then we create from these techniques the aortic point cloud and its centerline. The centerline of the object has been produced by the centroids of each aortic cross-section using axial slices as benchmark and has been evaluated and matched with the centerline produced by sagittal slices. Processing point cloud, we detect points, which form cross-sections perpendicular to the centerline. Finally, 2D curve fitting was proposed to the cross-sections in order to use them with the centerline and create the respective 2D NURBS model. We have successfully applied our promising method to three different patients' datasets.

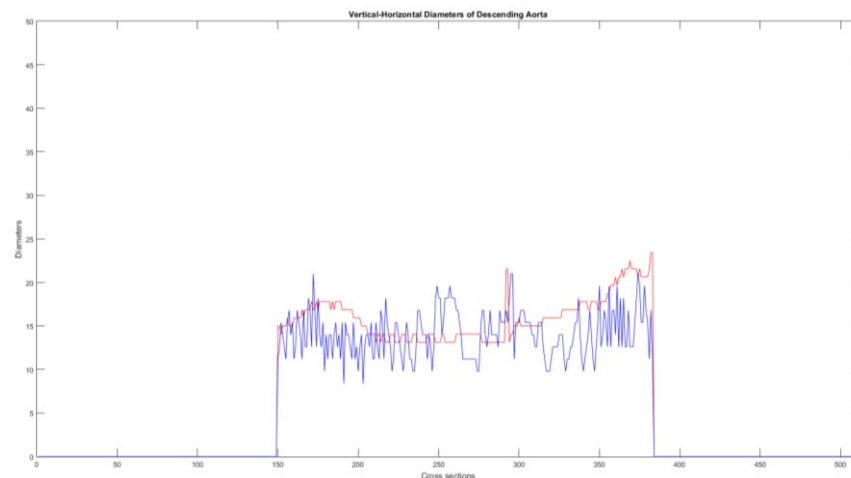
### **The dataset**

The patients, enrolled by the Medical Center "Voiatriki" at Athens, furthermore we managed to equip with additional datasets from open-source libraries from OSIRIX's DICOM image library. We gained our data from MRA process. Although, our semi-automatic method is capable to work with data from other imaging modalities such as CTA or even IVUS (intravascular ultrasound). We evaluate our methods in two different patient's data obtained from Siemens' Magnetom Avanto 1.5 Tesla and one from General Electric's Signa HDtx 1.5 tesla with different resolution level. First and second patient's images' dimensions are 512x416x88 pixels with voxels of size  $0.935 \times 0.935 \text{ mm}^3$  and its slice thickness is 1.4 mm. Third patient's images' dimensions are 512x512x104 pixels with voxels of size  $0.823 \times 0.823 \text{ mm}^3$  and its slice thickness is 2.8 mm.

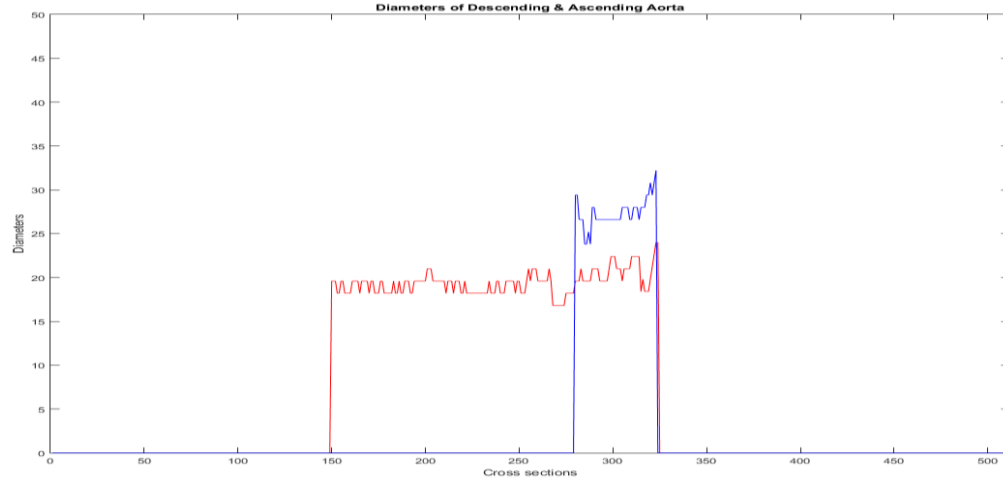
### 5.1 Evaluation of image segmentation

Automatic 3D model-based segmentation of all volunteer datasets was successful and the resulting segmented volume of interest is shown in Figure 5.3. We registered some variables to measure aortic characteristics. First of all as we process every aortic cross-section across the axial slices we registered the two diameters and the geometrical center of each elliptical cross-

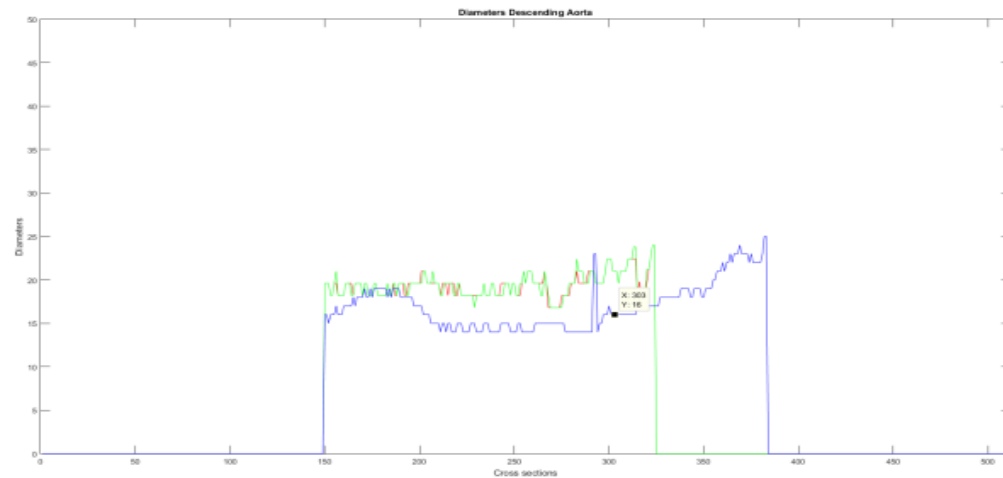
section. We computed two diameters, horizontal and vertical one because of the elliptical form of segmented boundary of aorta. In Figure 5.1 we represent a red and a blue signal for the horizontal and vertical diameter of descending aorta, respectively. These signals are close enough and the relation between them changes all the time (e.g. the bigger diameter of them) making difficult to characterize aorta as a circle or an ellipse by its form, which is definitely right. In Figure 5.2 we see two signals one for each 2 parts of aorta, descending and ascending respectively. The signal depicted at left in Figure 5.2 depicts the diameter of descending aorta with its values to be between 16.8-24 mm and its mean value to stand for 19.41 mm. On the other hand the signal depicted at right in Figure 5.2 depicts the diameter of the ascending aorta with its values between 23.8-32.2 mm and its mean value to stand for 27.39 mm. This fact is confirmed by the principles of the anatomy of aorta as we can see from Figure 1.2, back to the top. So it is confirmed that descending aorta has usually smaller diameters from the ascending one. In figure 5.3 we represent the signals of the descending aorta values from all the three volunteer's data that have been processed. As we see the values of the three different signals are close enough. Signals do not have the same length in x axis because of the different properties of image acquisition.



**Figure 5.1** Capturing the diameters of descending aorta. Present a red and a blue signal for the horizontal and vertical diameter of descending aorta, respectively.



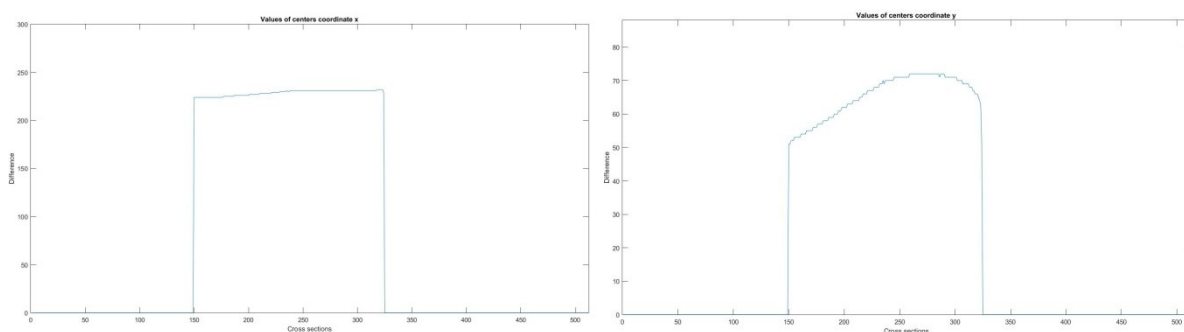
**Figure 5.2** From left to right there is a signal representing the diameter of descending aorta and the diameter of ascending aorta, respectively.



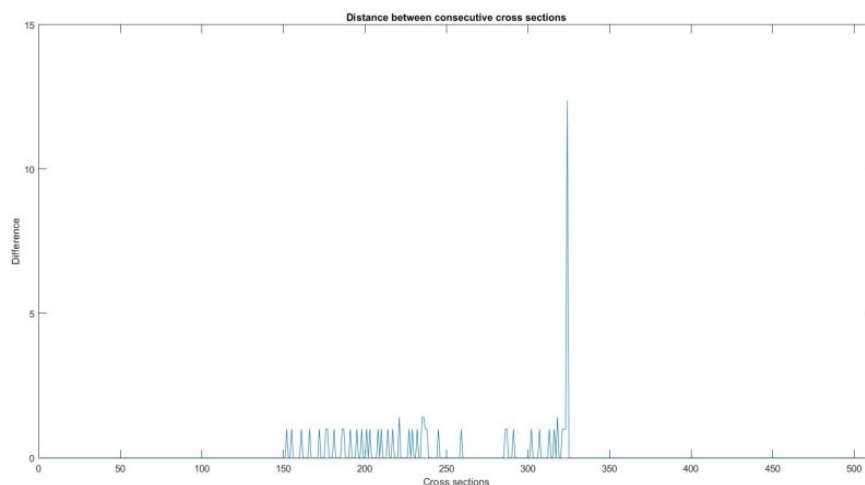
**Figure 5.3** Signals of the descending aorta values from all the three volunteer's data that have been simulated.

Registering the geometrical center of every cross-section of aorta not only aids as with the semi-automatic process, but permits us to make some computations for the evaluation of our methods. In Figure 5.4 at left we can see the tracking of the geometrical center's coordinate  $x$  across the cross-sections. We observe a slightly raise of coordinate's  $x$  value and this in fact happens because of the aortic natural slight slope to the left side. Figure 5.4 at right depicts the signal of the coordinate's  $y$  value. As we see the form of this signal looks alike to the slope that aorta makes from a sagittal view. This event confirms our correct tracking and measurement of aorta.

Figure 5.5 represents the convergence criterion that we use in order to apply the semi-automatic method for the segmentation and reconstruction of aorta. As we compute each cross-section's geometrical center we compute the distance between them and store all these distances. As we see the blue signal is very close to zero and its value does not change dramatically until the process come to a slice between 300<sup>th</sup> and 350<sup>th</sup> where the aortic arch extends. This specific slice is the first axial slice of the aortic arch as we see in Figure 4.16 and in this slice our proposed algorithm is not capable to compute a closed edge for the aorta, so the segmentation fails and so the geometrical center of the image will change dramatically. The results of all this criterion is that the distance value will raise abruptly.

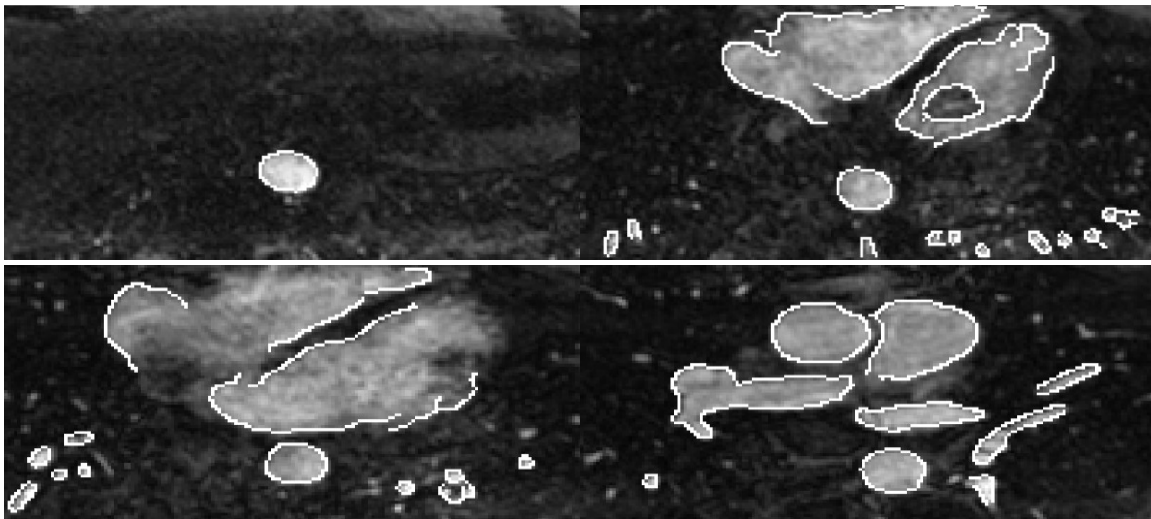


**Figure 5.4** From left to right there is a signal representing the values of geometrical center of coordinate x and the values of geometrical center of coordinate y.

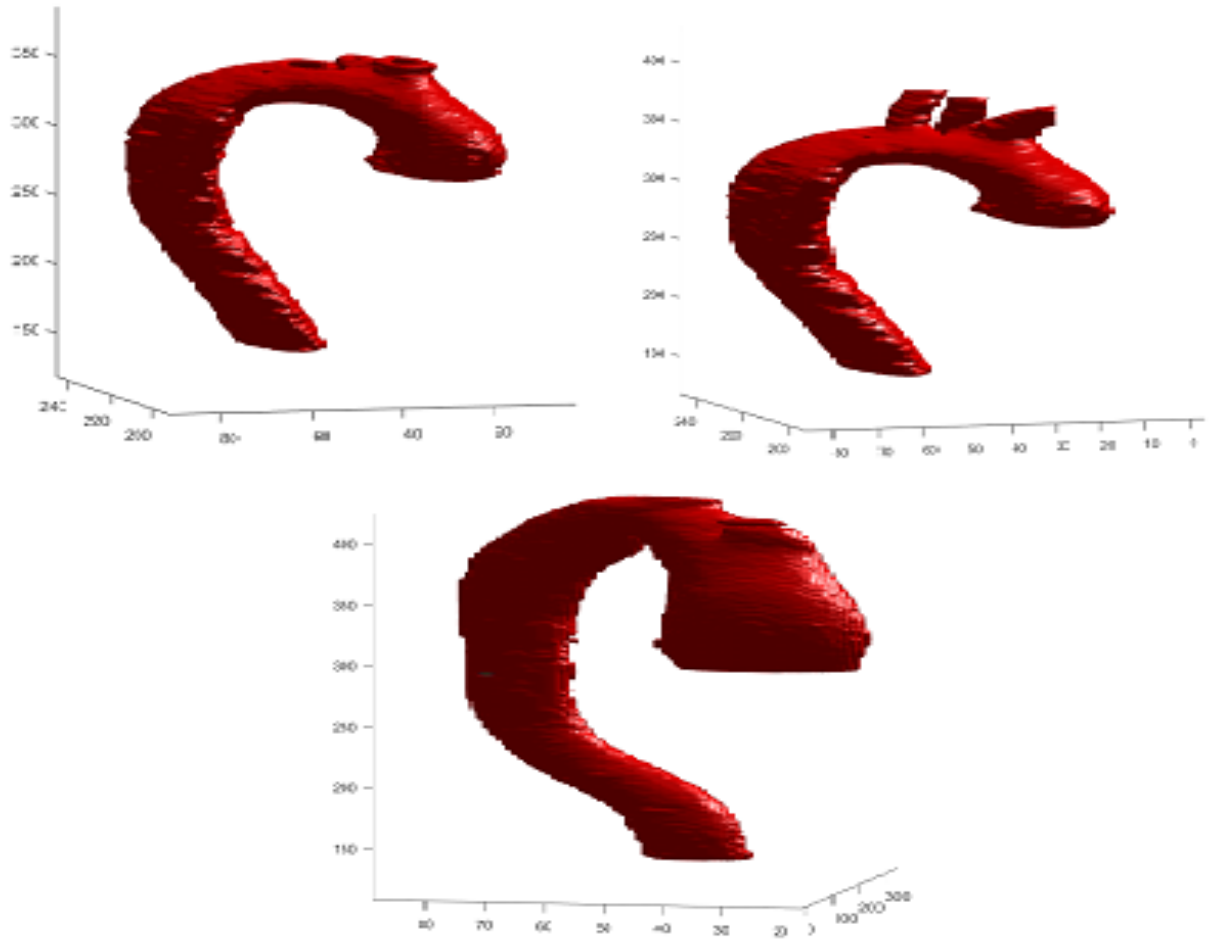


**Figure 5.5** This signal depicts the convergence criterion of our segmentation algorithm.

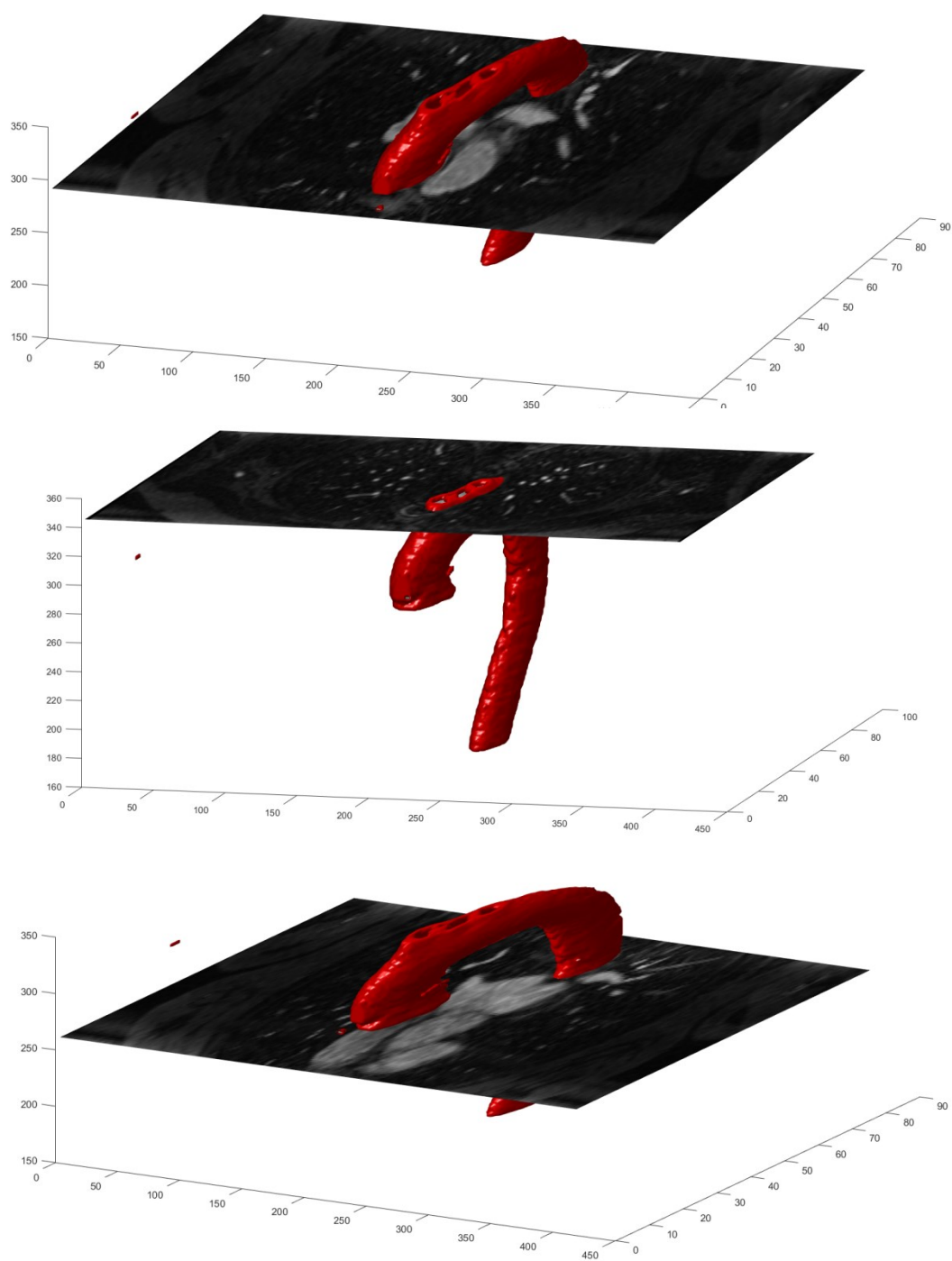
In order to evaluate our semi-automatic method and its functionality we compare its' results with the initial slices. We registered the produced edges and to the initial slices, as we can see from Figure 5.6. Observing the slices, we can see that the location of white pixels is exactly where the intensity of the images changes sharply. We could state that the segmentation outputs are very credible. The only thing that does not seem to be right is the uncontinuous edges in the region of the heart. Despite that fact, we can definitely confirm that aortic edges will be closed because of the edge linking process. Based on the MRI, CT scans and the methods presented above the reconstruction results are illustrated in Figure 5.6. We can back-project the 3D Model in MRI slices in order to evaluate the results. Figure 5.7 shows simulation models intersected with original images. We can conclude that the morphology of simulated model agrees with the characteristics of the organ depicted on the 2D slices .We can see the comparison between polygon models and initial slices at Figure 5.6. In figure 5.9 we evaluate the NURBS model of aorta.



**Figure 5.6** Initial axial slices. We register the edge detection results with white color in these images.

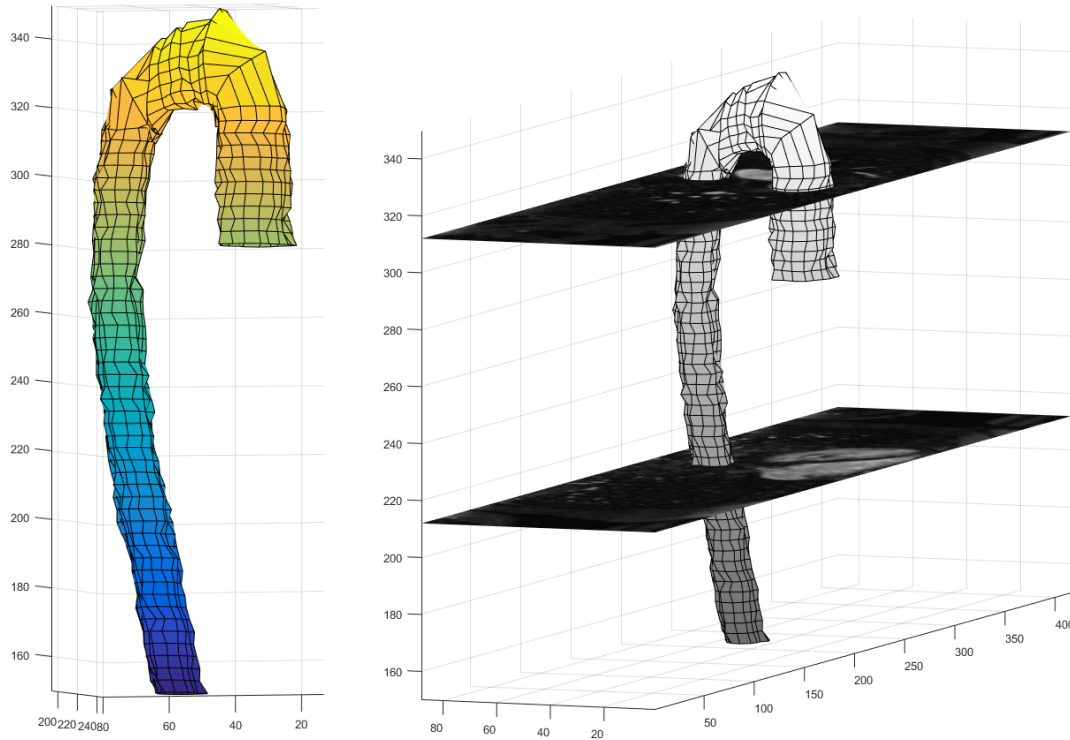


**Figure 5.7** Polygonal model of aorta.Three simulations of diffirent patient’s data input.



**Figure 5.8** Aorta with back-projected initial slices in order to compare and evaluate model.

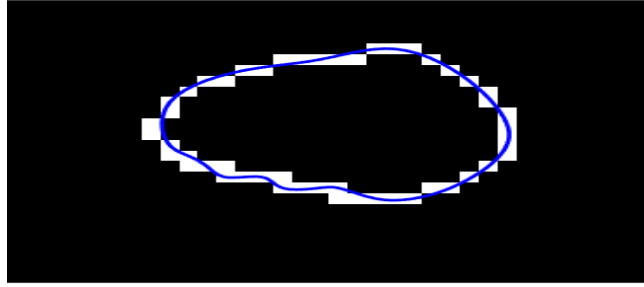




**Figure 5.9** Single NURBS model of aorta and NURBS model back-projected with initial axial slices.

## 5.2 Evaluation of NURBS model

Receiving satisfactory evaluation results for the segmentation method in chapter 5.1, we can proceed to the evaluation of final NURBS model. As presented in previous chapter, our segmentation and, subsequently, polygon mesh results are credible enough in comparison with metrics of values of an average normal aorta. Although this fact cannot confirm the accuracy of our final model because of the additional processing which has been applied in order to generate smooth surfaces from a voxelized model (polygon model). To evaluate our final model we need to compare it with the segmentation results and examine any divergence that could be resulted. In Figure 5.8 we can see with white color the boundary of an aortic cross-section and with blue color the generated cubic B-Spline curve of it. As we can see there is a difference between the voxelized model and the parametric smooth B-spline.



**Figure 5.10** White color the boundary of an aortic cross-section. Blue color is the generated cubic B-Spline curve.

An efficient way to measure those differences is the computation of total area that enclosed to both the voxelized perimeter and the B-Spline curve. Having already proved that the voxelized model is credible enough to the picture and also to the real physical values, we take that as the “ground truth” information of our metrics. The meaning of “ground truth” is something previously validated as true. In machine learning, the term "ground truth" refers to the accuracy of the training set's classification for supervised learning techniques. This is used in statistical models to prove or disprove research hypotheses. The term "ground truthing" refers to the process of gathering the proper objective (provable) data for a test. Therefore, in our case, ground truth represented by the pixels of voxelized perimeter's area of each cross section. Then we track the pixels which are enclosed by the corresponding B-Spline curve and take that as the “approximation truth” of our model. Our target is to compare the “ground truth” and the “approximation truth” measurements with some metrics.

The metrics which we will use is the accuracy, precision and sensitivity. These are the terms which are most commonly associated with a binary classification test and they statistically measure the performance of a test. In a binary classification, we divide a given data set into two categories on the basis of whether they have common properties or not by identifying their significance and in a binary classification test, as the name itself conveys, we deal with two datasets. Sensitivity indicates, in general, how well the test predicts one category. Whereas Accuracy is expected to measure how well the test predicts both categories.

The characteristics of a test that reflects the afore-mentioned abilities are accuracy, sensitivity, precision, positive and negative predictive values and positive and negative likelihood ratios [42-44]. So, in order to get values for accuracy, precision and sensitivity, we need to define some other variables first.

## Definitions

- True positive: the number of pixels that belong to “ground truth” and to the “approximation truth”
- False Positive: the number of pixels that belong to the “approximation truth” and do not belong to the “ground truth”
- False Negative: the number of pixels that belong to the “ground truth” and do not belong to the “approximation truth”

Accuracy: The accuracy of a test is its ability to differentiate the accepted and rejected cases correctly. To estimate the accuracy of a test, we should calculate the proportion of true positive and true negative in all evaluated cases. Mathematically, this can be stated as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}$$

Sensitivity: The sensitivity of a test is its ability to determine the accepted cases correctly. To estimate it, we should calculate the proportion of true positive in patient cases. Mathematically, this can be stated as:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Precision:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Examining those metrics for the slice in Figure 5.1 the “ground truth” value is 217 pixels and the “approximation truth” value is . True Positive factor is 199, which means that 199 pixels out of 217 of “ground truth” matrix exist in the “approximation truth” matrix. False Positive factor is 18, which means that e pixels belong’s to the “approximation truth” and not to the “ground truth”. False Negative factor is 2, which means that 2 pixels belongs to “ground truth” and do not belong to “approximation truth”. Finally the Precision of this test is 0.91, the Sensitivity is 0.99 and the Accuracy is 0.90.

In order to test the whole procedure we computed the metrics for all the slices that belongs to the thoracic descending aorta and summarize them for each patient. We examined our

method thoroughly and applied this metrics in every simulation that has been done. The metrics for our 3 different simulated datasets are shown in table below.

	<b>Sensitivity</b>	<b>Accuracy</b>	<b>Precision</b>
<b>1<sup>st</sup> simulation</b>	0.97	0.86	0.88
<b>2<sup>nd</sup> simulation</b>	0.98	0.91	0.91
<b>3<sup>rd</sup> simulation</b>	0.99	0.89	0.89

Hence, a good binary classification test always results with high values for all the three factors, Sensitivity, Precision and Accuracy, whereas a poor binary classification test results with low values for all. If Sensitivity is high is low then, there is no need to bother about the excellent candidates but the poor candidates must be reexamined to eliminate false positives (poor candidates mistakenly selected). But If Sensitivity is low then, there is no need to bother about the poor candidates but the excellent candidates must be reexamined to eliminate false negatives (excellent candidates mistakenly rejected). An average binary classification test always results with average values which are almost similar for all the factors.

## 6 Conclusions

This thesis presents a novel semi-automatic algorithm for segmentation and NURBS based reconstruction of human's body aorta from MRI or CT DICOM datasets. The methods applied in this work are state of the art in both medical image analysis and computational geometry. Our great achievement is the semi-automatic segmentation of the organ. User has only to import a dataset of DICOM files from an MRA of a CTA image modality and to identify with cursor an inner aortic point on an axial slice of the body. Then our algorithm is capable to segment the boundaries of the aorta and using medical visualization algorithms to reconstruct the segmented volume in 3D space. Furthermore, we worked on methods for converting the geometric grid. As we firstly reconstruct the volume from MRI images using volume rendering and more specific isosurface method later, we work on converting the unstructured grid of polygon model into a structure grid.

First of all the model is based on data points which has been preprocessed in order to create smooth horizontal and vertical cuts. These points form the input of NURBS approximation which enables fast modeling due to the already smoothened surface. We use the point cloud model to fit smooth NURBS surfaces in order to optimize our reconstructed volume and we follow novel techniques in order to create an enhanced control net for our model. A robust MATLAB based software was developed for image based reconstruction and meshing of blood vessels applied with minimal user intervention while allowing for multiple options for customization of biomechanical analyses. Moreover we measure the quality of NURBS surfaces and how to improve the mesh quality need to be studied further. We have successfully applied our method to three patient-specific examples. Quadrilateral solid NURBS surfaces are constructed for finite element analysis of blood flow to be performed.

As part of the future work, we would like to work on biomechanical applications and analysis of the organ. Finite element analysis can be performed in this NURBS model, using two-dimensional shell finite elements and three-dimensional hexahedral finite elements, in order to calculate the displacement, strain and stress field of the aortic walls, define the walls' strength and evaluate the probability of aortic rupture. Unifying this whole procedure it would be of great advance in medical practice because of the need of valid biomedical simulations.

## 7 References

- [1] European Cardiovascular Disease Statistics. 2012.
- [2] Melnick DE (1987) Clinical Simulations – Phgmalion Revisited?, in proceedings of the 1987 SCAMC, pp. 7 - 9.
- [3] Y. Bazilevs, Y. Zhang, V. M. Calo, S. Goswami, C. L. Bajaj, and T. J. R. Hughes, “Isogeometric analysis of blood flow: A NURBS-based approach,” *Proc. Int. Symp. CompIMAGE 2006 - Comput. Model. Objects Represent. Images Fundam. Methods Appl.*, pp. 91–96, 2007.
- [4] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs, “Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement,” *Comput. Methods Appl. Mech. Eng.*, vol. 194, no. 39–41, pp. 4135–4195, 2005.
- [5] M. Gayhart and H. Arisawa, “Automated Segmentation of the Aortic Artery : Evaluation of Images Obtained by Triple Rule-Out Protocol Computer Assisted Diagnosis System Based on Triple Rule-Out Protocol Project,” pp. 2–5, 2011.
- [6] M. E. Zervakis, F. Migliavacca, G. Dubini, E. Dordoni, K. Lambros, and D. I. Fotiadis, “The Effect of the Plaque Composition,” pp. 18–21.
- [7] Y. Zhang, Y. Bazilevs, S. Goswami, C. L. Bajaj, and T. J. R. Hughes, “Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow,” *Proc. 15th Int. Meshing Roundtable, IMR 2006*, vol. 196, pp. 73–92, 2006.
- [8] A. I. Sakellarios *et al.*, “Patient-specific computational modeling of subendothelial LDL accumulation in a stenosed right coronary artery: effect of hemodynamic and biological factors.,” *Am. J. Physiol. Heart Circ. Physiol.*, vol. 304, no. 11, pp. H1455-70, 2013.
- [9] L. Athanasiou *et al.*, “Three-dimensional reconstruction of coronary arteries and plaque morphology using CT angiography – comparison and registration with IVUS,” *BMC Med. Imaging*, vol. 16, no. 1, p. 9, 2016.
- [10] A. I. Sakellarios, G. Rigas, T. P. Exarchos, and D. I. Fotiadis, “A methodology and a software tool for 3D reconstruction of coronary and carotid arteries and atherosclerotic plaques,” 2016.
- [11] K. R. Koch, “Three-dimensional NURBS surface estimated by lofting method,” *Int. J. Adv. Manuf. Technol.*, vol. 49, no. 9–12, pp. 1059–1068, 2010.

- [12] A. Evangelista *et al.*, "Echocardiography in aortic diseases : EAE recommendations for clinical practice," pp. 645–658, 2010.
- [13] F. Auricchio, E. Faggiano, M. Conti, and C. Trentin, "An automatic tool for thoracic aorta segmentation and 3D geometric analysis," 2015.
- [14] P. M. Gruber, E. Hlawka, W. Nobauer, and L. Schmetterer, editors. Johann Radon. Gesammelte Abhandlungen, volume 1. "Osterreichische Akademie der Wissenschaften, Wien," Osterreich, 1987.
- [15] F. Natterer. The Mathematics of Computerized Tomography. JohnWiley & Sons Ltd, 1986.
- [16] "Processing and Visualization of Peripheral CT-Angiography Datasets."
- [17] S. M. Pizer, Adaptive Histogram Equalization and its Variations, Computer Vision, Graphics and Image Processing, Vol. 39, No. 3, 1987, 355-385.
- [18] T. Maintz, "Digital and Medical Image Processing," vol. 10, 2005.
- [19] R. Oak, "A Study of Digital Image Segmentation Techniques," vol. 5, no. 12, pp. 19779–19783, 2016.
- [20] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Third Edition, 2008.
- [21] T. Maintz, "Digital and Medical Image Processing," vol. 10, 2005.
- [22] J. R. Kline, What is the Jordan Curve Theorem? Amer. Math Monthly, 49, no. 5, 281-286.
- [23] D. Marr and E. Hildreth. Theory of edge detection. Proc. of the Royal Society of London. Series B, Biological Sciences., 207(1167):187–217, 1980.
- [24] C. E. Detector, "Canny Edge Detector," pp. 1–12.
- [25] L. Lam, S. Lee, C. Suen, "Thinning methodologies- A comprehensive survey", IEEE transactions on pattern analysis and machine intelligence, 1992
- [26] M. Kaur and P. Goyal, "A Review on Region Based Segmentation," vol. 4, no. 4, pp. 3194–3197, 2015.
- [27] S. J. Babu, "A Survey of Volumetric Visualization Techniques for Medical Images," vol. 2, no. 4, pp. 34–39, 2015.

- [28] R. M. Sherekar and A. Pawar, "A MATLAB image processing approach for reconstruction of DICOM images for manufacturing of customized anatomical implants by using rapid prototyping," *Am. J. Mech. Eng. Autom.*, vol. 1, no. 5, pp. 48–53, 2014.
- [29] William E. Lorensen Harvey E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", *Computer Graphics*, Volume 21, Number 4, July 1987. International
- [30] J. Carr, Thesis, Electrical and Electronic Engineering, University of Canterbury, "Surface Reconstruction 3D Medical Imaging," February, 1996.
- [31] Moustafa Galal Ibrahim, "An automated method of generating NURBS meshes for patient-specific geometries", University of Iowa, 2016.
- [32] J. Cottrell, T. Hughes, Y. Bazilevs, "Isogeometric Analysis- Towards integration of CAD and FEA", Wiley, 2009.
- [33] K. Somasundaram and P. Kalavathi, "MEDICAL IMAGE CONTRAST ENHANCEMENT BASED ON," vol. 3, no. 1, pp. 15–18, 2011.
- [35] L. Piegl and W. Tiller. *The NURBS Book*. Springer, 1996
- [36] Sajadi, Atefeh Sadat, and Seyed Hojat Sabzpoushan. "A New Seeded Region Growing Technique for Retinal Blood Vessels Extraction." *Journal of Medical Signals and Sensors* 4.3 (2014): 223–230. Print.
- [37] Hill DLG, Batchelor PG, Holden M, Hawkes DJ. "Medical image registration". *Phys Med Biol* 2001; 46:R1-R45
- [38] H. Park, "An approximate lofting approach for B-spline surface fitting to functional surfaces," *Int. J. Adv. Manuf. Technol.*, vol. 18, no. 7, pp. 474–482, 2001.
- [39] Roth, S. (1982) Ray Casting for Modeling Solids. *Computer Graphics and Image Processing*, 18, 109-144.
- [40] H. Qin and D. Terzopoulos, "D-NURBS: a physics-based framework for geometric design," *IEEE Trans. Visual. Comput. Graph.* 2, 1 (1996) 85–95.
- [41] N. Avenue, "Nurbs-based segmentation of the brain in medical images," vol. 17, no. 6, pp. 995–1009, 2003.
- [42] Šimundić A-M. Measures of diagnostic accuracy: basic definitions. *Med Biol Sci.* 2008;22(4):61-5.



- [43] Altman D, Bland J. Statistics Notes: Diagnostic tests 1: sensitivity and specificity. *BMJ*. 1994;308(6943):1552.
- [44] Davidson M. The interpretation of diagnostic tests: A primer for physiotherapists. *Aust J Physiother*. 2002;48(3):227-32.\
- [45] J. Maintz and M. Viergever, A survey of medical image registration, *Medical Image Analysis* 2(1998), no. 1, 1–36.
- [46] A. Toga, *Brain warping* , Academic Press, San Diego, 1999.
- [47] J. Hajnal, D.J. Hawkes, D. Hill, and J.V. Hajnal (eds.), *Medical image registration* , CRC Press, 2001.
- [48] J.P.W Pluim and J.M. Fitzpatrick (Editors), *Special issue on image registration* , *IEEE Transactions on Medical Imaging* 22 (2003), no. 11.