Three-Dimensional Modeling of Natural Heterogeneous Objects

By
Miller Gómez Mora

Thesis Supervisor:
Leonardo Flórez Valencia, PhD.

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Faculty of Engineering
Pontificia Universidad Javeriana
Bogotá D.C., Colombia

November 29, 2016

PONTIFICIA UNIVERSIDAD JAVERIANA

STATEMENT OF DISSERTATION APPROVAL

The thesis titled **Three-Dimensional Modeling of Natural Heterogeneous Objects** by Miller Gómez Mora is reviewed and approved by:

Daniel Ricardo Suarez Venegas, PhD.
Pontificia Universidad Javeriana Bogotá-Colombia.

Miguel Enrique Cerrolaza Rivas, PhD.
Universidat Politécnica de Catalunya, Spain.

Fabio Augusto Gonzalez Osorio, PhD.
Universidad Nacional de Colombia.

Bogotá D.C., Colombia, October 14, 2016

Contents

List of Figures

List of Tables

Abstract

Most natural objects contain multiple regions within complex volumetric structures. These structures are usually organized in complicated geometric configurations. Such objects are referred to as natural heterogeneous objects or simply heterogeneous objects. In medicine and other related fields when an natural heterogeneous object is studied, computed tomography exposures are taken through several parallel slices. These slices are then stacked in volume data and reconstructed into three-dimensional computer models in order to study the object structure and functionality.

To be able to successfully build three-dimensional computer models of natural heterogeneous objects, accurate identification and extraction of all regions comprising the natural heterogeneous object is important. However, building three-dimensional computer models of natural heterogeneous objects based on medical images is still a challenging problem and pose two inherent issues related to the inaccuracies which arise from and are inherent to the data acquisition process.

The first issue is the appearance of aliasing artifacts in the boundary between regions, a common issue in mesh generation from medical images, also known as stair-stepped artifacts. These jagged edges, caused by image voxelisation, are visually unappealing and are unsuited for engineering applications. A computer-aided reverse-engineering method (*CAREM*) for generating consistent smooth models for specific regions based on medical images is proposed to address this issue. The goal of this method is to construct three-dimensional computer models from a given three-dimensional domain (medical image) representing anatomical regions or organs.

The second issue is the extraction of smooth three-dimensional multi-region meshes that conform to the region boundaries of natural heterogeneous objects described

in medical images. To solve this issue, a region-aware method *(RAM)* for natural object modeling is proposed. This method uses a vector implicit function structure for organizing all regions comprising the natural heterogeneous object. The goal of this method is to generate three-dimensional computer models with smooth interfaces conforming to different regions of the natural heterogeneous object needed for many computer simulation applications.

Although constructing consistent and coherent three-dimensional computer models is far more from a trivial task specially for natural heterogeneous objects, experimental results demonstrate that the proposed *CAREM* method and *RAM* method are effective and powerful tools to model natural heterogeneous objects as well as constituent regions. Furthermore, the scheme of representation used in this dissertation reduces the amount of information needed to represent volume data.

Hence, this dissertation contributes to the framework of heterogeneous object modeling, as well as to the popular and intricate study of implicit object representation, however, the emphasis of this research is placed on accuracy and shape fidelity needed for biomedical applications.

Dedication

With all my love to my son Miller Augusto Gómez Flórez.

Bogotá D.C., Octuber 2016

Acknowledgments

I would like to thank a lot of people who supported and helped me with my research which led to this dissertation. First of all, I want to thank my research advisor Leonardo Flórez Valencia for his good advices throughout all stages of my research. I am grateful to various colleagues, who worked with me or gave me some advices during my research: Wilmar Hernandez and Cesar Bustacara. I am very grateful to the Universidad Distrital Francisco José de Caldas for financially supporting my studies which made it all possible for me. My biggest thanks are of course for my Parents Maria Maura Mora and Ignacio Gómez, and all my family for their support and encouragement.

Part I

Preliminaries

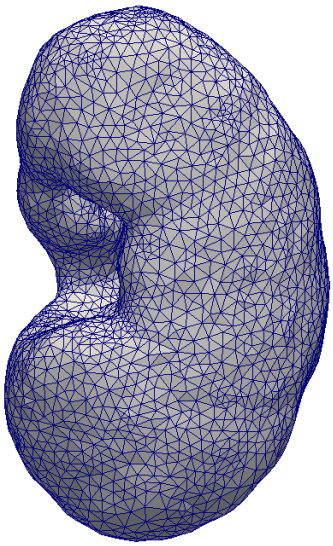
Chapter 1

Introduction

MODELING natural heterogeneous objects, their properties and internal relations is an important research topic in computer vision and computer graphics with broad application in different engineering fields. The process of modeling natural objects normally includes a mathematical representation and a discrete computer-oriented modeling. In computer graphics, for instance, most objects are represented as manifold surfaces embedded in three-dimensional Euclidean space. The result of such a modeling process is a three-dimensional computer model called surface model that can be displayed as two-dimensional images through a rendering process, or be used in a computer simulations of physical phenomena. Thus, the aim of development such computer models is to provide a set of tools through which the properties of represented objects can be studied in a manner which will facilitate simulating the real object manipulation using the computational model.

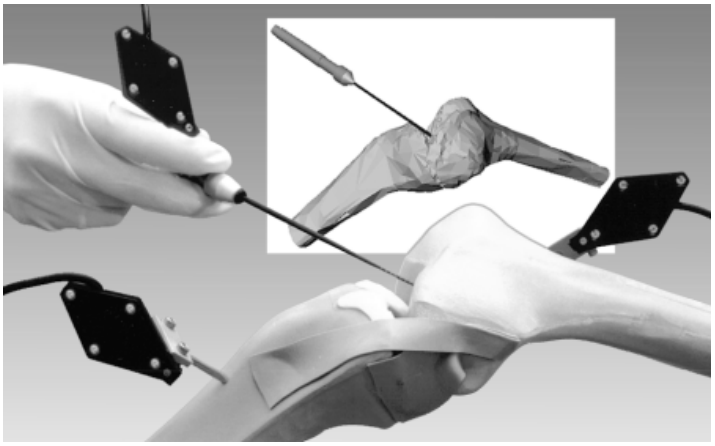
Although in many engineering fields surface modeling in general and finite-element modeling in particular are powerful tools of understanding natural objects, in other fields, such as manufacturing, medicine, virtual reality or biomedical engineering, are not enough for several applications where information about the interior of an object is necessary and the generation of volume models is required. For instance, computer analysis and simulation using volume models can be performed in an effort to predict and improve the performance of manufacturing products before the designs are finalized and built (bones, artery, organs, etc) (see Figure 1.1a).

Volume models are also used to simulate effects of a surgical interventions (see Figure 1.1c). Thus, volume models of body parts could be used in surgical planning laboratories as a studying tool before physicians go through with surgeries.



(a) Modeling and visualization of a human kidney.

(b) Manufacturing of accessories and patient specific prosthesis.



(c) Surgical simulation and planning.

Figure 1.1: Biomedical applications of volume models.

Another example, from biomedical engineering is to simulate and evaluate the effects of limb replacement without large number of physical real trials (see Fig-

ure 1.1b). Once volume models of anatomical structures are available, they can enable physicians accurately plan the prosthetic manufacturing and increase the patient comfort.

The idea of modeling anatomical organs and structures for engineering simulation purposes is attractive for several reasons: First, by supplying a base region material, the locations upon which tissues will integrate with the surrounding environment are known a priori; Secondly, by controlling the shape, or at least the surface contour, one can manipulate the mechanical environment in which tissue exists; Finally, with control over the anatomical domain comes control over the interfaces between organs. However, the construction of finite element models for the purpose of analysis is in itself a laborious task especially for biological objects (e.g. fruits, human body parts etc.) which often have a more complicated geometry than mechanical components.

This chapter is organized as fallow: the motivation behind this dissertation and the research objectives are presented in Section 1.1 and Section 1.2 respectively. Section 1.3 presents the document organization.

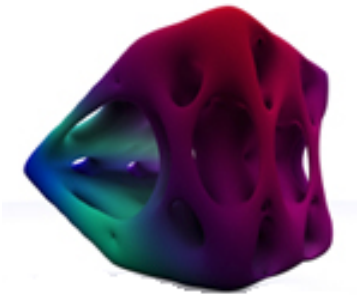
1.1 Motivation

The term *natural heterogeneous object* is defined such that it is a tangible and visible entity occurring naturally and consisting (composed) of different elements or parts. There are three classes of natural heterogeneous objects (Kumar and Dutta, 1998) which are shown in Figure 1.2:

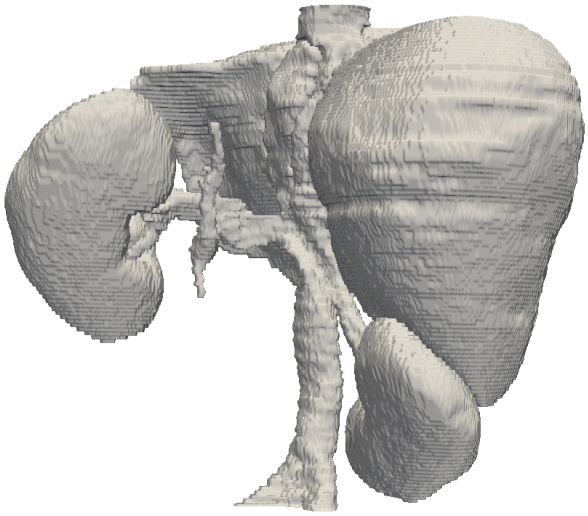
- Multi-material natural objects,
- Natural objects with embedded/nested regions,
- Natural objects without clear region boundary.

Most of the existing heterogeneous object modeling approaches are designed to model either complex geometries with simple material distributions, that is in fact a solid modeling approach, or compound material distributions with simple (regular) geometries, that is a functionally graded material modeling approach.

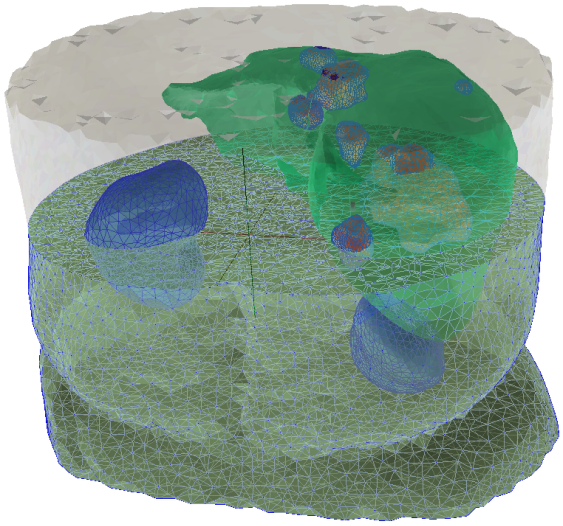
In both cases geometry makes reference to objects with a single region avoiding modeling multi-material objects and objects with embedded regions. Even so, modeling these heterogeneous objects has attracted enormous attentions in the last time (Wang et al., 2011), (d'Otreppe et al., 2012), (Yuan et al., 2012), (Ohtake and Suzuki, 2013).



(a) An object without clear region boundary.



(b) A multi-region object.



(c) An object with region embedded in it.

Figure 1.2: Examples of natural heterogeneous objects.

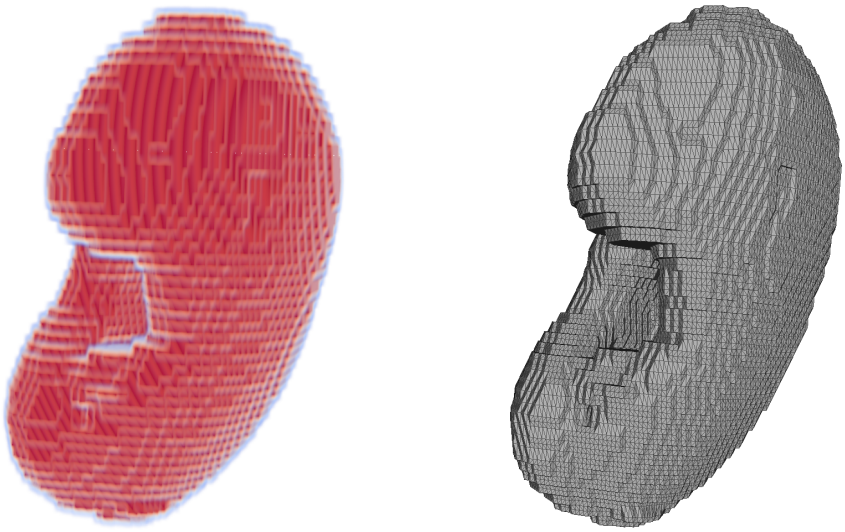
The main goal of this dissertation is to obtain three-dimensional computer models of any natural heterogeneous object (live or inanimate), However, the efforts have been concentrated on obtaining three-dimensional models of anatomical organs and structures. Human anatomy and biological structures consist of a variety of tissues, organs and structures which are either multi-material natural objects (see Figure 1.2b), or natural objects with embedded regions (see Figure 1.2c). These objects are called multi-region natural objects to distinguish them from natural objects composed of only one region (see Figure 1.2a).

Nowadays, medical imaging modalities are being extensively used in fields, such as medicine, virtual reality, and biomedical engineering, for generating computer models of natural objects, including their interior (Stytz et al., 1991). The concept of generating three-dimensional computer models from a stack of 2D images was first published by Jerry Fram in 1980 (Fram and Deutsch, 1975) and since then the idea of reproducing human anatomy with a computational model has become of great research interest.

Although three-dimensional computer modeling can save resources (time and money), especially when compared with benchmark testing or experimentations, the extraction of volume models from medical imaging modalities is a nontrivial problem and pose two inherent issues related to reconstruction errors and low sampling rates of the acquired dataset (Vivodtzev et al., 2003).

The first issue is the appearance of aliasing artifacts in the mesh generation process from medical images, also known as stair-stepped artifacts (see Figure 1.3). These meshes generally have jagged edges caused by image voxelisation, that are visually unappealing and unsuited for finite element simulations (see Figure 1.3b).

The second issue is the extraction of smooth three-dimensional multi-region meshes that conform to the region boundaries described in the medical images (see Figure 1.4). Regardless of the medical imaging modality used, multiple intensity regions are typically present within such images. It is crucial to accurately identify and extract these regions in order to successfully build volume models of natural heterogeneous objects (see Figure 1.4a).



(a) Aliasing artifacts caused by image voxelisation.

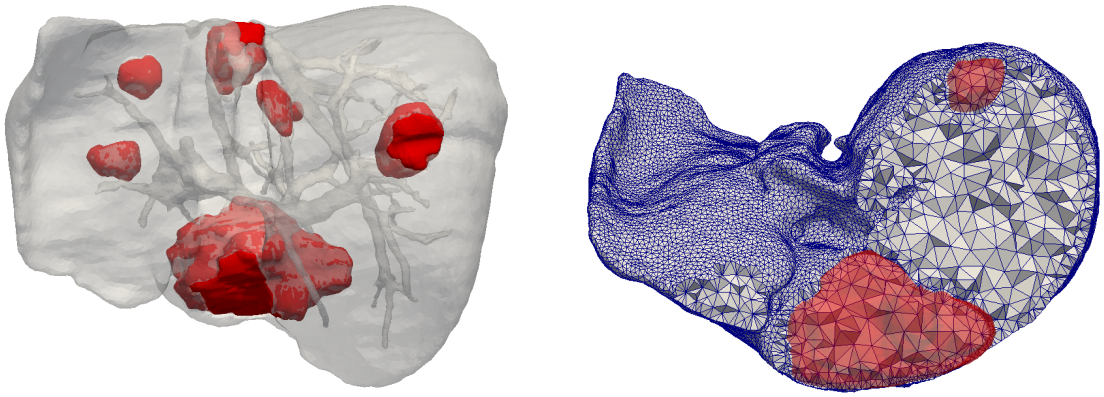
(b) Stair-stepped artifacts present in mesh generation.

Figure 1.3: Appearance of stair-stepped artifacts during mesh generation from medical images.

Additionally, when building three-dimensional computer models it is also important to distinguish between computer modeling for visualization purposes and computer modeling for biomedical simulation purposes because they have different goals, even if the global framework is the same. Three-dimensional computer models for biomedical simulation purposes have stronger requirements: the final model should be valid for computer aided design (CAD) simulations and undergo further operations (see Figure 1.4b). From this point of view the major bottleneck in the complete modeling–analysis cycle is the definition of the geometry of the object under consideration.

Such a serious issues overshadow a lot of potential modeling and analysis applications. This makes necessary in this dissertation to explore new modeling methods in order to build three-dimensional computer models of natural heterogeneous objects such as anatomical organs and structures. The availability of such modeling methods remains central to three-dimensional computer modeling, analysis and fabrication of such objects. The goal is to obtain three-dimensional computer models that can be efficiently used not only to visualize internal structures of heterogeneous objects (Sun et al., 2004), but also to generate three-dimensional bioCAD models for realistic physically-based simulations

(Zaidi and Tsui, 2009).



(a) Identification of a human liver with some tumors embedded in it.

(b) 3D multi-region mesh conforming to the region boundaries described in Figure 1.4a.

Figure 1.4: Extraction of smooth 3D multi-region mesh from medical images.

Three-dimensional bioCAD modeling is a new and exciting interdisciplinary field that promises to combine the innovations in medical imaging and well tested and proven reliable tools from engineering design. Indeed, three-dimensional bioCAD models are prerequisites of the downstream applications in Computer-aided engineering (CAE) analysis and Computer-aided manufacturing (CAM) fabrications of natural heterogeneous objects. This has shown to have promise in practical applications from bio-engineering and medical research to computer animation, and multi-region modeling for fabrication.

Most recent advances in heterogeneous object modeling focus mainly on manufacturing and solid free-form fabrication processes with the purpose of designing and manufacturing of solids with varying material properties (Schroeder et al., 2005) (see Figure 1.2a). However, modeling natural heterogeneous objects, such as anatomical organs and structures, differs vastly from the design of heterogeneous engineering components due to the fact that such objects tend to have more complicated geometries than man-made structures and display a vast variation of material properties within the regions (Moustakides et al., 2000). For instance, liver-tumors must be modeled together with and within the liver (see Figure 1.2c). Such three-dimensional computer models can help us gain a better understanding of natural heterogeneous objects in general and of the human anatomy in particular.

To address this problem, the research methodology is mainly focused on the following research questions:

- *What mathematical representation can accurately and compactly represent heterogeneous objects composed of multiple regions?*
- *How to develop a practical modeling method for generating accurate three-dimensional computer models of specific regions composing natural heterogeneous objects?*
- *How to create a region-aware method for three-dimensional computer modeling of natural heterogeneous objects that maintains information of all region interfaces?*

Hence, the hypothesis that drives this dissertation is:

- ***A region-aware modeling method based on an implicit representation allows to generate accurate geometric models of natural heterogeneous objects composed of multiple regions.***

1.2 Research objectives

This research focuses on three-dimensional computer modeling of natural heterogeneous objects from which three-dimensional bioCAD models can be generated. The main objective of this work is:

To develop a general method for three-dimensional modeling of natural heterogeneous objects using a systemic approach and a compact and consistent representation

In this dissertation, a three-dimensional computer modeling approach based on reverse engineering is presented. This approach aims at creating smooth three-dimensional computer models that can be efficiently used for visualization and simulation purposes. The emphasis of such approach is placed on accuracy, in terms of geometric quality and low volumetric error, and shape fidelity needed for biomedical applications.

Additionally, this research involves the following specific objectives:

- *The analysis of state-of-the-art representations that can accurately and compactly represent natural objects composed of multiple regions, in order to identify the advantages and disadvantages of these representations.*

Modeling natural heterogeneous objects is an open problem seldom addressed in the past. To accomplish such a goal, it is important to review related works on existing natural object representations and relevant modeling approaches.

- *The development of a three-dimensional modeling method based on reverse engineering for generating accurate three-dimensional computer models from medical images that are topologically correct and geometrically smooth.*

The continuous development in the field of medical imaging modalities, such as computer tomography (CT) and magnetic resonance imaging (MRI), along with reverse engineering techniques have made computer aided technologies to be used extensively in modern medicine and biomedical engineering (Prince and Links, 2006). Medical images can be considered as special cases of implicit geometry representations, since boundaries of regions in the image are not available in an explicit form. These region's boundaries can be detected by edge detection methods (Gonzalez and Woods, 2002), but these typically work on pixel level and do not produce smooth boundaries. A more natural approach is to keep the image-based representation, and to form implicit functions with a level set representing region's boundaries.

- *The development of a region-aware modeling method based on implicit functions that maintains information of all region interfaces and generates accurate three-dimensional computer models in terms of geometric quality and low volumetric error.*

Natural heterogeneous objects might be composed of spatially different regions. A modeling method that represents all regions as well as their incident surfaces is very important. These models must represent not only the outer object geometry accurately, but also contain information of the inner region interfaces.

1.3 Document organization

The structure of this document is as follows:

Chapter 2 describes in detail the representation schemes using in geometric modeling of solid objects that allow, due to their structure, reflecting the lack of homogeneity within the set of points, which is the key element in the modeling of natural heterogeneous objects. The advantages and disadvantages of each scheme are discussed at the end of the chapter. This will help to understand the subsequent chapters of this dissertation.

Chapter 3 focuses on modeling specific regions composing natural heterogeneous objects. This process is addressed by mean of a computer-aided reverse-engineering based modeling *(CAREM)* method based on medical images used in biomedical applications. Since medical images are jaggy, not properly ordered and contains so much noise, traditional voxel models are not adequate for simulations and CAD analysis. To tackle this issue the *CAREM* method builds a compact analytic representation via an implicit surface reconstruction. As part of this method, an effective Voxel-To-Point conversion process is proposed. Some examples are given to illustrate the use of the *CAREM* method.

Chapter 4 focuses on the problem of modeling natural heterogeneous objects composed of multiple regions. Here, a region-aware model *(RAM*) for natural object modeling is presented. The *RAM* method is based on a mathematical model in which the geometry of the natural object is represented by a set of feature-regions implicitly represented. These feature-regions are organized in a vector implicit function VIF structure. To demonstrate the ability of the *RAM* method to handle complex geometries some examples are shown.

Chapter 5 presents the results from applying both the *CAREM* method and the *RAM* method described in Part II. In the chapter is given a description of the criteria that can be used to evaluate the model representation. Then capabilities and limitations of these methods are discussed to make clear what kind of object's configurations the methods can and cannot model.

Chapter 6 concludes with a summary of the dissertation, presents the principal contributions, and outlines several avenues of future research based on the present work.

1.4 List of publications

The main elements of this dissertation have been published in different journals and proceedings of conference.

Part of the work on three-dimensional computer modeling, presented mainly in Chapter 2 and partly in Chapter 3, were accepted for publication under the title:

- Gómez-Mora, M. and Flórez-Valencia, L. (2013). *Three-dimensional object reconstruction*. Revista Vínculos, 9(1): 75-86.
- Gómez-Mora, M. *Aplicación de realidad virtual en la rehabilitación cognitiva* (2013). Revista Vínculos, 10(1): 130-135.
- Gómez Mora, M. and Flórez-Valencia, L (2015). *Modeling of heterogeneous objects: an approach based on implicit functions*. Revista Tecnura, 20(47): 15-27

Based on the material of Chapters 4 and 5, the following preprint has been prepared to submit :

- Gómez Mora, M. and Flórez-Valencia, L. (2016). *A region-aware method for three-dimensional bioCAD modeling of natural heterogeneous objects.* Computer-Aided Design, 2016(X), XX–XX.

Finally, three conference proceedings, based on this work, can be found on the web:

- Gómez Mora, M. and Flórez-Valencia, L. (2012). *Implicit surface reconstruction from three-dimensional image data with radial basis functions*. In SIGGRAPH BOGOTA 2012 posters.
- Gómez Mora, M. and Flórez-Valencia, L. (2015). *Surface reconstruction from three-dimensional segmented images using implicit functions*. In Computing Colombian Conference (10CCC), pages 317-323. IEEE.
- Bustacara, C., Gómez Mora, M., and Flórez-Valencia, L. (2016). *Anisotropic diffusion for smoothing: A comparative study*. In International Conference on Computer Vision and Graphics (ICCVG 2016), pages XX-XX. Springer

Chapter 2

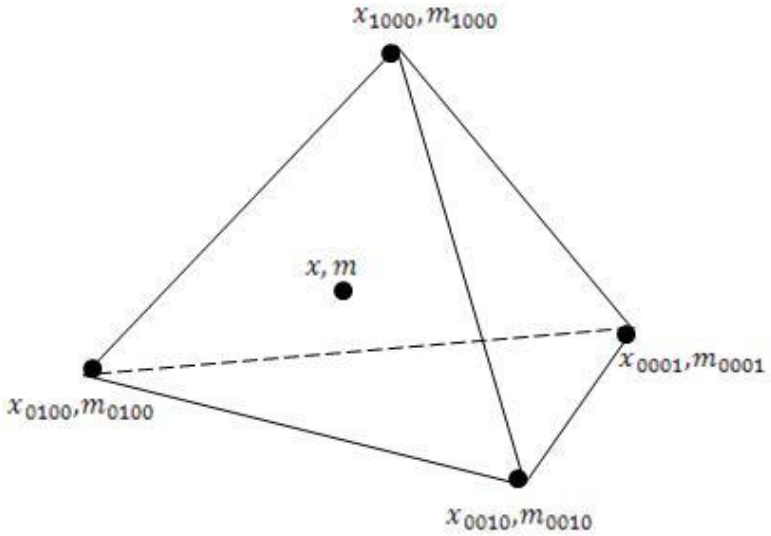
Three-dimensional object modeling

2.1 Introduction

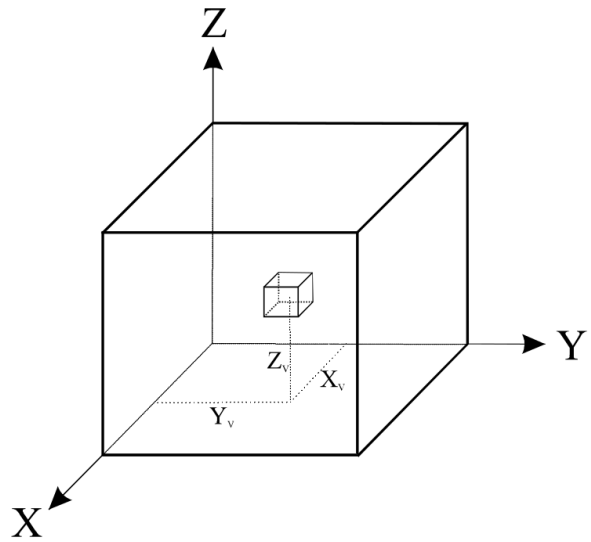
In this chapter a variety of modeling approaches that are considered to be relevant to the general problem of three-dimensional modeling of natural heterogeneous objects are reviewed. This survey will by no means be exhaustive, but it does form the basis for understanding the issues related to the modeling of natural heterogeneous objects, and it will give the reader a sufficient background to follow the rest of this dissertation.

Several mathematical models have been developed to model natural objects; all of them have their own strengths and weaknesses and depend on the field of application. Three fundamental approaches can be taken at the moment of modeling natural objects: express the object including its interior and boundary that is a solid modeling approach (see Section 2.2), or use only an expression of its boundary that is a surface modeling approach (see Section 2.3). In both cases an explicit representation or an implicit representation of the natural object can be used.

Additionally, based on the idea of partitioning the natural heterogeneous object into homogeneous regions an approach called composite models is reviewed in Section 2.4. Finally, a discussion on the advantages and disadvantages of the examined representations is presented in Section 2.5.



(a) A tetrahedron node with material information.



(b) A n×n×n volume and a voxel with coordinates (Xv, Yv, Zv).

Figure 2.1: Examples of Cells for solid and volume modeling.

2.2 Solid modeling

Due to the fact that it is impossible to store in the computer the whole set of points that can constitute a continuous natural solid object, normally, the space

occupied by it is discretized. Thus, the natural object is represented by decomposing the occupied space using a set of cells and maintaining information concerning to each one of them (Mantyla, 1988). The kind of cells used and the way in which these are combined determines the different strategies carried out on this basis idea (see Figure 2.1).

However, the different strategies have a common requirement: cells must be disjoint or have only faces and / or edges and / or vertices in common. That is, they must form a solid covering without spatial overlap between cells, or what is the same, a space partition. Below both concepts are formally defined.

Definition 1. (Covering): Let *"I"* be an index set of the form $I = \{1, \ldots, n\}$. It is said that a family of subsets $A_i \subseteq A \mid i \in I$ of a set A is a covering of A if the following conditions are met:

$$\begin{gathered} A_i \neq \emptyset, \quad \forall i \in I \\ \bigcup_{i \in I} A_i = A \end{gathered} \tag{2.1}$$

Definition 2. (Partition): Let *"I"* be an index set of the form $I = \{1, \ldots, n\}$. It is said that a family of subsets $A_i \subseteq A \mid i \in I$ is a partition of A if the family is a covering and it is true that:

$$A_i \cap A_j = \emptyset, \quad \forall i, j \in I, \quad i \neq j \tag{2.2}$$

In other words, the different strategies have in common that space, representing the natural object, is divided into cells that do not overlap and, therefore, any point belonging to it can be identified as being within a particular cell (or cells if fall on the border between neighboring cells). Exhaustive enumeration (see Subsection 2.2.1) and hierarchical spatial partitioning schemes (see Subsection 2.2.2) are two explicit strategies that allow, due to its structure, reflecting the lack of homogeneity within the set of points, which is the key element for modeling natural solid objects. Additionally, an object can be defined by a scalar function defined on d-dimensional Euclidean spaces through an implicit solid modeling (see Subsection 2.2.3).

2.2.1 Exhaustive enumeration scheme

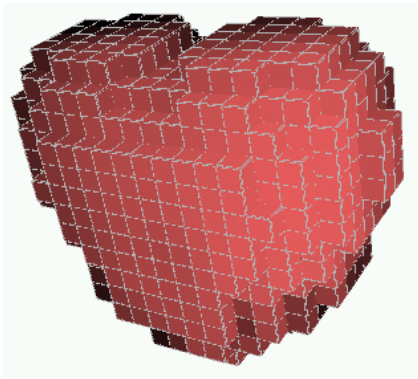
In this scheme the interior of a solid object is discretized into smaller elements, such as voxels models, and tetrahedrons (Kou and Tan, 2007).

Voxel Models

One of the ways of representing a natural object is to store its occupied volume by a set of data samples called voxels (volume elements) (Kaufman et al., 1993) (see Figure 2.1b). The voxel-based representation can be generalized as (Kou and Tan, 2007):

$$O = \{V_i\} = \{(x_i, y_i, z_i, m_i), \quad 1 \leq i \leq n\} \tag{2.3}$$

where V_i is a voxel element in the collection of n voxels that constitute the natural object O, (x_i, y_i, z_i) represents the geometric location of the voxel V_i and m_i represents the composition of the material inside the volume.



(a) A voxel-based solid model of a heart image.

Coordinates (x,y,z)	position: 0-outside, 1-inside
(0,0,0)	0
(0,0,1)	0
...	...
(50,25,8)	1
(50,25,9)	0
(50,25,10)	1
...	...
(512,512,129)	0

(b) Binary data table indicating the position of each voxel in the image.

Figure 2.2: A voxel-based solid model and its data table of a heart image.

Typically, each discrete sample can store a specific object property. In the simplest case, each sample stores a binary value of "0", indicating that the sample is outside the object, or "1",indicating that the sample is inside the it. For example,

Figure 2.2a illustrates a voxel-based model and its corresponding data table of a heart image depicted in Figure 2.2b.

In a general case, samples can store more general information. For example, three-dimensional scenes depicted in such a way can be captured from natural objects through different voxel techniques, such as computed tomography, magnetic resonance imaging, ultrasound, geophysical measurements, etc. (Nielson, 2000).

Material property at a point can be evaluated only by identifying the voxel in which is located. The material properties can be estimated in several ways: by scanning techniques, such as MRI and CT, or by using various functions and distances (Angenent et al., 2006).

Finite element-based representation

In this representation, the interior of solid objects are discretized using volume meshes (see Figure 2.3).

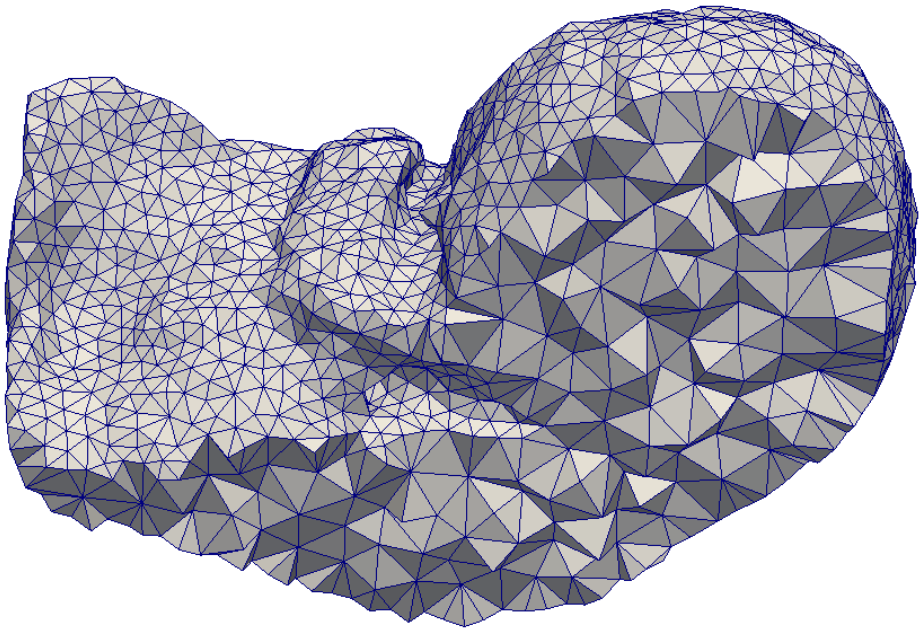


Figure 2.3: A finite element-based representation of a human liver.

A collection of subvolumes or polyhedra is used to represent both the geometry

and the material properties with parameters specified over the nodes (tetrahedrons) of the mesh $c_k \in C$ (see Figure 2.4). The vertices store their geometric position as well as the material compositions, which can be symbolically described as:

$$\begin{aligned} C &= \{c_1, c_2, ..., c_n\} \\ c_k &= \{G(c_k), M(c_k), 1 \leq k \leq n\} \\ M(c_k) &= F(\mathbf{x}_{k1}, \mathbf{x}_{k2}, ..., \mathbf{x}_{kn_g}) \end{aligned} \tag{2.4}$$

where c_k denotes a representative polyhedron, $G(c_k)$ and $M(c_k)$ denotes the geometries and material distributions of c_k, where $\mathbf{x}_{k1}(x_1, y_1, z_1, m_1)$ is a representative vertex of c_k, n is the number of polyhedrons that compose the whole natural object, and the function F indicates that the material composition of the points in each sub-volume. The material composition of the model is represented by a vector-valued function of $\mathbf{m}(\mathbf{x})$, where each component m_i represents the relevant material volume fraction at a point x within the model.

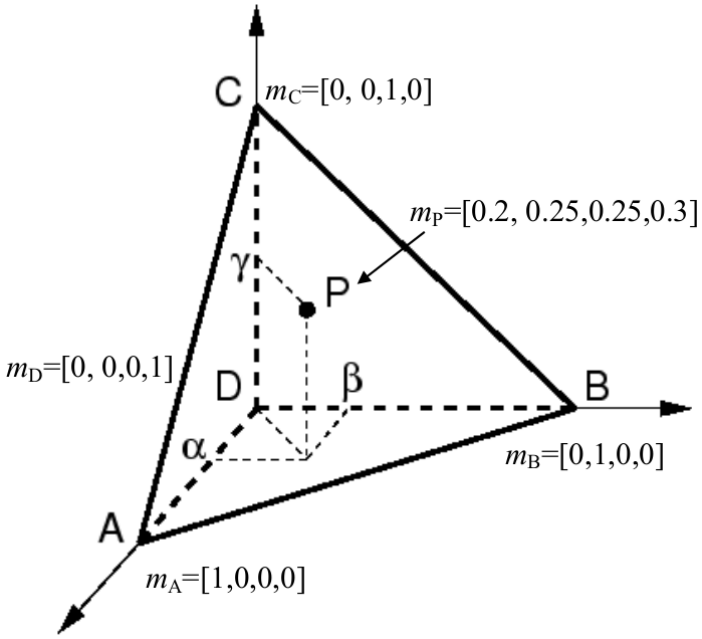


Figure 2.4: Material blending with the barycentric Bernstein polynomial.

The modeling space is divided into subdomains that are topologically simple enough to represent analytically the shape of the object and the material function. For each tetrahedron domain c_k (see Figure 2.3), the shape and composition

are formulated in terms of control points $\{\mathbf{x}_{k,i}\ ,|\ \mathbf{i}\ |= n_g\}$ and control compositions $\{\mathbf{m}_{k,i}\ ,|\ \mathbf{i}\ |= n_m\}$ which are blended by barycentric Bernstein polynomials (Jackson et al., 1999):

$$[\mathbf{x}_k(\mathbf{u}), \mathbf{m}_k(\mathbf{u})] = \left[\sum_{|\mathbf{i}|=n_g} B_{\mathbf{i}}^{n_g}(\mathbf{u})\mathbf{x}_{k,\mathbf{i}}, \sum_{|\mathbf{i}|=n_m} B_{\mathbf{i}}^{n_m}(\mathbf{u})\mathbf{m}_{k.\mathbf{i}}\right] \tag{2.5}$$

where n_g and n_m are the degrees of variation in shape and composition. $|\ \mathbf{i}\ |= i_0 + i_1 + ... + i_k$ and k is the dimension of cell k in this case 4 (see Figure 2.4).

The proposed representation is able to represent objects with a complex composition distribution.

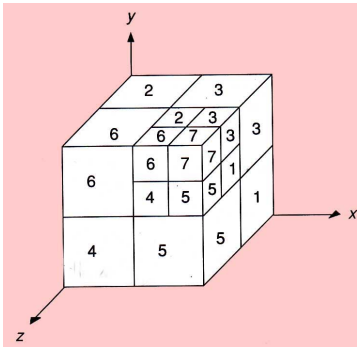
2.2.2 Hierarchical spatial partitioning scheme

The idea of hierarchical spatial partitioning schemes is based on subdividing the space's region into several subregions, and this same subdivision is recursively applied to each of the newly oriented regions, until a stopping criterion is met. Whereas the initial region, usually a cubic shape, includes all the object, the stopping criterion is used to classify the region whether it contains only a portion of the natural object or only space outside the it. Generated regions using the recursive procedure are kept in a tree called spatial partitioning tree. Most of these schemes use planes for the subdivision of the regions. The choice of these planes determines the different variants in such schemes. The best known variants are: Bintrees (Samet and Tamminen, 1985), (Diehl, 1988), BSP-trees (Thibault and Naylor, 1987), (Naylor, 1990), (Paterson and Frances Yao, 1992), Kd-trees (Bentley, 1975), (Greß and Klein, 2004) and Octress (Meagher, 1982), (Yamaguchi et al., 1984), (Samet and Webber, 1988). Next, the Octree and Kd-tree representations are briefly described.

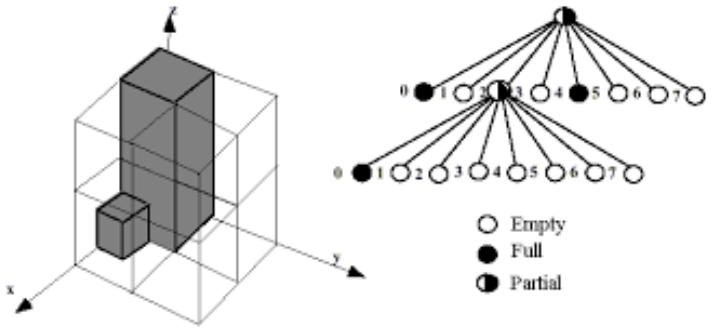
Octree representation

Octree representation was independently developed by Jackings and Tanimoto (Jackins and Tanimoto, 1980), and Reddy and Rubin (Reddy and Rubin, 1978). In octree representation, the tree-dimensional space surrounding the object is

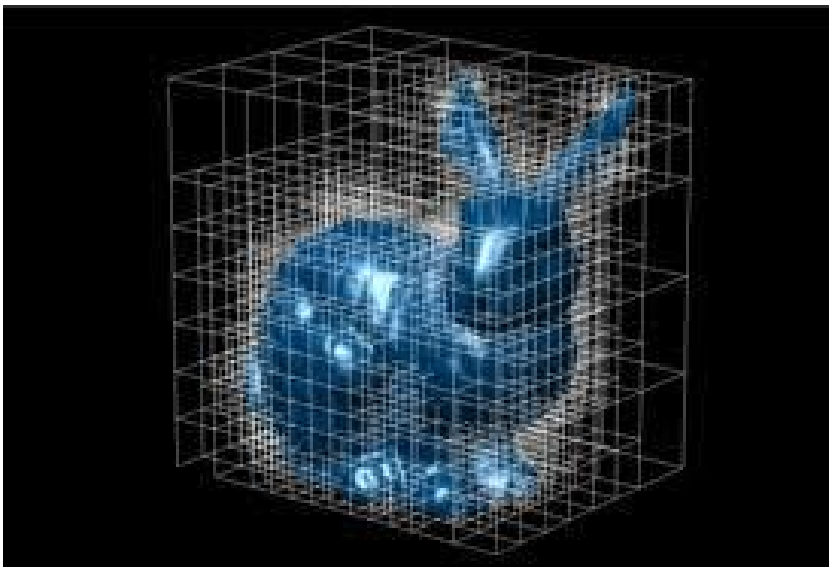
called a universe. The universe is divided into cubes (octants). Each cube may be labeled as in Figure 2.5a. Cubes partially occupied by the object are subdivided and labeled in the same way recursively until they are of the minimum allowed size.



(a) The octree enumeration Jackins and Tanimoto (1980).



(b) An octree representation and its tree structure defining subdivision.



(c) The octree representation of the Stanford bunny model.

Figure 2.5: Octree representation.

In this case, an octree is a tree that hierarchically defines the object's shape starting with a root node that covers the entire universe. Each cube represented by a node in the tree is then successively subdivided into 8 smaller cubes (dependent on the contour information) with the cube's edge lengths halved. Further subdivision of a cube is stopped if it either lies completely inside or outside the

object.

Figure 2.5b shows an example for such an octree subdivision. The volumetric object on the left is defined by the tree on the right. Each node in this tree specifies if one of its subcubes lies completely inside (black), completely outside (white), or on the object's surface (gray). In the latter case, the corresponding voxel is further subdivided until a maximum layer defining the highest resolution is reached. This way, it is assured that the object's surface is accurately modeled while regions outside or inside the object can be efficiently described by large cubes.

Thus, the result from the recursive subdivision process is represented by a 8-degree tree (see Figure 2.5b). Each node of the octree is assigned the same label as that of the corresponding octant. Essentially, the octree representation is an approximate representation of the three-dimensional object. As an example, the octree representation of the Stanford bunny is depicted in Figure 2.5c.

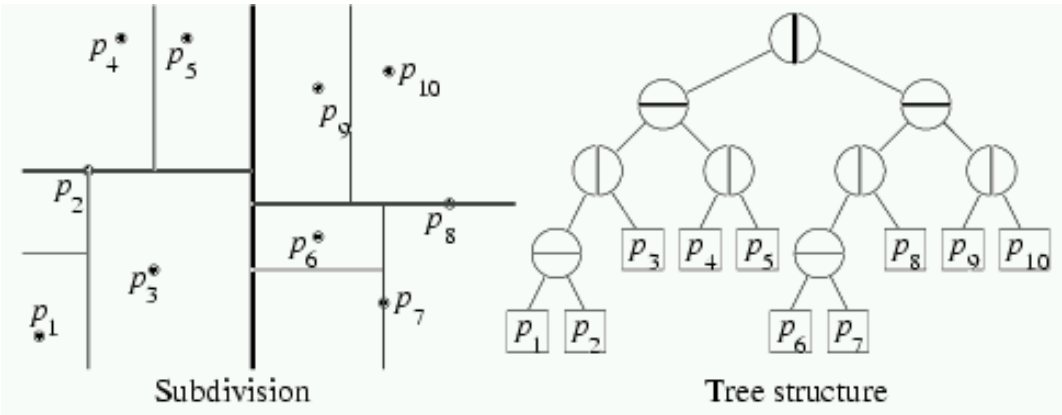
Kd-tree representation

A kd-tree is a generalization of a binary search tree that stores a collection of points in k-dimensional space (Bentley, 1975). The root node of the kd-tree is a box which contains all data points P and the whole domain Ω.

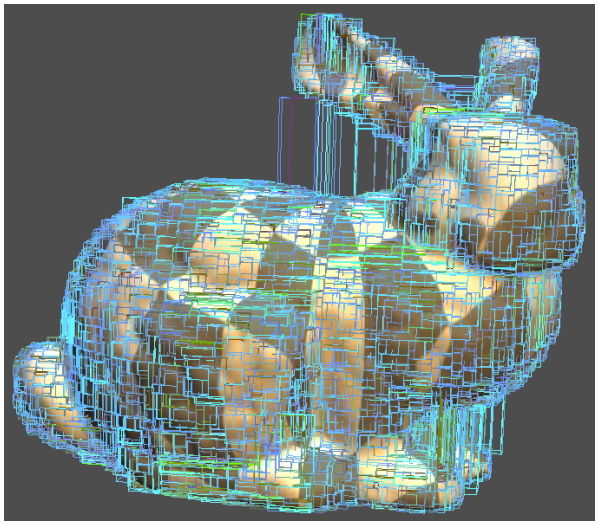
Building a kd-tree starts from the root-node, defined as discriminator 0, and bisect recursively the cells through their longest axis, so that an equal number of particles lie in each subvolume (see Figure 2.6a). This bisection is accomplished using Hoare's median finding algorithm, which is an average time operation per level of the tree, making the tree building process an operation. The depth of the tree can be chosen so that it ends up with at most one particle in the leaf cells (buckets). A kd-tree representation of the Stanford bunny is depicted in Figure 2.6b.

Several factors motivated the use of kd-tree structure over the classical octree. The simplicity of the structure and the availability of fast median finding algorithms allow for a very efficient tree-construction. Pointers are unnecessary since each node in the tree can be indexed so that the finding of children, parent and sibling nodes are simple bit-shift operations. The use of buckets, by which

only nodes are required, makes the tree structure memory-efficient. Most significantly, it can be extended to a parallel, distributed tree structure in a very natural way.



(a) The kd-tree structure (Bentley, 1975).



(b) A kd-tree representation of the Stanford bunny model.

Figure 2.6: kd-tree representation.

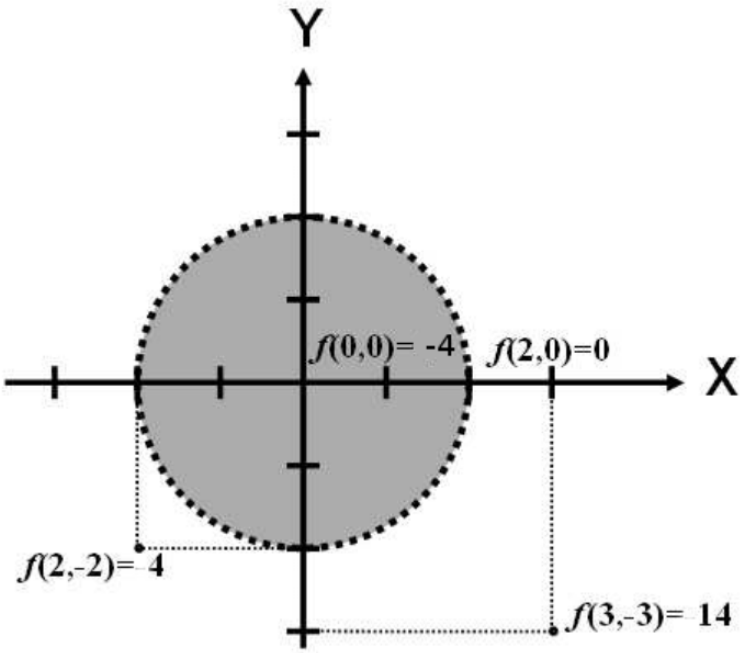
2.2.3 Implicit solid modeling

As seen in the previous Section, spatial decomposition schemes have the common feature that the cells form an abstract solid partition defined as sets of continuous points. Another way of representing natural objects in $\mathbb{R}^d$ is through

a scalar function defined on d-dimensional Euclidean space with the inequality(Bajaj, 1997):

$$F(\mathbf{p}) = f(\mathbf{p}) \leq T;\ T \in \mathbb{R},\ \mathbf{p} \in \mathbb{R}^d,\ f(\mathbf{p}) : \mathbb{R}^d \rightarrow \mathbb{R} \tag{2.6}$$

where T is a threshold value. The subset $\mathbf{p} \in \mathbb{R}^d \ : \ f(\mathbf{p}) \leq T$ is called *natural solid object,* and the subset $\mathbf{p} \in \mathbb{R}^d \ : \ f(\mathbf{p}) = T$ is called *iso-surface* (see Subsection 2.3.2).



(a) A 2D disk of radius 2 defined by the function $f(x, y) = x^2 + y^2 - 2^2$.

$$\begin{cases} f(\mathbf{p}) < T, & \mathbf{p}\ is\ inside\ the\ object \\ f(\mathbf{p}) = T, & \mathbf{p}\ is\ on\ the\ object's\ boundary \\ f(\mathbf{p}) > T, & \mathbf{p}\ is\ outside\ the\ object \end{cases}$$

(b) Conditions to classify any point $\mathbf{p}$ in the space $\mathbb{R}^d$.

Figure 2.7: Implicit solid modeling.

It might be easier to understand how an implicit function is defined making reference to Section 2.2.1. Since the voxel representation allows storing discrete samples of the object's volume, then, it could be known that a sample is within

the object if the representation stores a positive nonzero value. The same approach can be extended to any point in space rather than just discrete voxels (Menon et al., 1996).

An implicit function can be defined using Euclidean coordinates as input parameters and returning the scalar value associated to the function (see Figure 2.7b). Such scalar value can be interpreted in a similar way as if it was stored in a voxel, just inverting the sign. If the value returned by the implicit scalar function for the given point is less than a threshold value, this point is inside the object (see Figure 2.7a). If the scalar value returned is equal to the threshold value, the point is on object's surface. If the scalar value is greater than the threshold value, the point is outside the object.

The scalar value returned by the function can be interpreted as a distance to the object's surface described by the function. For example, the implicit function $f(x,y) = x^2+y^2-2^2$ defines a disk of radius 2 centered at the origin of coordinates (see Figure 2.7a). In this particular case, the interest is placed in a disc with a threshold value equal to zero, i.e., values less or equal to $T = 0$ are sought. Figure 2.7a shows a series of scalar values returned by this function for different points in a 2D plane.

The gray area is a subset of 2D plane where values returned by the function $f(x,y)$ are negative ($f(0,0) = -4$). The dashed contour is a set of points where the function$f(x,y)$ returns 0 (point (2,0) belongs to the periphery of a circle as $f(2,0) = 0$). For any other point in a 2D plane $f(x,y)$ returns positive values ($f(3,-3) = 14$). Hence any point on a 2D plane can be classified unambiguously. A set of points for which the function$f(x,y)$ returns any less than or equal to 0 describes a set of points that belong to the volumetric object (see Figure 2.7a). Complex functions describing arbitrary objects in d-dimensional space can define in the same way. For instance, In the constructive representations, primitives are defined by implicit surfaces, $F(x,y,z) = 0$ (see Subsection 2.3.2). Thus, the surface divides the space into two parts: $F(x,y,z) \geq 0$ and $F(x,y,z) < 0$.

Ricci (Ricci, 1973) organized multiple implicit surfaces into binary constructive solid geometry trees to construct complex solid geometries from simpler primitives (see Figure 2.8). In particular, constructive solid geometry is supported by boolean algebra and a set of well-understood regularized set operators, allowing complicated objects to be represented as combinations of various ordered

union, intersection and difference operations on simpler solid objects, which may be bounded primitives (Requicha, 1977).

Requicha (Requicha, 1977) proposed r-sets (regular sets) as appropriate models for representing abstract solids.

Definition 3. (R-set): An r-set is any bounded, closed, regular and semi-analytic subset of E^3 (Kumar et al., 1999), (Requicha, 1980).

An r-set model may be viewed as the limit of a certain sequence of manifold solids (see Definition 3)), or more intuitively, as a generalized manifold solid (Desaulniers and Stewart, 1992). However, r-sets are not algebraically closed under the traditional set of boolean operations. To fulfill the property of *closing against rigid transformations and certain boolean operations*, the set of regularized boolean operations was established (Requicha and Tilove, 1978), (Voelcker and Requicha, 1977). This provides a correct result by performing a closing of the interior of the set resulting from the classic boolean operation. For instance, the 2D object depicted in the left part of Figure 2.8 is built from three 2D simple primitives: two rectangles (A and C), and one circle (B). The constructive solid geometry model is $C+(A-B)$, where $C(x,y) = \max(|\ x-(1.5)*2\ |\ -3, |\ y-(2.5)*2\ |\ -2)$, $A(x,y) = \max(|\ x\ |\ -6, |\ y\ |\ -3)$, and $B(x,y) = (x-1)^2 + y^2 - 1)$.

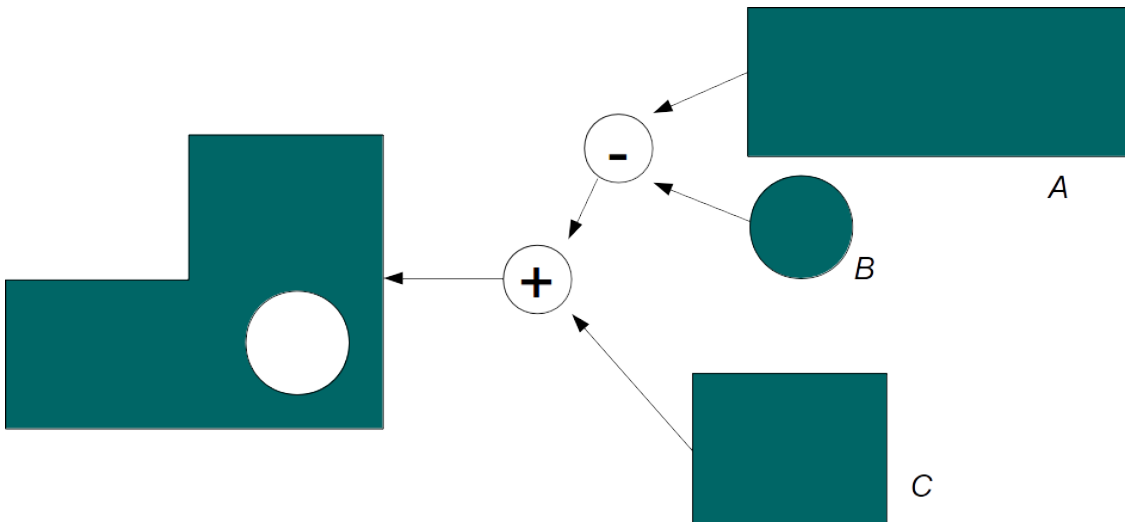


Figure 2.8: Constructive solid geometry based model (Ricci, 1973)

Similarly, Pasko (Pasko et al., 1995) generalized the representation of implicit surfaces in a system which uses blending, warping and boolean operations to

combine different forms of implicit models, and denoted it as function representation (Frep).

However, traditional boolean operations assume objects of interest are homogeneous in material compositions and the major concern is focused on boundary evaluation algorithms (Mortenson, 1997), (Qian and Dutta, 2003). More specifically, the task of boolean operation is to construct the new boundary representation for the output object, from input objects A, B and C (operands) and boolean operators (difference (-) and union (+)) (see Figure 2.8).

2.3 Surface modeling

Most natural objects can be modeled as surfaces. In order to be able to build three-dimensional surface models at least two-manifold surfaces have to be obtained. A two-manifold can be interpreted intuitively as a surface that does not intersect itself (Massey, 1991).

Definition 4. (Manifold): A d-dimensional manifold with boundary is a Hausdorff space such that each point has an open neighborhood homeomorphic to $\mathbb{R}^d$ or to $\mathbb{R}^d_+ = \{(x_1, ..., x_n) \in \mathbb{R}^d \,|\, x_n \geq 0\}$. A two-manifold with boundary is an d-dimensional manifold with d=2.

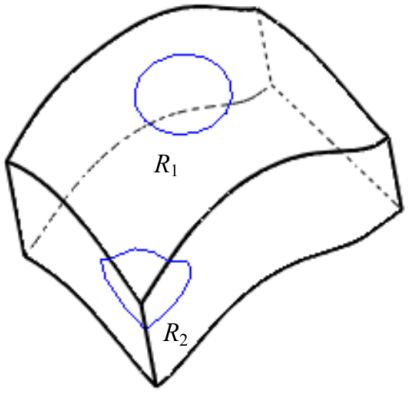
In a manifold representation (see Figure 2.9), every point on a surface is two-dimensional, i.e., every point has a neighborhood which is homeomorphic to a two-dimensional disk (Weiler, 1988). To better understand such manifold conditions, an example of a two-manifold solid object is shown in Figure 2.9a. Intuitively, for any point on the boundary surface of a manifold object, its neighboring region is homeomorphic to a 2D disk.

Definition 5. (Homeomorphism): A homeomorphism is defined as a continuous invertible map $f : K \to H$ whose inverse $f^{-1} : K \to H$ is also continuous.

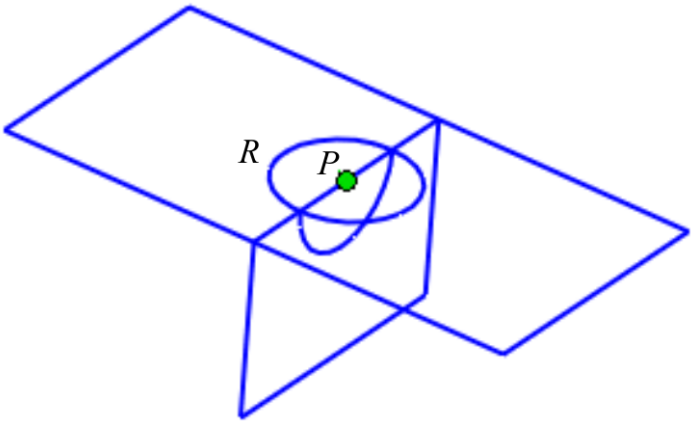
As seen from Figure 2.9b, even though the region R1 exists in three dimensional space, it is still topologically flat when the surface is examined closely enough in a small area around any given point (Weiler, 1988). The same observation can also be derived for the region R2 in Figure 2.9b and such objects are said

to be two manifold objects. However, in Figure 2.9c, the point of interest P is at the intersection of more than two topologically two-dimensional surfaces and the region R around P is not topologically flat, no matter how close the region is examined around the point.

(a) A two-manifold solid object.



(b) Homeomorphism with a 2D disk.



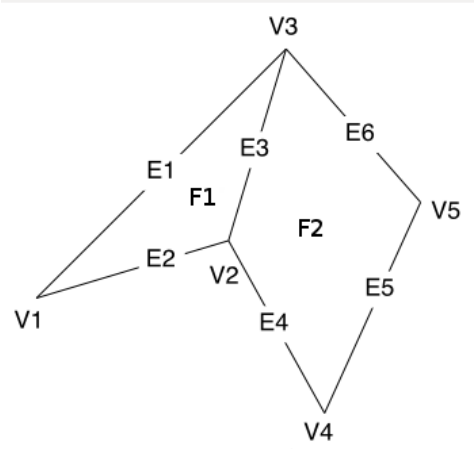
(c) Non-manifold faces.

Figure 2.9: Manifold surface representation.

Although a wide range of surface representations have been developed, only two of them will be presented: boundary representation (see Subsection 2.3.1), and implicit surface representation (see Subsection 2.3.2).

2.3.1 The boundary representation

The boundary representation model (BRep) is used to define object's shapes as sets of connected surface elements (Braid, 1974). BRep models consist of geometric and topological information. While geometric information includes a description of points belonging to the surface of an object, topological information specifies the connections between points on the object's surface (Braid, 1974).



(a) A BRep model.

Vertex	Coordinates
V_1	x_1, y_1, z_1
V_2	x_2, y_2, z_2
V_3	x_3, y_3, z_3
V_4	x_4, y_4, z_4

Edge	Vertices
E_1	V_1, V_2
E_2	V_2, V_4
E_3	V_1, V_4
E_4	V_2, V_3
E_5	V_3, V_4
E_6	V_1, V_3

Face	Edges
F_1	E_1, E_2, E_3
F_2	E_3, E_4, E_5, E_6

(b) Data table for vertices, edges, and faces.

Figure 2.10: A BRep model and its data tables

The most common type of BRep is polygonal representation (see Figure 2.10). Thus, the model of a surface (S) consists of a set of vertices (V), edges (E) and faces (F) (Van Dam et al., 2013).

$$
\begin{gathered}
S = \{V, E, F\} \\
V = \{v_1, \ldots, v_n\} \\
E = \{e_1, \ldots, e_l\};\ e_i = \{j, k\};\ 0 < j, k \leq n \\
F = \{f_1, \ldots, f_m\};\ f_i = \{e_j, e_k, \ldots, e_p\};\ 0 < j, k, p \leq l
\end{gathered}
\tag{2.7}
$$

For example, Figure 2.10 illustrates a Brep model and its corresponding data tables for vertices, edges, and faces.

2.3.2 Implicit surface modeling

In Subsection 2.2.3 a natural object was expressed implicitly including its interior and boundary. However, such a natural object can also be modeled using only an expression of its surface through a function that satisfies a specified condition, such as:

$$
S = \{(x_1, \ldots, x_n)\ :\ F(x_1, \ldots, x_n) == true\} \tag{2.8}
$$

Thus, an implicit surface S in $\mathbb{R}^d$ can be defined as the zero-level set of some function $f : \mathbb{R}^d \to \mathbb{R}$:

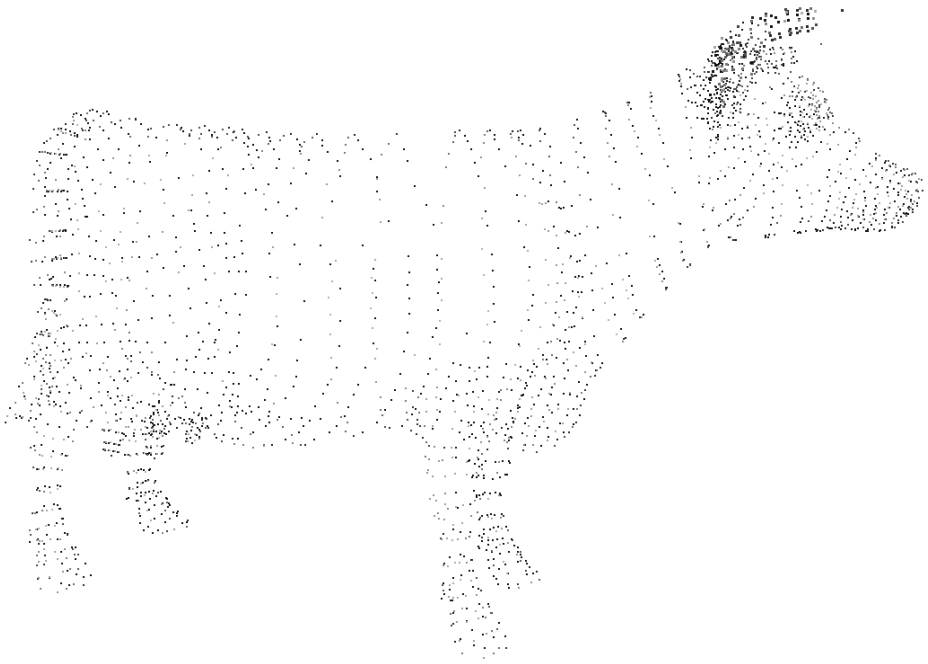
$$
S = \{\mathbf{x} \in \mathbb{R}^d :\quad f(\mathbf{x}) = 0\} \tag{2.9}
$$

There are some ways to define f. On the one hand, it can have a closed analytical form, for example, the zero level-set of the function $f(x, y, z) = x^2 + y^2 + z^2 - 1^2 = 0$ defines a unit sphere, classifying the space into two half-spaces $f > 0$ and f<0. On the other hand, an implicit function f can also be defined through data-driven methods taking the form of a signed distance field (Hoppe et al., 1992), or an indicator function (Kazhdan et al., 2006).

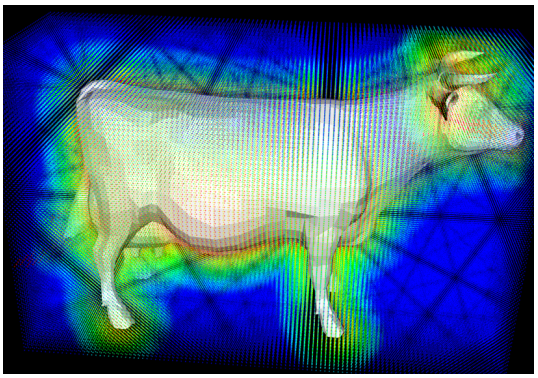
Signed distance fields

Hoppe (Hoppe et al., 1992) proposed an innovative method that approximates a signed distance field $f : \mathbb{R}^3 \to \mathbb{R}$ by assigning each point $\mathbf{p} \in \mathbb{R}^3$, in the point cloud P, a signed projection onto the tangent plane of its closest point to S, denoted $\mathbf{p}_i$:

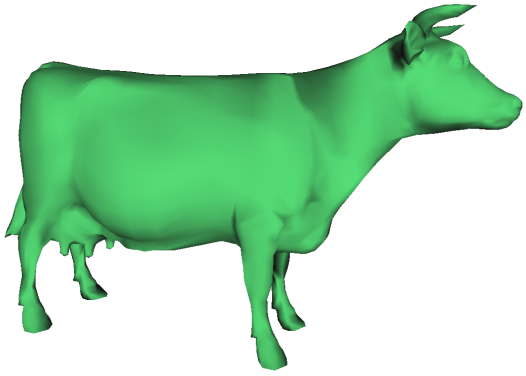
$$
f(\mathbf{q}) = (\mathbf{q} - \mathbf{p}_i) \cdot \mathbf{n}_i \tag{2.10}
$$



(a) Point cloud with 2900 points generated from a cow model.



(b) Distance map of Figure 2.11a over a grid x={0,...,12}, y={0,...8}, z={0,...,5} with a step of 0.1.



(c) Surface extraction from Figure 2.11b using the Marching Cubes algorithm.

Figure 2.11: Surface reconstruction using signed distance fields

In Figure 2.11c a point cloud $\mathbf{P}$ containing 2900 points scanned from the surface of a cow. The normal field $\mathbf{N}$ can be estimated from the point cloud $\mathbf{P}$ by a local principal component analysis of all points $\mathbf{p}_i \in \mathbf{P}$, in a k–neighborhood of $\mathbf{p}_i$. More precisely, for every point $\mathbf{p}_i$, the covariance matrix C_i is computed on the k–neighborhood and the eigenvector corresponding to the smallest eigenvalue is considered as the normal to point $\mathbf{p}_i$.

$$C_i = \sum_{j=0}^{k} (\mathbf{p_j} - \mathbf{o}_i)(\mathbf{p_j} - \mathbf{o}_i)^T \tag{2.11}$$

Note that normal field $\mathbf{N}$ must be oriented in order to obtain a good estimation of the signed distance field (see Figure 2.11b).

The orientation of the normal field $\mathbf{N}$ can be modified in order to ensure consistency of the surface orientation, i.e. two close points $\{\mathbf{p_i}, \mathbf{p}_j\} \in \mathbf{P}$ have a similar orientation. Ideally, when the point set $\mathbf{P}$ is dense, the sampled surface is smooth. The surface is then defined by the zero level set of the signed distance function f (see Figure 2.11c).

Radial basis functions

Another option to define f is to interpolate a set of irregularly distributed points using various forms of radial basis functions (RBF), such as thin-plate splines or Gaussian (Carr et al., 2001), (Turk and O'brien, 2002).

The method presented in (Carr et al., 2001) constructs the surface by finding an implicit function defined via RBFs whose zero level-set represents the surface (see Figure 2.12a). More specifically, the authors use globally-supported basis functions of the form $\phi(r) = \| r \|_2$. The implicit function f may be expressed as:

$$f(\mathbf{p}) = P(\mathbf{p}) + \sum_{i=1}^{n} \lambda_i \phi(r_i), \quad r_i = \| \mathbf{p} - \mathbf{p}_i \| \tag{2.12}$$

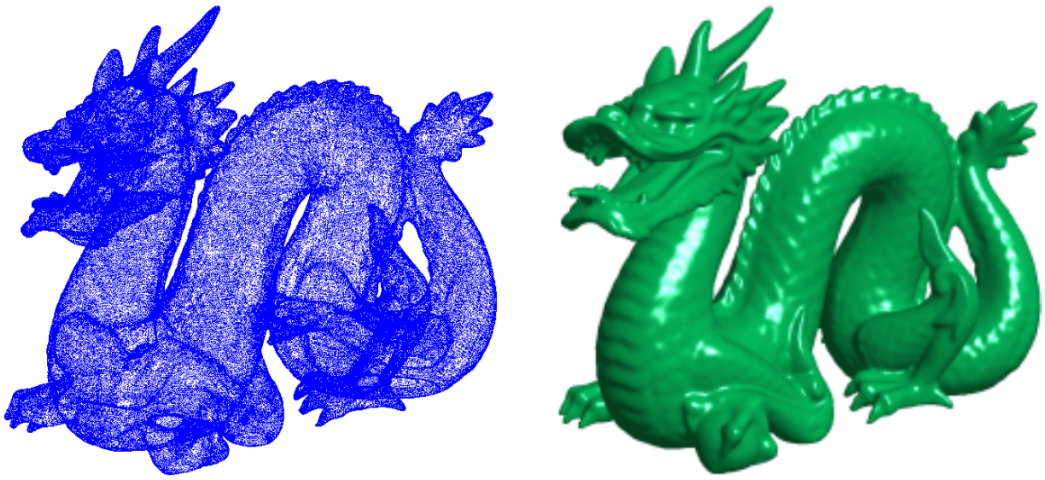
where $P(\mathbf{p})$ denotes a low-degree polynomial and the basis functions are shifted by the evaluation point $\mathbf{p}$. Coefficients λ_i are found by prescribing, as interpolation constraints, a function value of 0 for $\mathbf{p}_i \in \mathbf{P}$.

Coefficients λ_i are calculated so that a set of interpolation conditions of the form $f(\mathbf{p}_j) = d_j, \quad j = 1, \ldots, n$ are satisfied. This leads to a linear system of equations of the form

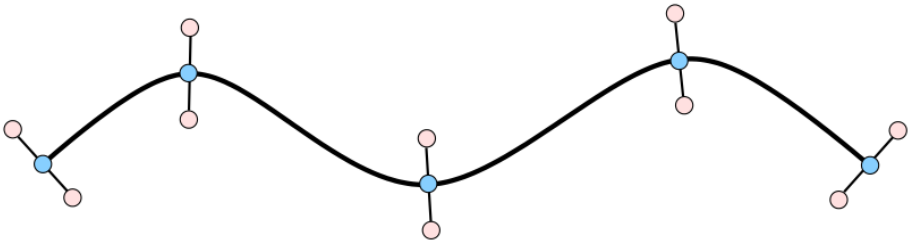
$$A\mathbf{c} = \mathbf{d} \tag{2.13}$$

where A is a symmetric N × N matrix with components $A_{ij} = \phi(\| \mathbf{p}_i - \mathbf{p}_j \|)$, $\mathbf{c} = [c_1, \ldots, c_n]^{\mathrm{T}}$is the unknown vector of coefficients and $\mathbf{d} = [d_1, \ldots, d_n]^{\mathrm{T}}$. The interpolated surface is arbitrarily taken to be the zero iso-contour of the implicit function, i.e. we set $f(\mathbf{p}) = 0$ if $\mathbf{p}_i$ is on the surface.

This system has a solution if A is non-singular and a non-trivial solution may be obtained only if $\mathbf{d} = 0$. This requires that, at least, one interpolation point must lay away from the iso-surface. These points are often referred to as off-surface points. Off-surface constraints are necessary to avoid the trivial solution of $f(\mathbf{p}) = 0$ for $\mathbf{p} \in \mathbb{R}^3$. Positively and negatively valued constraints are set for points displaced at $\mathbf{p}_i$ along $\mathbf{n}_i$ in the positive (negative) direction (see Figure 2.12b).



(a) Point cloud and surface reconstruction of a dragon model.



(b) Off-surface constraints points.

Figure 2.12: Radial basis functions (Carr et al., 2001).

Displaced points are selected such that each one's closest point in P is $\mathbf{p}_i$. However, when the input contains some noise, determining the proper placement

of off-surface points can become challenging. Coefficients λ_i are found via a dense linear system in n, however, by exploiting the structure of the radial basis function ϕ, fast multipole methods are used to reduce complexity from $O(n^3)$ to $O(n \log n)$ (Carr et al., 2001).

Alternatively, f can be defined in a piecewise manner using an adaptive space subdivision scheme (Frisken et al., 2000), (Ohtake et al., 2005). Given a set of points $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$ sampled from a surface equipped with unit normals $\mathbf{N} = \{\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_n\}$, the multi-level partition of unity (MPU) approach defines an implicit function $f(\mathbf{x})$ that is an approximation of the signed distance from $\mathbf{P}$. This function divides the space into the interior $f(\mathbf{x}) < 0$ and the exterior $f(\mathbf{x}) > 0$ of the object. The boundary surface corresponds to the zero-level of the distance function $f(\mathbf{x})$.

Globally, a MPU function is composed of overlapping local approximation functions $Q_i(\mathbf{p})$ that are blended together using nonnegative compactly supported functions $\phi_i(\mathbf{p})$ that sum up to 1 everywhere on a bounded Euclidean domain Ω.

$$f(\mathbf{p}) = \sum_{i=1}^{n} \phi_i(\mathbf{p}) Q_i(\mathbf{p}), \tag{2.14}$$

where each $\phi_i(\mathbf{x})$ is computed as:

$$\phi_i(\mathbf{p}) = \frac{w_i(\mathbf{p})}{\sum_{j=1}^{n} w_j(\mathbf{p})} \qquad \sum_i \phi_i(\mathbf{p}) \equiv 1 \qquad \forall \mathbf{p} \in \Omega \tag{2.15}$$

Indicator functions

The function f can also be defined in the form of an indicator function χ. The indicator or characteristic function of a subset A of some set X, maps elements of X to the range {0,1} (Folland, 2013) (see Figure 2.13), i.e.

$$\chi_A(x) := \begin{cases} 1 & if\ x \in A \\ 0 & if\ x \notin A \end{cases} \tag{2.16}$$

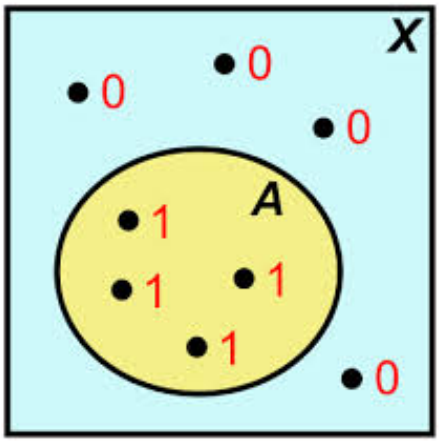


Figure 2.13: Indicator function χ_A of a subset A belonging to a Domain X

The (smoothed) gradient of χ corresponds to a vector field $\mathbf{V}$ formed by an integral over the (unknown) surface, which can be approximated by a summation over the oriented points (Kazhdan et al., 2006) (see Figure 2.14).

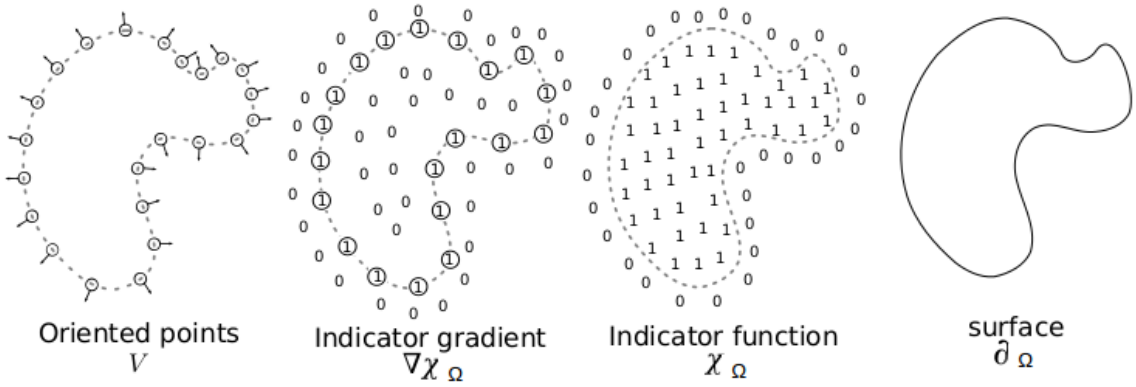


Figure 2.14: Four-stage illustration for Poisson surface reconstruction process. Taken from (Kazhdan et al., 2006)

To obtain a least-squares solution of $\triangle\chi = \mathbf{V}$, the divergence operator is applied to both sides, i.e. $\nabla \cdot \nabla\chi = \nabla \cdot \mathbf{V}$, resulting in a Poisson equation:

$$\triangle\chi = \nabla \cdot \mathbf{V} \tag{2.17}$$

In order to scale to larger resolutions, Kazhdan (Kazhdan et al., 2006) directly solves for χ in the spatial domain via a multi-grid approach, hierarchically solv-

ing for χ in a coarse-to-fine resolution manner using an octree representation. This method is called Poisson surface reconstruction (PSR) (Kazhdan et al., 2006) and uses the Euler-Lagrange formulation, where the minimum is obtained by solving the Poisson equation:

$$\triangle\chi = \nabla \cdot \nabla\chi = \nabla \cdot \mathbf{V} \tag{2.18}$$

The approach of Kazhdan (Kazhdan, 2005) transforms the problem into the frequency domain, where the Fourier transformation of $\triangle\chi$ and $\nabla \cdot \mathbf{V}$ results in a simple algebraic form for obtaining the Fourier representation of χ.

2.4 Composite model representation

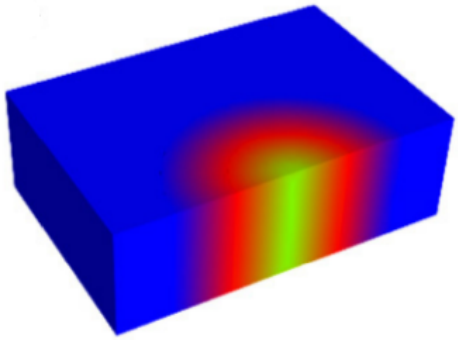
Despite the diversity of model representations, natural objects are represented with only one region that is in fact a solid modeling approach or solid modeling with functionally graded materials. However, most objects (either natural or man-made) contain multiple regions within complex volumetric structures. Such objects can be modeled using a composite model representation.

Intuitively, a composite model can be seen as a collection or group of regions with an interface separating them (see Figure 2.15), and each region belongs to a generic "material class" (Wang and Wang, 2005), which may be represented by any of volume models or surface models.

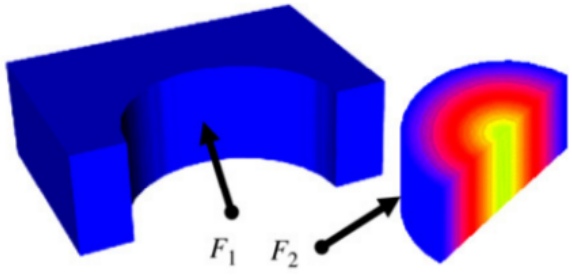
Mathematically, composite models can be described as:

$$\begin{gathered} O = \{O_1, O_2, \ldots, O_n\} \\ O_i = \{P = (P_g^{(i)}, P_m^{(i)}) \;\mid\; F^{(i)}(P_g^{(i)}) = P_m^{(i)},\; P_g^{(i)} \in \Omega_m^{(i)}, \quad 1 \leq i \leq n\} \end{gathered} \tag{2.19}$$

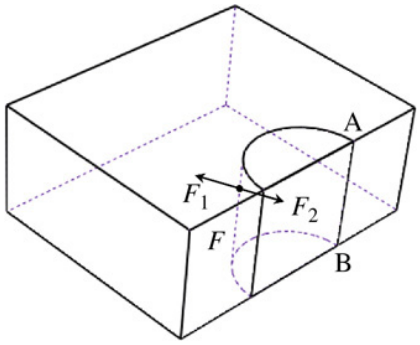
where *O* is a heterogeneous composite object, O_i is the i-th region belonging to *O*, $F^{(i)}(\cdot)$ is the i-th mapping function defined in the geometric subdomain $\Omega_{\mathrm{g}}^{(\mathrm{i})}$, $\Omega_m^{(i)}$ is the i-th material subdomain, and *n* is the number of regions.



(a) An object with complex material distributions.



(b) A manifold assembly model of (a).



(c) A non-manifold cellular model of (a)

Figure 2.15: Composite model representation of heterogeneous objects. Taken from (Kou and Tan, 2007).

Based on the idea of space partitions some researchers (Kumar et al., 1999), (Chen and Feng, 2004), (Shin and Dutta, 2001), (Kou et al., 2006), (Cheng and Lin, 2005) propose three strategies to model heterogeneous objects: assembly models, cell models and constructive models.

Assembly models are a natural and intuitive way to represent heterogeneous objects. The whole object is modeled as a set of components and each component is a region with a unique material distribution. Figure 2.15b shows an assembly model of the object in Figure 2.15a, which is decomposed into a homogeneous solid part, and a functionally graded material part. Kumar and Dutta (Kumar and Dutta, 1998), and Hu and Sun (Sun and Hu, 2002) proposed regularized operators (difference, intersection and union) and other operators to model the geometry of the assembly.

To represent heterogeneous composite objects using the assembly model, spa-

tial partitions must be properly established. There are at least two methods for accomplishing this. The constructive approach (see Subsection 2.2.3), which generates spatial divisions of the boundaries of existing primitives, and the rm-set representation. Kumar and Dutta (Kumar and Dutta, 1997) introduced the $rm-set$ and $rm-object$ for the representation of heterogeneous solid objects. The $rm-set$ representation is a direct extension of the $r-set$ model (see Definition 3), with additional material distributions included.

Thus, multi-region objects are represented as multiple $rm-set$, and each $rm-set$ is assumed to be homogeneous, however, different $rm-sets$ have different (homogeneous) material compositions. Such objects are compact in geometry representation (manifold representation) and the material distribution is assumed to be homogeneous.

Although manifold surfaces that usually enclosed volumes are appropriate for many natural forms, certain natural objects, such as hair, feathers, or leaves, are not easily representable as volumes. Unlike assembly representations, cell models use non-manifold cell representations for modeling space partitions, i.e. the geometry is modeled essentially with non-manifold boundary representations. For a non-manifold object the homeomorphic condition is not necessarily satisfied (see Figure 2.9c). In Figure 2.15c, non-manifold conditions occurring around the edge (border) AB, because the neighborhood of a point on the border AB is not homeomorphic to a disk in two-dimensions (Weiler, 1988), whereas for an assembly model all regions are topologically manifold.

In cell representation a topological element (for instance, a face) differs from the reference sides, i.e., the use of a face is actually the use of one of two sides of the face, and each side is oriented with respect to the same geometry of the face. Either of two sides of the face that share the same border element are called co-boundary (Weiler, 1988).

Some researchers believe that constructive modeling is an elegant way to build natural solid objects using a tree data structure with operations in the nodes and primitives in the leaves (Pasko et al., 2001), (Wang et al., 2011). However, traditional boolean operations are based on the premise that objects are homogeneous in material compositions and they are not directly applicable to heterogeneous object modeling tasks. To model heterogeneous composite objects traditional boolean operations need to be further extended.

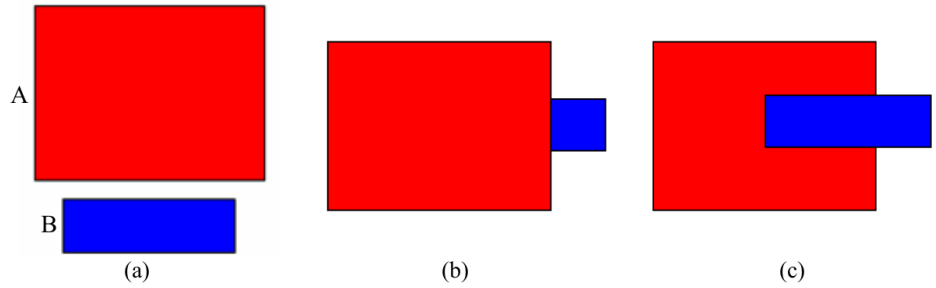


Figure 2.16: Extended boolean operations. (a) Input primitives A and B. (b) M_a dominant union. (c) M_b dominant union.

Most of the extended boolean operations represent composite objects through spatial decompositions. Complex objects are first decomposed into regions and different modeling strategies are then applied to such regions. For example, to model a heterogeneous composite object $C = A \cup^* B$, which is joined from two input homogeneous objects $A(G_a, M_a)$, and $B(G_b, M_b)$, where G and M represent geometry and material information for the heterogeneous composite object, the resultant object C is usually decomposed into 3 subsets (three-set-decomposition) (Sun and Hu, 2002), (Siu and Tan, 2002)

$$\begin{array}{c} S_1(G_a /^* (G_a \cap^* G_b), M_1) \\ S_2(G_a \cap^* G_b, M_2) \\ S_3(G_b /^* (G_a \cap^* G_b), M_3) \end{array} \tag{2.20}$$

where $\bigcup$*, $\bigcap$* and /* denote regularized boolean union, intersection and difference operator (Requicha, 1977).

Most of the extended boolean operations directly represent the material distributions of point set S_1 and S_3 with the same material distributions as M_a and M_b, and blend the material compositions with certain weighting ratios for point set S_2. General material definitions for each subset can be defined as:

$$\begin{array}{c} M_i = W_a^{(i)} M_a + W_b^{(i)} M_b \\ W_a^{(i)} + W_b^{(i)} = 1, \; i = 1, 2, 3 \end{array} \tag{2.21}$$

where $W_a^{(i)}$ and $W_b^{(i)}$ are material blending weights for M_a and M_b in subset i.

Sun and Hu (Sun and Hu, 2002) proposed M_a dominant and M_b dominant boolean operators, and apply constant blending weights on each decomposed subset (see Figure 2.16). For all M_a dominant operators, $W_a = 1$, $W_b = 0$; and for all M_b dominant operators, $W_a = 1$, $W_b = 0$. Siu and Tan (Siu and Tan, 2002) proposed similar heterogeneous insertion and immersion operators in their source-based scheme.

2.5 Summary

In this chapter, three fundamental approaches for three-dimensional modeling of natural heterogeneous objects was reviewed. The approaches were mainly focused on solid modeling, surface modeling, and composite modeling. In general way, existing heterogeneous object modelings extend traditional solid models with a two-step or a sequential scheme (Park et al., 2001), (Cavalcanti et al., 1997), (Siu and Tan, 2002), (Sun and Hu, 2002), (Kumar and Dutta, 1997). In the first step, only geometric information is defined which is typically a solid modeling process. In the second step, material information is incorporated and mapped onto the three-dimensional modeled geometries.

Exhaustive enumeration and hierarchical spatial partitioning schemes are two volume modeling strategies that allow explicitly representing just about any arbitrarily natural object; whether it is convex, concave or has interior holes. The precision is determined only by the size of smallest cells. However, a key limitation of such schemes is the large amount of memory used for the representation. This is because as a better approximation to represent a natural object is wanted, gradually a reduction in size of the base cell is needed and therefore an increase in number of cells is obtained.

Additionally, solid objects can be modeled through implicit functions Blinn (1982), Wyvill et al. (1986). Implicit functions are usually independent with resolution, i.e., they can generate a model at any given precision. Implicit solid modeling uses a sign Euclidean distance function that defines the natural object by assigning each point in space the shortest distance between the current point and any point belonging to the object surface. Although a set of implicit functions can be organized in a constructive solid geometry tree for generating shapes of

natural objects, the sign is used to distinguish between inside and outside when it is considered that the natural object has only one internal region (Pasko et al., 1995), (Ricci, 1973). However, this is not the case for most natural objects which are composed of different regions and complex structures.

Instead of modeling natural objects including their interior and boundary, only modeling their boundary can be useful. In fact, boundary representation (BRep) plays an important role in modern computer graphics since individual polygons, such as triangles, provide a reasonable approximation of surfaces of real world objects. When BRep representation is used, a high degree of realism is generally achieved by significant increasing the number of constituting polygons, and by increasing the complexity and sophistication of the algorithms involved in the rendering process. Unfortunately, BRep models are not very suitable when boolean and relational operations need to be applied on them. Besides, BRep often introduces additional restrictions if object's topology needs to be modified. Moreover, they are unable to represent the internal structure of the objects being modeled. This means that modeled objects are actually hollow, so there is no easy way to represent and explore its internal properties.

Unlike Brep representation implicit surface based approaches construct implicit functions from which the surface can be extracted as a level set. These methods can be broadly classified as global or local. Global implicit function methods commonly define the implicit function as the sum of radial basis functions centered at each of the input samples (Carr et al., 2001), (Turk and O'brien, 2002). Despite the fact that traditional methods based on radial basis function are elegant for solving the reconstruction problem, in practice they are difficult to use in large problems (Dense point clouds). The ideal basis functions are globally supported and non-decaying, so the solution matrix is dense and ill-conditioned. Local implicit function methods consider subsets of nearby points at a time. But, a blending scheme has to be considered for function continuity Ohtake et al. (2005). An alternative option is to use an implicit function approach based on indicator functions that combines benefits from both global and local methods. This approach was proposed by Kazhdan (Kazhdan et al., 2006) in the Poisson Surface Reconstruction method.

Approaches based on composite modeling tend to utilize manifold BRep and part-assembly representation to model heterogeneous engineering components. As-

sembly modeling is an approach used by CAD and product visualization computer software systems to handle multiple files that represent components within a product. Regions within an assembly are represented as solid models or surface models. While this is clearly a useful capability that follows naturally from part-based representations of man-made objects, it is not so clear how this capability can be extended to model natural objects, such as anatomical organs and structures.

In summary, implicit representations are more compact and exact in data representations, resolution independent, and can provide specific accuracy to efficient geometry queries (Kou and Tan, 2007). Moreover, recent studies have been shown more and more interest in heterogeneous object modeling using components implicitly represented (Pasko et al., 2001), (Adzhiev et al., 2002),(Biswas et al., 2004), (Wang and Wang, 2005), Yuan et al. (2012), whereas explicit representations are more efficient in material composition querying, but in practice explicit representations are very inefficient because the more precise the object is defined the more points or voxels should be set and the amount of data can be prohibited in this case. Furthermore, it is a common practice in solid modeling to decompose complex models into assemblies, which are collections of parts. Often this assembly relationship can be nested, so that assemblies contain other assemblies as parts, and so on. After an initial conceptual stage involving a top-down division of the natural object into assembly parts, such parts can be modeled independently, and then integrated into the final three-dimensional computer model.

Part II

Methods

Chapter 3

Modeling regions composing natural heterogeneous objects

3.1 Introduction

NOWADAYS, the basic and most widely used data structure to generate volume models from medical images is the voxel model (Sun et al., 2004), which is considered to be the most accurate representation of natural objects, including the human anatomy (Caon, 2004). Due to the fact that voxel models lack geometric representation, a meshing process is necessary for obtaining an object representation suitable for engineering design and analysis. However, such meshes generally have jagged edges caused by image voxelisation, that are unsuited for finite element simulations (see Figure 3.1). In this chapter, this issue is addressed as a reverse-engineering process along with an implicit surface reconstruction process for reconstructing regions composing natural heterogeneous objects.

In Section 3.2 the computer-aided reverse-engineering based method *(CAREM)* is introduced. This method takes as input a voxel model generated from medical images using a segmentation method (see Section 3.3). The voxel model is converted into a point cloud data through a Voxel-To-Point conversion process (see Section 3.4). Instead of directly triangulate the point cloud to form a faceted model (Starly et al., 2004), this point cloud is converted to an implicit surface representation applying an implicit surface reconstruction approach previous to

the meshing process (see Section 3.5). Some examples of modeling anatomical organs and structures using the *CAREM* method are presented in Section 3.6. Finally the chapter is summarized in Section 3.7.

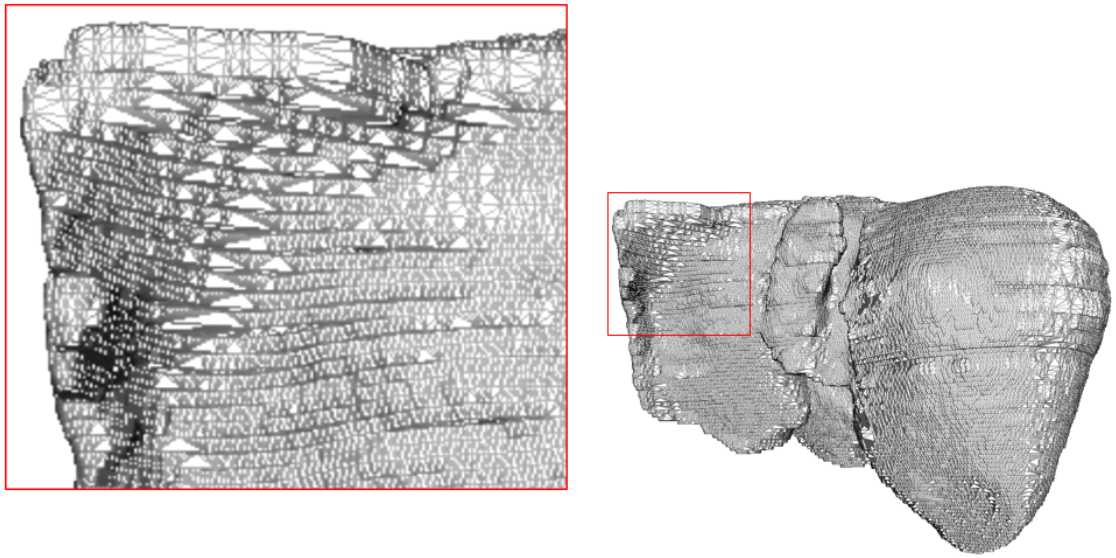


Figure 3.1: Bad shaped triangles generated by using the Marching Cubes algorithm.

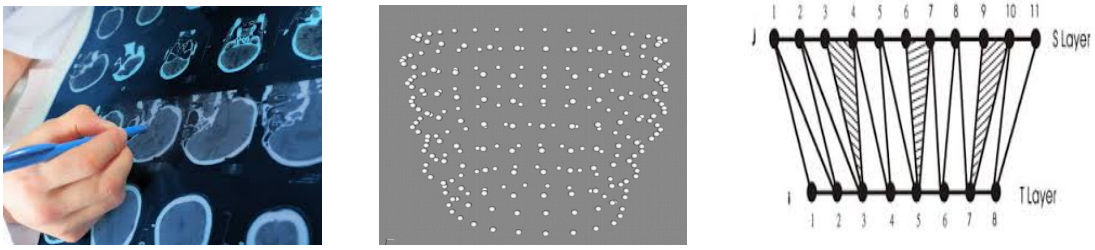
3.2 CAREM: A computer-aided reverse-engineering based modeling method

Modeling specific regions composing natural heterogeneous objects often starts from acquisition of volumetric data using an appropriate medical imaging modality such as computer tomography. The volumetric domain (CT image) D (see figure 3.6), expressed by a trivariate function as

$$\begin{gathered} f : D \subset \mathbb{R}^3 \rightarrow \mathbb{R} \\ (x_i, y_j, z_k) \rightarrow f(x_i, y_j, z_k),\ i, j, k = 1, ..., n \end{gathered} \tag{3.1}$$

is partitioned into a set of regions of interest $\Omega_1, \Omega_2, ..., \Omega_n$. The goal is to construct three-dimensional computer models of such regions of interest that must be topologically correct, and geometrically smooth

Currently, there are two techniques used to generate surface models from medical images (Moustakides et al., 2000): The first approach builds the surface from serial parallel contours extracted from sequential medical images (see Figure 3.2).



(a) Liver. (b) Left kidney. (c) Right kidney.

Figure 3.2: Surface mesh from serial parallel contours.

The medical images are segmented into regions of interest (see Figure 3.2a), and edge points around the boundary are extracted and ordered (Knopf and Al-Naji, 2001) (see Figure 3.2b). Finally, points from different layers are connected to build the surface triangulation (see Figure 3.2c). Accuracy of the fitted surface model depends largely on extracted edge points and the the function used to fit curves to those contours.

The second option is to convert the initial description of the voxel model to a boundary surface triangulation by using the Marching Cubes algorithm (Lorensen and Cline, 1987). This algorithm takes eight neighbor locations from adjacent segmented images at corners of each cube within the base mesh, and determines the triangles needed to represent the part of the isosurface that passes through each cube (see Figure 3.3).

Material information related to the vertex inside the cube is obtained directly from two segmented slices. If a vertex has a different identification (ID) from neighboring vertices, a boundary surface must exist between this vertex and the others in order to separate different region materials. There are 256 different combinations of region materials a cube might have. These combinations can be reduced to 15 patterns using symmetry (see Figure 3.3). Individual triangles are then merged into the desired surface. However, as segmented data from medical images is, in general, huge, jaggy, not properly ordered and noisy, the Marching

cubes triangulation reflects these defects (see Figure 3.1), which strongly impede the computational efficiency if it is used directly for simulation purposes (Lin et al., 2001).

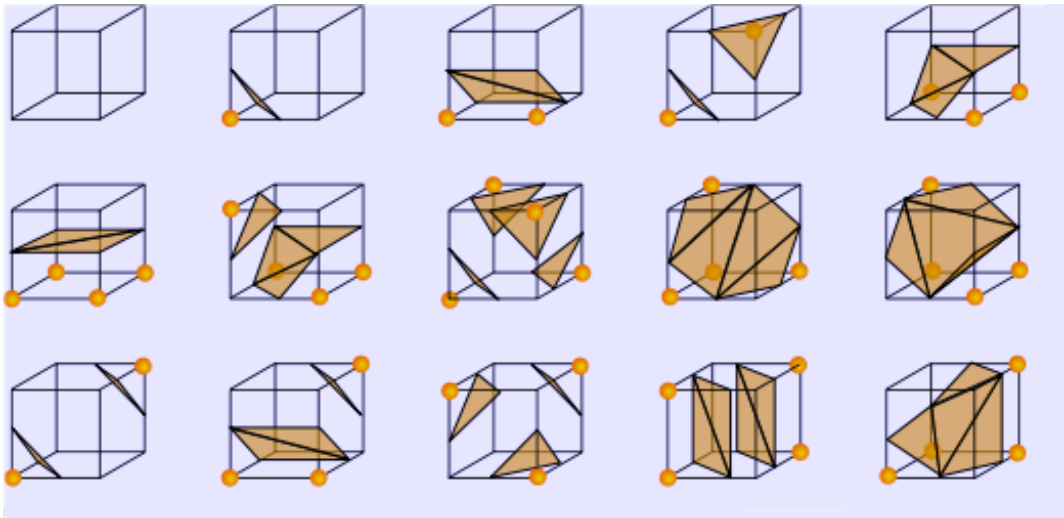


Figure 3.3: Marching Cubes combinations reduced to 15 patterns using symmetry. Taken from (Lorensen and Cline, 1987)

3.2.1 Algorithm

To overcome these issues a computer-aided reverse-engineering based modeling (*CAREM*) method is proposed along with an implicit surface reconstruction process for modeling different regions composing natural heterogeneous objects (see Algorithm 3.1).

Algorithm 3.1 The *CAREM* method

Input: A medical image containing the region of interest Ω_i.
Output: A closed manifold three- dimensional model of the region Ω_i.
1: Create $VoxelModel \leftarrow ExtractRegion(\Omega_i)$.
2: Generate point cloud $\mathbf{V} \leftarrow VoxelToPointCovertion(VoxelModel)$.
3: Compute $\chi \leftarrow ImplicitSurfaceReconstruction(\mathbf{V})$.
4: Extract closed manifold surface model $SurfaceMesh(\chi)$.
5: Extract closed manifold volume model $VolumeMesh(\chi)$.

Reverse engineering is an important method in constructing CAD models from existing physical objects or devices to determine its structure or function (Wang,

2010). However, reverse engineering in CAD applications has stronger requirements than reverse engineering with applications in computer graphics: it aims at providing valid, accurate and consistent models. Moreover, modeling of natural objects differs vastly from modeling of engineering components. Natural objects tend to have more complicated geometries than mechanical objects (Moustakides et al., 2000).

Figure 3.4 shows a flow chart of the *CAREM* algorithm for computing a closed manifold triangular mesh as a boundary surface of specific regions of interest Ω_i. The *CAREM* method consists of three mayor processes (see Figure 3.4): voxel model generation from medical imaging data (see Section 3.3), voxel to point conversion (see Section 3.4), and three-dimensional reconstruction (see Section 3.5).

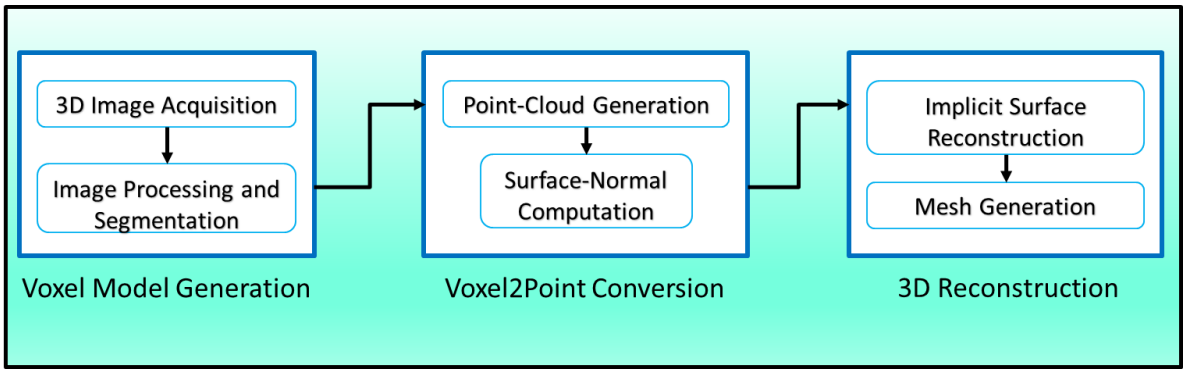


Figure 3.4: Flow chart for applying the *CAREM* method for region modeling

Next, these three processes are described in details.

3.3 Voxel model generation process

In the first process anatomical organs and structures are measured using medical imaging scanners (see subsection 3.3.1). Medical acquisition modalities are the most important source of anatomical and functional information, which is indispensable for today's clinical research, diagnosis and treatment, and is an integral part of modern health care. Current imaging modalities provide large amount of data which are capable of generating representations of natural objects, including their interior (Stytz et al., 1991).

With the availability of medical data came the need to improve and automate medical image processing techniques. Due to noise and lots of irrelevant information in medical data, image processing is necessary. Filtering and smoothing techniques are usually used to improve and simplify images through reducing noise and keeping important features (Schroeder et al., 2006). Additionally, it is also necessary to identify which type of organs and structures are present in the data space, and the precise location of edges between different tissue types. This process called image segmentation yields as result voxel models (see subsection 3.3.2).

3.3.1 Three-dimensional image acquisition

An ideal three-dimensional computer model would provide an accurate and efficient approximation for huge data sets, as well as the exact evaluation of function values and gradients which are required for high-quality visualization and physical simulation. When building models based on medical image data, there are two main components related to accuracy: data collection and data post-processing.

Data collection is the field where a radiologist is concerned with and reliable data acquisition depends on variables, such as choice of appropriate medical image modality, a choice of a 2D image reconstruction algorithm and patient's dosage. While computer scientists have little control over data they receive for post processing, their research can greatly improve medical data processing and enhance accuracy of obtained medical data through a wealth of medical image processing and reconstruction techniques.

Computer tomography (CT) and magnetic resonance imaging (MRI) are two types of medical imaging scanners. CT is more effective in modeling hard tissues and sharply defined density changes, such as interface between bones and soft tissues, whereas MRI is better in differentiating soft tissues of similar density. However, the resolution of MRI is consistently worse than CT. Next, computer tomography is briefly described.

Computed tomography

Computer tomography was the first imaging modality to offer detailed view of the internal anatomy in three-dimensions (Caon, 2004). The foundation of CT image generation is the measurement of x-ray attenuation along a line between an x-ray source and an x-ray detector (Prince and Links, 2006). A 2D cross-sectional image is then reconstructed following the collection x-ray attenuation coefficients along all lines within the cross-section. In fact the term computed tomography alone CT is often used to refer to X-ray CT, although other tomography types exist, such as positron emission tomography (PET) and single-photon emission computed tomography (SPECT).

Seven generations of basic CT scanners have been introduced, with each consequent model introducing faster acquisition of data and higher image quality. For instance, helical CT was developed in the 1980s to address the growing need for fast volume data acquisition (Evans, 2008). In such scanners, x-ray source and detectors rotate continuously. The patient's table is set into forward motion sliding the patient through the source-detector plane (see Figure 3.5). As result, as the name of the scanner implies, the position of the source carves out a helix with respect to the patient (Evans, 2008).

In CT, a number of 2D radiographies (slices) are acquired by rotating the X-Ray tube around the patient's body (see Figure 3.5). The x-ray tube makes a short burst of x-rays that propagate through a cross-section of a patient. The detector detects the exit beam intensity which is integrated along a line between the x-ray source and each detector (Evans, 2008). Thus, the integrated x-ray intensity at any given detector is given by (Evans, 2008):

$$I_d = \int_0^{E_{max}} S_0(E) E_{\exp} \left[- \int_0^d \mu(s; E) ds \right] dE \tag{3.2}$$

where $S_0(E)$ is the x-ray spectrum and $\mu(s; E)$ is the linear attenuation coefficient along the line between the source and the detector.

Full three-dimensional image is built using image reconstruction algorithms (see Figure 3.6b), which depends on the scanner's geometry. The most commonly reconstruction algorithms used are parallel-beam, fan-beam, helical-scan, or cone-

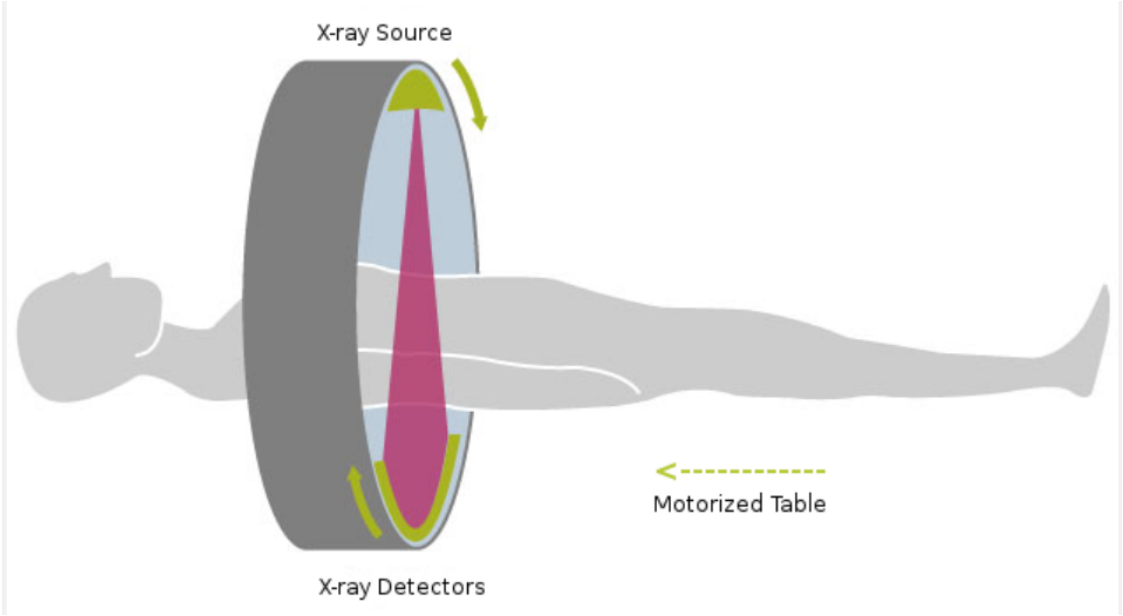
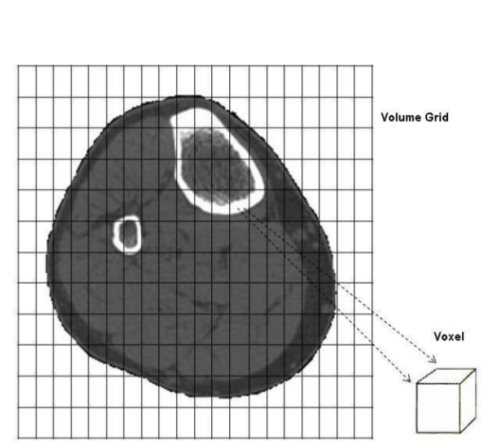
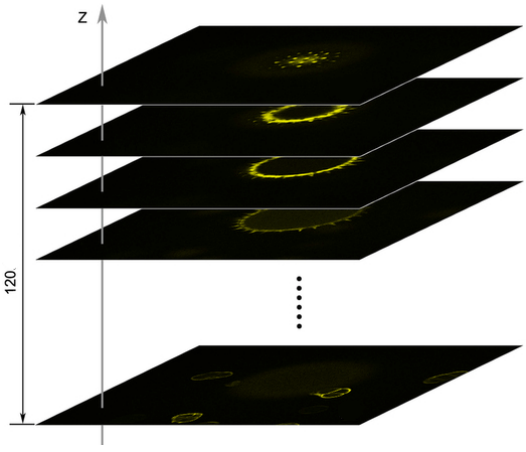


Figure 3.5: Schematic representation of the process involved in 3D computed tomography.

beam, and filtered back-projection (Evans, 2008). Three-dimensional objects are traditionally reconstructed as a stack of 2D slices. Figure 3.6 illustrates the relationship between 3D and 2D.



(a) A 2D projection taken from multiple views around the object (see Figure 3.5).



(b) A 3D image reconstruction of an object from a series of 120 2D axial projections.

Figure 3.6: Relation between 2D and 3D projections.

Hounsfield scale

Effective energy varies between different CT scanners and is dependent on the type of x-ray tube used. This in turn means that the same object will produce a different value of μ on different scanners. Thus, to compare medical data produced from different CT scanners, numbers are calibrated and expressed as CT values (see Table 3.1).

Table 3.1: Hounsfield units of various tissues (Webster, 1988).

Tissue	CT number (HU)
Bone	1000+
Hemorrhage	60-110
Liver	50-80
Muscle	44-59
Blood	42-58
Gray matter	32-44
White matter	24-36
Heart	24
Cerebrospinal Fluid	0-22
Water	0
Fat	-20 to -100
Lung	-300
Air	-1000

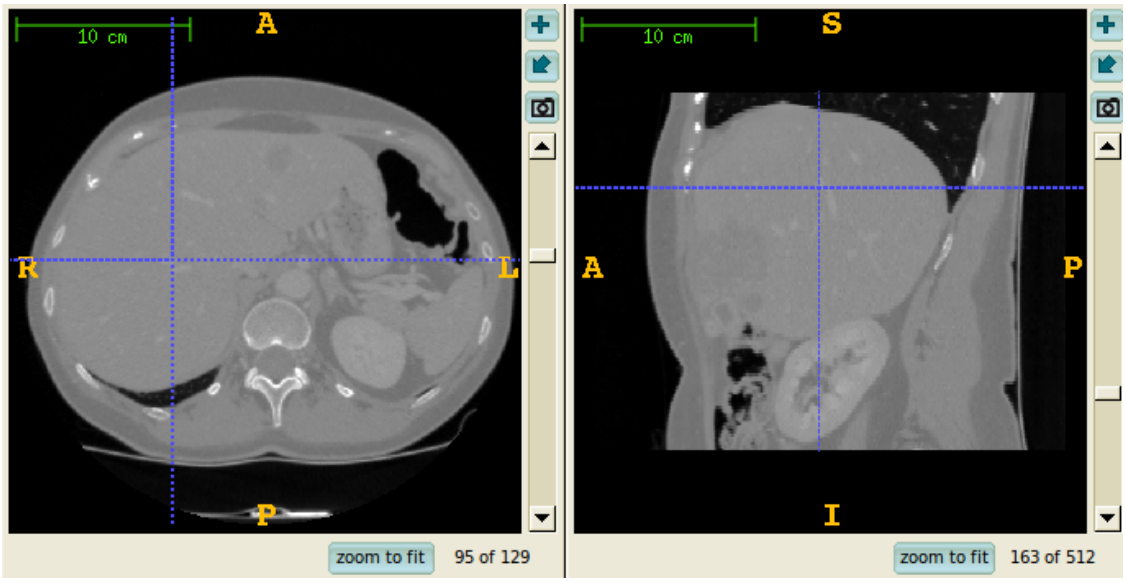
A CT scanner measures and reconstructs the value of coefficient μ at each pixel. CT value of a region material, given the effective energy of the source beam, is computed from the linear attenuation coefficient of the region material and is calculated using the following relation:

$$CT\ number\ h = \left[\frac{\mu m - \mu w}{\mu w}\right] * a \tag{3.3}$$

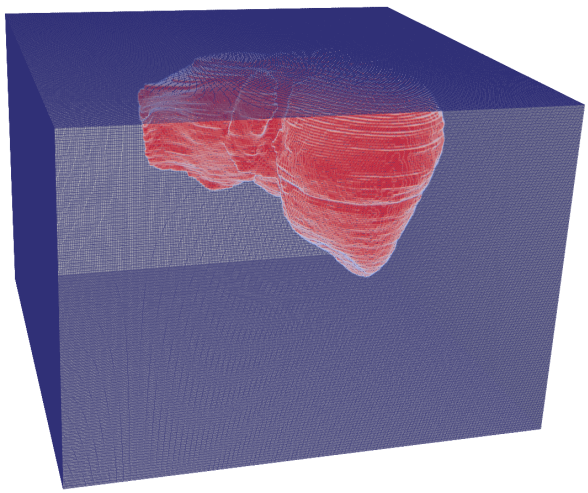
where μm and μw are linear attenuation coefficients for the region material and water, respectively, and a is a scaling factor.

CT values are expressed in Hounsfield units (HU) (see Table 3.1). Hence, Hounsfield

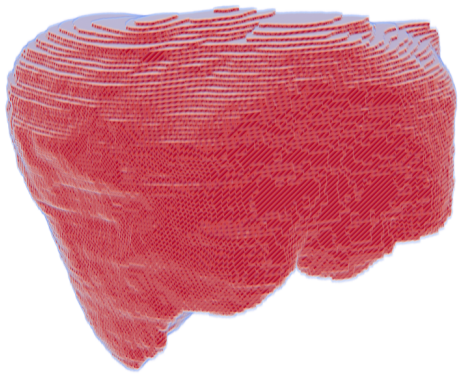
scale is a linear transformation of original coefficient measurement where radiodensity of distilled water at standard pressure and temperature is defined as zero. Since $\mu = 0$ for air, it is h = -1000 HU for air. Typically, largest Hounsfield values are found in the bone, where $h \approx 1000$ HU (see Table 3.1).



(a) Raw data of a chest CT



(b) Liver segmentation from Figure 3.7a, located in the image domain.



(c) Volume visualization of the segmented human liver.

Figure 3.7: Image segmentation of a human liver.

3.3.2 Image segmentation

In order to successfully build a three-dimensional computer model of specific regions composing natural objects, it is crucial to accurately identify and extract such regions from medical images. In the context of medical imaging, raw data has no structural descriptions beyond intensity values (see Figure 3.7a). So data have to be segmented into homogeneous regions in order to distinguish objects of interest against their surroundings within a process called segmentation (see Figure 3.7b).

Image segmentation is the process of identifying separation of different objects within the data set. Anatomical organs and structures can be extracted by manual or partially automated segmentation methods. However, segmentations have to be anatomically meaningful to be used in further applications, such as quantitative shape analysis, visualization and subsequent model building (Prince and Links, 2006) (see Figure 3.7c).

Thresholding

Thresholding is a simple technique for images that contain regions of interest on a background of different, but uniform, brightness. Each pixel is compared to a threshold T: if its value is higher than T, the pixel is considered to be foreground and is set to white, and if it is less than or equal to T it is considered background and set to black. Thus, the success of thresholding depends critically on the selection of an appropriate threshold (see Figure 3.8).

In some situations an upper and a lower threshold can be defined, separating the image into structures of interest and background. This method works well, for instance, for bone segmentation from CT scans since bone tissue attenuated significantly more during image acquisition and is therefore represented by much higher values on the Hounsfield scale compared to soft tissues.

(a) An original lung CT image.

(b) Result from applying a threshold equal to -300 to Figure 3.8a.

Figure 3.8: Threshold applyed on a lung CT image.

Region-based methods

Region-based methods find connected regions based on some similarity of pixels within them. The objective is to produce connected regions that are as large as possible. Region growing is a button-up procedure that starts with seed pixels, and then grows regions by adding neighboring pixels that have similar properties to the seed. Connectivity (4- or 8-) is used to define what are the neighboring pixels. Figure 3.9 shows the segmentation of a three-dimensional CT image. The image was segmented using two seed points placed in different lungs.

Although medical image segmentation is an established and mature field, automating this task remains a difficult issue due to the sheer size of datasets and the inherent complexity and variability of the anatomical shapes of interest (Kass et al., 1988), (De Santis et al., 2007). However, manual segmentation is a laborious and time consuming process that is subject to error. Thus, a semi-automatic approach is used in the *CAREM* method for the voxel model generation process. Indeed, all voxel models of anatomical organs and biological structures used in this research work were segmented using a region-growing method based on an trial and error calculated threshold from the Insight Segmentation and Registration Toolkit (ITK) (www.itk.org).

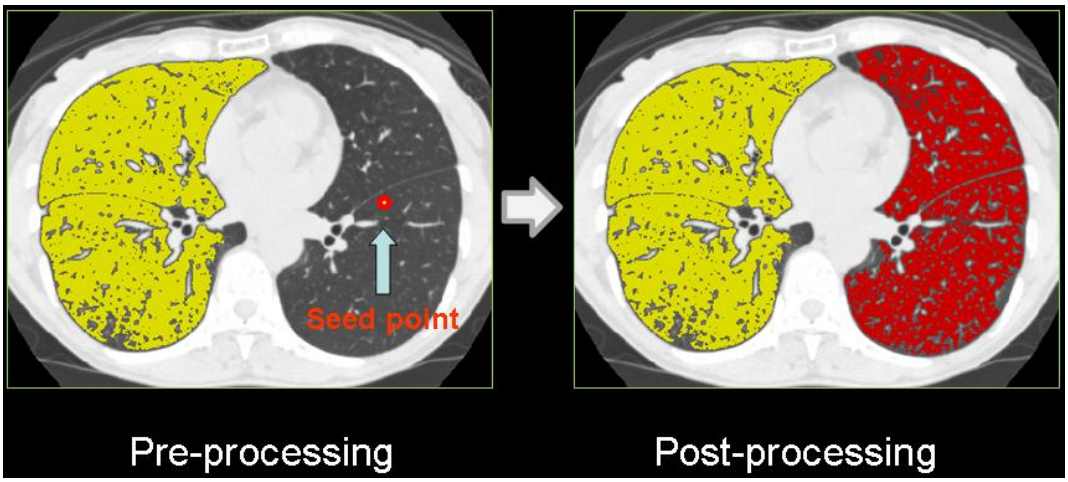
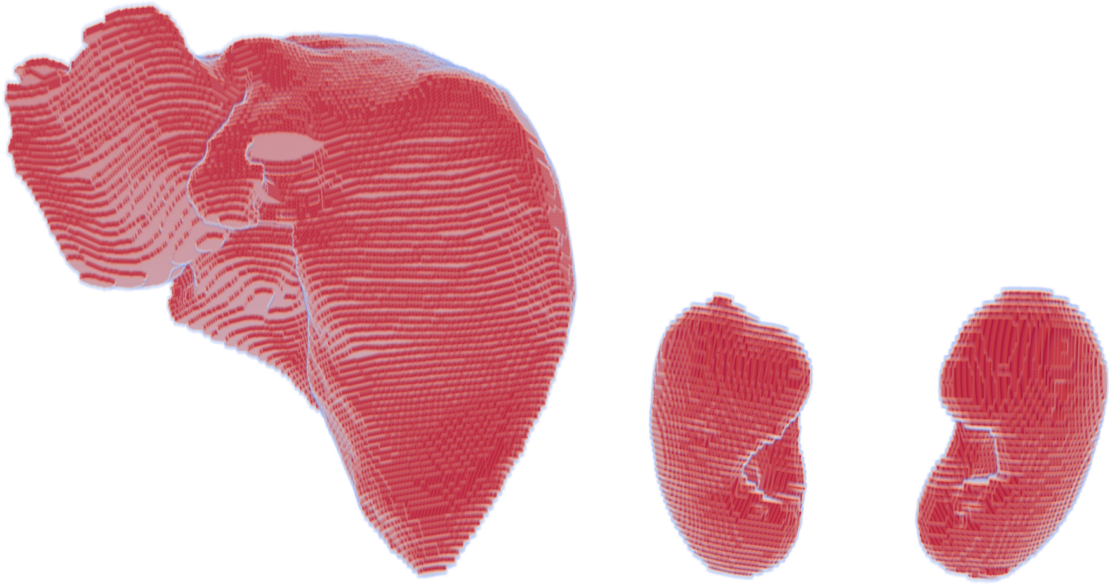


Figure 3.9: Result from applying the region-growing method to Figure 3.8a

For example, Figure 3.10 depicts voxel models of three human organs: a liver (see Figure 3.10a), a left kidney (see Figure 3.10b), and a right kidney (see Figure 3.10c), all of them obtained from a patient image provided by IRCAD (IRCAD, 2015). To segment these organs a threshold T shown in Table 3.1 and a region growing algorithm were applied.



(a) Liver. (b) Left kidney. (c) Right kidney.

Figure 3.10: Some voxel models obtained using the *CAREM* method.

3.4 Voxel-To-Point conversion process

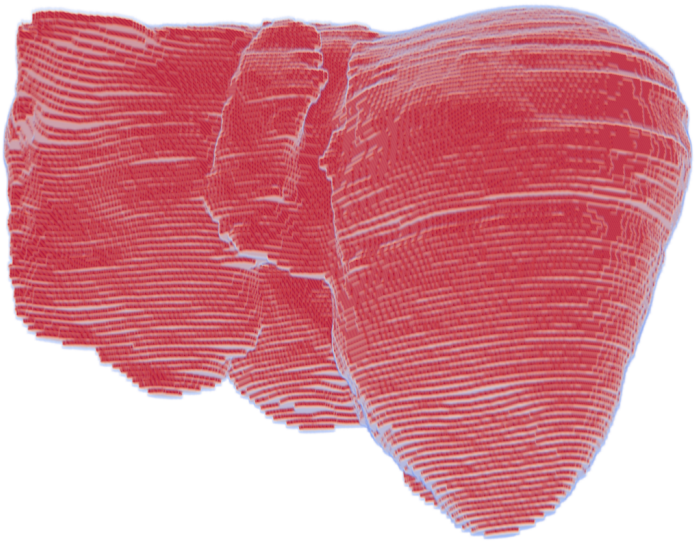
In Subsection 3.3.2 the medical image from a given natural heterogeneous object Ω is segmented into n regions with uniform material. The result from applying the segmentation process are voxel models. Although voxel models are currently considered to be the most accurate representation of the human anatomy, such volume descriptions tend to impose high demands on memory due to their three dimensions. Additionally, voxel models lack geometric representation, which would make them unsuitable for bioengineering analysis and simulation. Therefore, efficient representations of large voxel volumes are essential for recovering the object's geometry.

In Section 3.5 an implicit surface reconstruction approach is proposed previous to the meshing process. However, the main requirement of all implicit surface reconstruction methods is a set of three-dimensional points P that sample the surface of the physical object. In the field of computer graphics the set of points P is usually obtained by laser range-finders, mechanical touch probes and computer vision techniques, such as depth from stereo (Ohtake et al., 2005). As the result of the first step of the *CAREM* method is a set of voxel models, a process for extracting boundary points and surface normals from these models has to be developed. This process is called Voxel-To-Point conversion process (see Figure 3.4).

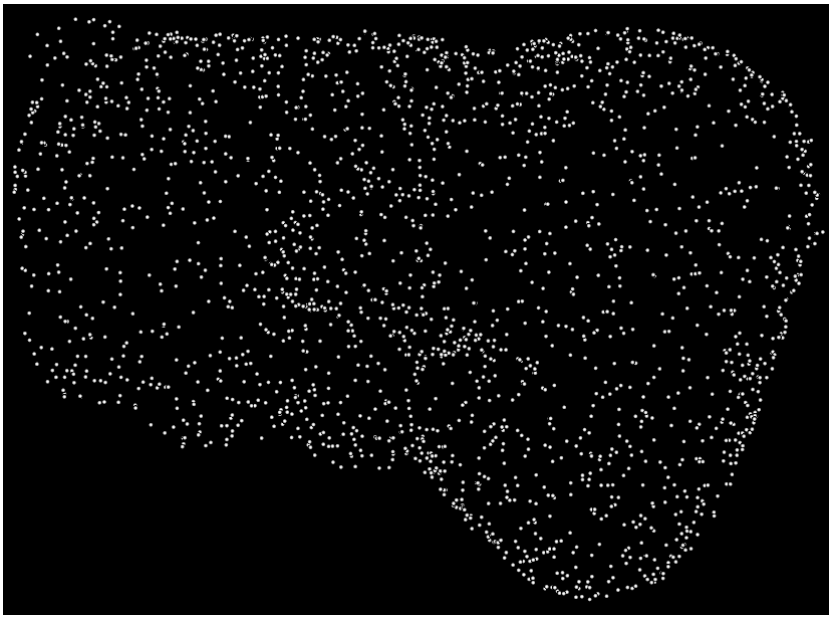
The Voxel-To-Point conversion process starts with the boundary surface triangulation produced by the Marching Cubes algorithm (Lorensen and Cline, 1987) (see Subsection 3.4.1). Then the information related to the position of each vertex is taken along with the normal vector calculated using a 6-point central difference gradient operator (see Subsection 3.4.2).

3.4.1 Point cloud generation

The voxel model obtained in the previous section is considered to extract the required boundary points P_i. The voxel model is processed by taking each voxel cell in turn. If the eight voxel values evaluated at the voxel cell corners are identical, the cell is located inside or outside the region.



(a) Voxel model considered for extracting boundary points.



(b) Point cloud generated from 3.11a.

Figure 3.11: Extraction of boundary points (only a 10% is shown).

However, if one or more values are different from their neighbors, the cell crosses the object boundary. In that case, a boundary point is created at the center of the voxel cell indicating that the region surface runs near this point (see Figure 3.11b).

In Chapter 5 is shown the number of points generated by this procedure. As

this number tends to be rather large, a decimation procedure is frequently used. However, experiments were carried out without this decimation in order to highlight the robustness of the *CAREM* method.

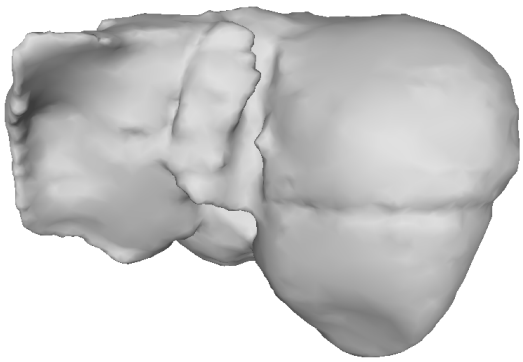
3.4.2 Surface-normal computation

In some occasions, knowing where a surface is heading towards might be useful information. Indeed, the three-dimensional reconstruction process of the *CAREM* method assumes a set of normal vectors $\mathbf{N}$ along with the point cloud $\mathbf{P}$, i.e., for each $\mathbf{p}_i \in \mathbf{P}$ there is an accompanying normal $\mathbf{n}_i \in \mathbf{N}$. In fact, the implicit surface reconstruction process requires as input data a set of points $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$ sampled from the three-dimensional segmented image with corresponding normals $\mathbf{N} = \{\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_n\}$.

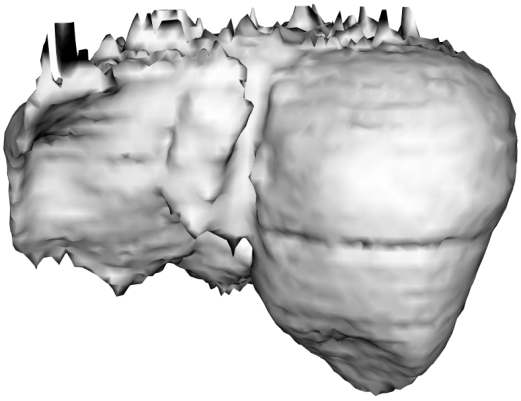
Surface normals are typically computed directly from the point cloud. A popular and simple method for computing the normal at a given point $\mathbf{p}$ is to perform principal component analysis (PCA) in a local neighborhood of $\mathbf{p}$ (Hoppe et al., 1992). More specifically, if a local neighborhood of $\mathbf{p}$ is denoted by N_p then the most basic way to apply PCA is by an Eigen analysis of the covariance matrix:

$$C_p = \sum_{\mathbf{q} \in N_p} (\mathbf{p} - \mathbf{q})(\mathbf{p} - \mathbf{q})^{\mathbf{T}} \tag{3.4}$$

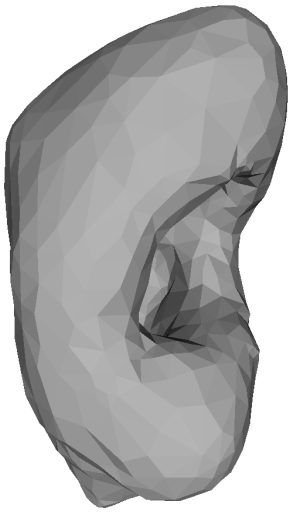
The eigenvector of C_p associated with the smallest eigenvalue defines un-oriented normals. Note that if the smallest eigenvalue is 0, then the region defined by $\mathbf{p}$ and N_p is planar, since eigenvectors associated with the second largest eigenvalues capture all of the variance in the data. However, the issue is to find a local neighborhood of points, where the neighborhood must be small enough to accurately represent a point's tangent space. Likewise, the neighborhood's scale must be proportional to the sampling density at the point. This could lead to bad oriented surface normals that impacts directly in the accuracy of the final reconstruction (see Figure 3.12b).



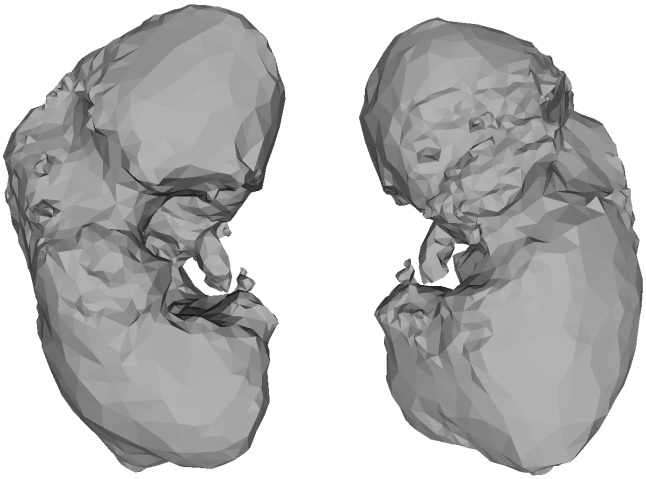
(a) Surface reconstruction of a liver with good oriented surface normals.



(b) Final surface reconstruction with bad oriented surface normals.



(c) Surface reconstruction of a kidney with good oriented surface normals.



(d) Two view of the kidney surface reconstruction with bad oriented surface normals.

Figure 3.12: Effects of oriented surface normals over the surface reconstruction method.

As the estimation of sampling density is itself a challenging issue, particularly when faced with nonuniform sampling the information related to normals is directly calculated from segmented images. However, there are some issues that makes the estimation of surface normals from segmented data especially difficult:

- On account of the data's binary nature, there is always an abrupt change

in data values at object's surfaces, resulting in reconstruction artifacts due to under-sampling.

- Once the volume is segmented, any prior information about the object's surface is discarded and estimating a new surface from binary samples proves the best option.
- Additionally, normals must have consistent directions, either pointing inside or outside of the surface (see Figure 3.13).

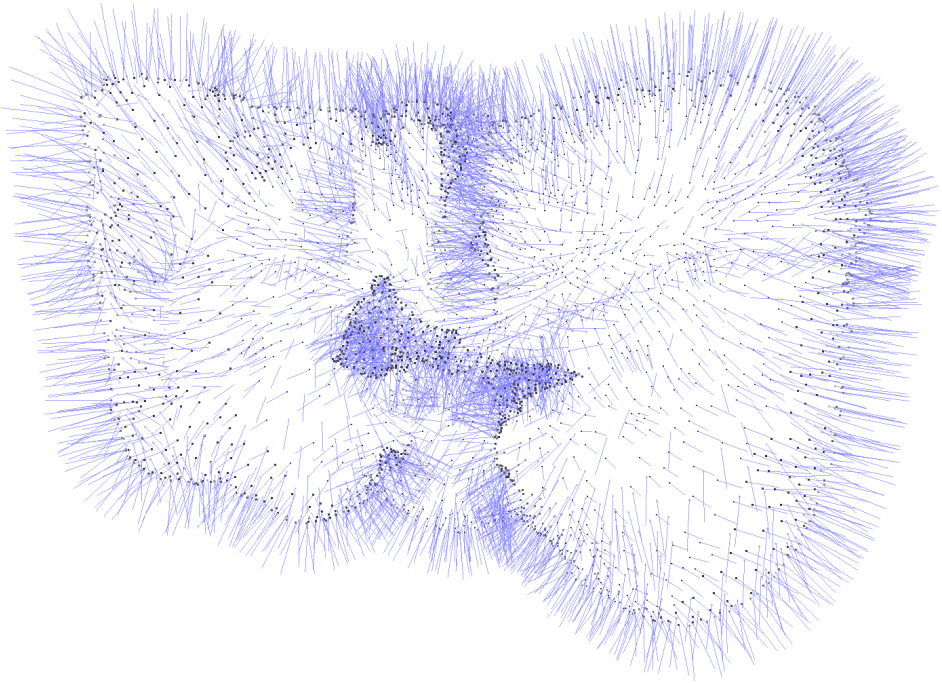
Thus, when the tangential surface of a very specific voxel is wanted, it is necessary to take a look at its surroundings. Hence, a radius r of voxels is defined, in which surface voxels are specified . Within the defined radius, all first order surface voxels will be derived.

For the present work, the radius r=1 and the oriented normal n to the surface is given by (Hartmann, 1999)

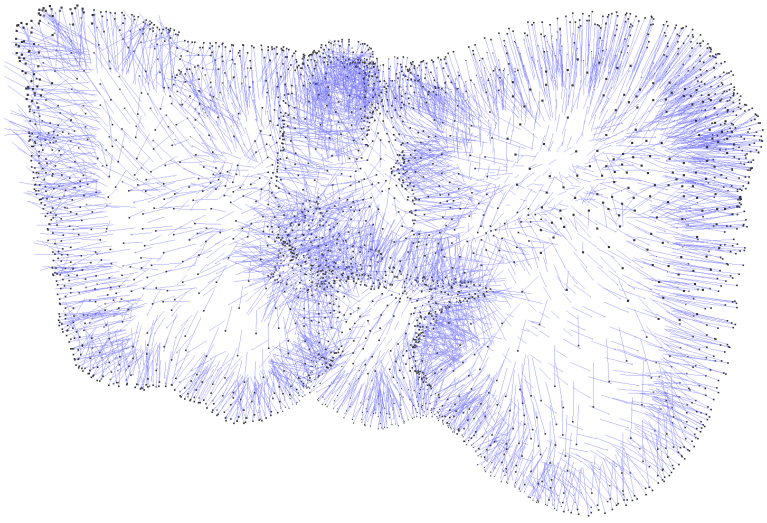
$$\mathbf{n} = \frac{\nabla f}{\parallel \nabla f \parallel}, \quad \nabla f = \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \tag{3.5}$$

Thus, the surface normal information is calculated using a 6-point central difference gradient operator of the form $\mathbf{n} = (n_x, n_y, n_z) = (d(x+1, y, z) - d(x-1, y, z), d(x, y+1, z) - d(x, y-1, z), d(x, y, z+1) - d(x, y, z-1))$.

The sign of the surface normal vector, calculated as above, is not always clear. So, it is not really known whether the surface normal vector is facing inside or outside the model. As the result of the segmentation process is a voxel model, not only a shell, it can be easily solved this ambiguity. Since the only undetermined property is the vector sign, voxels on either side can be checked. On one side, there should be object's voxels presented, on the other side, they should be background voxels. This assumption is valid since only surface voxels are considered. Figure 3.13a depicts a set of oriented surface normals pointing outside of the point cloud.



(a) Outward pointing surface normals.



(b) Inward pointing surface normals.

Figure 3.13: Generation of oriented (outward and inward pointing) surface normals from MC triangulation of the liver (only a 10% is shown).

3.5 Three-dimensional reconstruction process

In this section, the set of points $\mathbf{P} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\} : \mathbf{s}_i = (\mathbf{s}_i, \mathbf{n}_i)$ obtained in Section 3.4 is defined as a continuous real function of point coordinates $y = f(\mathbf{p})$, where $f(\mathbf{p}) \leq 0$ define points that belong to the object and $f(\mathbf{p}) > 0$ define points outside the object (Kou and Tan, 2007) (see Subsection 3.5.1). In a second step (see Subsection 3.5.2) a meshing algorithm is used to approximate $f(\mathbf{p}) = 0$ by a surface mesh or a volume mesh.

3.5.1 Implicit surface reconstruction

Even though large amount of research has been done in the past on reconstruction of smooth surfaces from point clouds, this is still a challenging problem (see Subsection 2.3.2). However, based on the observation that segmented data consists of one value for points inside the object and a second value for points outside of the object, the inward pointing normal field of the boundary of a region can be interpreted as the gradient of the region's indicator function χ_Ω. The function χ_Ω is often defined to have value 1 inside and value 0 outside the model (see Figure 3.14-left).

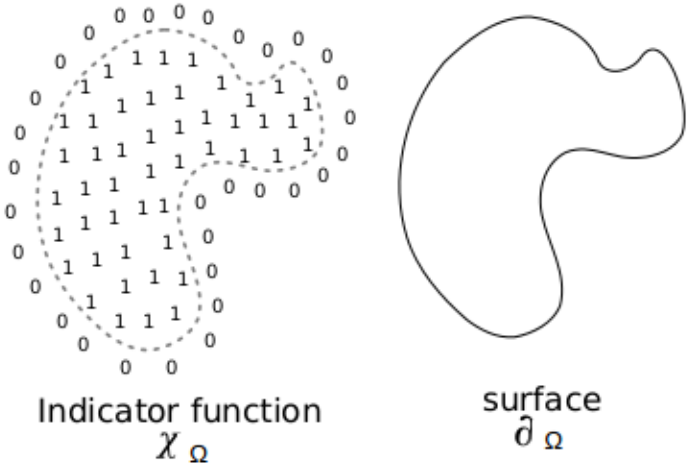


Figure 3.14: *CAREM* surface reconstruction process using the Poisson surface reconstruction method. Taken from (Kazhdan et al., 2006)

Kazhdan (Kazhdan et al., 2006) showed that the isosurface induced by the indicator function defines the reconstructed surface ∂_Ω (see Figure 3.14-right).

Algorithm 3.2 Constructing the octree structure

Input: An oriented point set $\mathbf{V} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\} : \mathbf{s}_i = (\mathbf{p}_i, \mathbf{n}_i)$.
Output: An octree structure.
for all $\mathbf{s_i} \in \mathbf{V}$ do
  $o \leftarrow OctreeRoot$
  for $d = 0\, to\, MaxDepht$ do
    $k \leftarrow OctantIndex(\mathbf{p}_i)$
    $o \leftarrow DetOrCreateChild(o, k)$
    RefineNeighbors(o)
  end for
end for

In Section 3.4 a set of points $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$ from a segmented voxel model was obtained along with their corresponding normals $\mathbf{N} = \{\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_n\}$. These two vector are combined in a point set $\mathbf{V} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$, where $\mathbf{s_i} = (\mathbf{p}_i, \mathbf{n}_i)$. Here, the basic idea is to reconstruct the surface from $\mathbf{V}$, estimating the indicator function χ by ensuring its gradient as-close- as-possible to the normal field of $\mathbf{V}$ in terms of least-squares using $\| \nabla\chi - \mathbf{V} \|_2^2$. Thus, the smoothed gradient of χ corresponds to a vector field $\mathbf{V}$ formed by an integral over the (unknown) surface, which can be approximated by a summation over the oriented points (Kazhdan et al., 2006).

If the divergence operator is applied to this problem, then this is equivalent to solving the following Poisson equation:

$$\nabla \cdot \nabla\chi = \triangle\chi = \nabla \cdot \mathbf{V} \tag{3.6}$$

Since the indicator function (and therefore its gradient) only contains high-frequency information around the surface of the region, an adaptive, multi-resolution basis is used to represent the solution. Specifically, an octree $\mathcal{O}$ is adapted to the point samples (see Algorithm 3.2), and then a function space is defined by associating a tri-variate B-spline F_o to each octree node $o \in \mathcal{O}$.

The B-spline F_o is translated to the node center and scaled by the node size, and the span $\mathcal{F}$ of the translated and scaled B-splines defines the function-space over which the Poisson equation is solved. (see also Szeliski and Lavallée (1996)).

Expressed in this basis,

$$\chi(\mathbf{p}) = \sum_{o \in \mathcal{O}} \chi_o F_o(\mathbf{p}) \quad and \quad \vec{V}(\mathbf{p}) = \sum_{o \in \mathcal{O}} \mathbf{V}_o F_o(\mathbf{p}) \tag{3.7}$$

the Poisson equation reduces to the sparse symmetric system

$$Lx = b \tag{3.8}$$

where $x = \{x_o\}$ and $b = \{b_o\}$ are $|\mathcal{O}|$-dimensional vectors of octree coefficients, the matrix entries are the inner products $L_{o,o'} =< F_o, \triangle F_d >$, and the divergence coefficients are

$$b_o = \sum_{d \in \mathcal{O}} < F_o, \nabla \cdot (\mathbf{V}_o, F_d) > \tag{3.9}$$

The assembly of matrices requires $O(M(N/M)^2$ operations, and to solve all linear systems requires $O(M(N/M)^3$ operations. Concerning the evaluation, the number of operations required is in $O(M + N/M)$.

Solving the Poisson equation, the function χ is obtained that approximates the indicator function. The surface S is then represented by the zero crossing of the indicator function, or with a suitable isovalue, typically the average or median value of χ evaluated at all of the input points. However, the function χ can differ from the true indicator function due to the point sampling density that is approximated during octree construction. To mitigate errors related to this step, Kazhdan in (Kazhdan et al., 2006) adjusts the implicit surface by globally subtracting the average value or median of the function χ at the input samples.

The implicit surface described as the zero level set of the indicator function is then input to the meshing algorithm. This mesh is either a surface mesh approximating the boundary of the modeled object or a volume mesh of it.

3.5.2 Mesh generation

Although resulting implicit based model could be rendered directly as a legal isosurface object in a raytracing engine POV-Ray, it has been chosen to export it

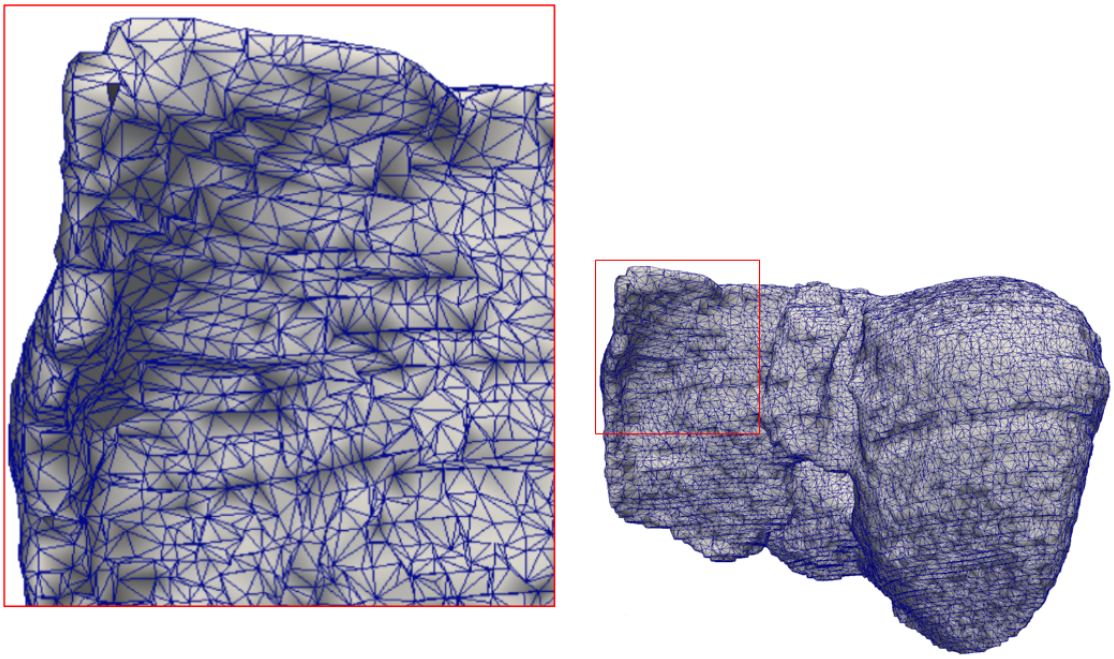
as a three-dimensional bioCAD mesh in order to obtain firstly a photo-realistic rendering of the natural object model, and secondly a valid bioCAD model for simulation purposes. Once the implicit surface is fitted to the discrete point-set, an unstructured volume mesh M_{Vol}, or an unstructured surface mesh M_S can be obtained for each region of interest in the given heterogeneous object Ω. The first step, in the mesh generation process, is to introduce the geometry description. Thus, to generate finite element meshes, either surface or volume, the length of element edges for each point in space must be specify as a function $h(x, y, z)$. After the definition of the mesh size, the standard edge length is computed in a normalized way as in Frey (Frey, 2000) and a range of acceptance is defined, elements outside the range will be refined or thickened.

The meshing process produces high-quality unstructured meshes for visualization purposes and for obtaining an accurate finite-element solution. For simulation purposes one of the most important quality metrics for both surface meshes and volume meshes is the minimal angle that avoids having flat triangles. This optimization is solve with the Delaunay triangulation (Bern et al., 1993). In the experiments both the surface mesh and the volume mesh were constrained to have triangles with minimum angle above 30 degrees as shown in Figure 3.15b and Figure 3.16.

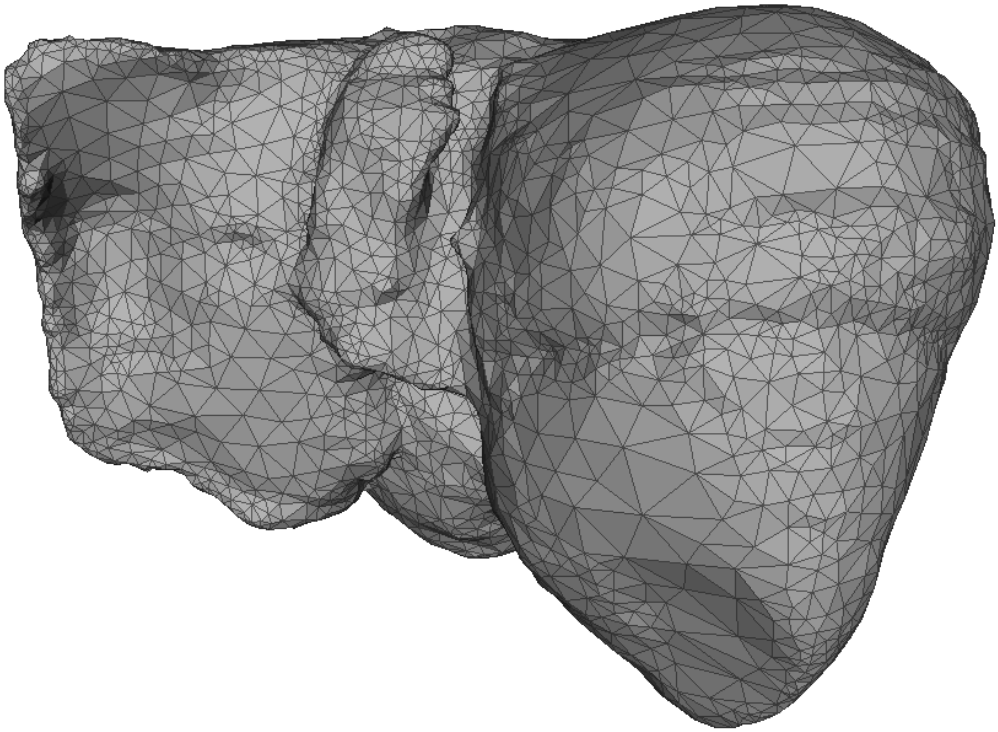
Surface meshing

Surface meshes are generated when applications are only interested in the boundary surface of a given region, for example, in computer graphics applications, or in boundary element methods. Using surface meshes often results in a more accurate solution due to the efficiency of shell elements, and in a preprocessing step for generating volume meshes. Therefore, engineers should consider using surface models of solid models whenever possible. Although, in old days surface models were visually awkward (since they have zero thickness and occupy zero volume in the space), nowadays this is eliminated by allowing a rendering of thickness, so that surface bodies can be visually the same as solid bodies.

For surfaces given as the zero level set of an implicit function $\phi(\mathrm{x})$ in $\mathbb{R}^3$, a mesh approximation of the isosurface can be generated by using the Marching Cubes algorithm (Lorensen and Cline, 1987) by chosen an appropriated isovalue Γ. In



(a) Marching Cubes mesh contains a huge number of triangles many of them badly shaped.



(b) Surface mesh fitting the implicit surface of the liver with a triangle size adapted to the curvature.

Figure 3.15: Surface mesh generation from implicit surfaces.

this case, the isovalue may be set to the average of the reconstructed indicator function χ at sample positions, weighted by the sample's area.

This method is really good for producing nice looking triangle meshes to be used in various applications, such as games, virtual museum and others. But, when the final surface mesh is produced by the Marching Cubes algorithm, it contains a huge number of triangles, many of them badly shaped (see Figure 3.15a). This makes a mesh simplification and a remeshing procedures be usually needed in order to produce appropriate three-dimensional bioCAD meshes.

There are two types of poor quality triangles: sharp and flat. The first refers to a triangle that has one angle close to zero and two close to 90 degrees. The second one has two angles close to zero and one close to 180 degrees. Although the first issue can be tackle using the edge collapsing technique, which will join the two nodes opposite to the close to zero angle, there is not a proper solution for the second issue.

Due to the above, instead of using the Marching Cubes algorithm, the algorithm proposed by Alliez (Jamin et al., 2015) is used. This algorithm computes a set of sample points on the implicit surface and extract an interpolating surface mesh from the three-dimensional triangulation of these sample points. The resulting mesh is adapted both to physical and geometric features of computational tasks, i.e, the surface mesh contains only quality triangles, with uniform sized triangles or triangles with size adapted the curvature, i.e., few triangles where the surface is flat, and many triangles where the surface has high curvatures (see Figure 3.15b).

Let $M_S = (V, F)$ be a closed triangular mesh, surrounding the interior of a region of interest Ω. $V = \{\mathbf{v}_i : 1 \leq i \leq n_V\}$ denotes the set of vertices, and $F = \{f_k : 1 \leq j \leq n_F\}$ the set of oriented triangular faces. The area of a particular face f_j with vertices $\{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2\} \in V$ equals the outer product of the two vectors $\mathbf{r}_1 = (\mathbf{v}_0, \mathbf{v}_1)$ and $\mathbf{r}_1 = (\mathbf{v}_0, \mathbf{v}_2)$:

$$Area(f_j) = 1/2(\mathbf{r}_1 \times \mathbf{r}_2) = 1/2(y_1z_2 - y_2z_1,\ z_1x_2 - x_1z_2,\ x_1y_2 - x_2y_1) \tag{3.10}$$

where $\mathbf{r}_1 = (x_1, y_1, z_1)$ and $\mathbf{r}_2 = (x_2, y_2, z_2)$. Here the symbol x denotes the cross products of two vectors.

The area of an arbitrary closed triangular mesh M_S is

$$\begin{aligned} Area(M_S) &= 1/2 \int_{\partial S} \mathbf{r} \times \mathbf{dl} \\ &= 1/2 \sum_{i=1}^{n-2} (v_1, v_{i+1}) \times (v_1, v_{i+2}) \end{aligned} \tag{3.11}$$

where ∂S is the boundary of region Ω and **dl** is the differential tangent vector of the boundary (Lien and Kajiya, 1984). Thus, the area of S equals the sum of the areas of all triangles.

Volume meshing

Given that some finite-element simulations are based on a volume representation of the region of interest, it is necessary to convert the triangular surface representation to a tetrahedral volume representation. Let $M_{Vol} = \{V, C\}$ be a volume mesh of a region of interest Ω_i. $V = \{v_l : 1 \leq l \leq n_V\}$ denotes the set of vertices, and $C = \{c_k : 1 \leq k \leq n_C\}$ the set of tetrahedrons. Each tetrahedron $c_k = \{i_1^k, i_2^k, i_3^k, i_4^k\}$ consists of a sequence of exactly four indexes indexes in the vertex list due to the *CAREM* method only produces tetrahedral cells.

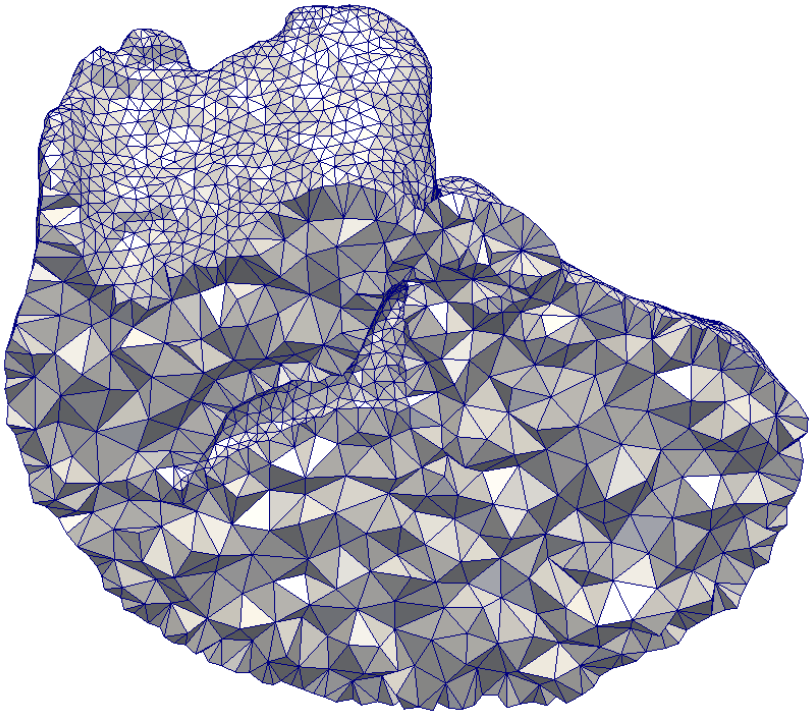
Suppose a tetrahedron C_k with vertices $\{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\} \in V$, and the vertex v_0 located at the origin. The coordinates of the vertices are:

$$\begin{cases} v_0 = & (0,0,0) \\ v_1 = & (x_1, y_1, z_1) \\ v_2 = & (x_2, y_2, z_2) \\ v_3 = & (x_3, y_3, z_3) \end{cases} \tag{3.12}$$

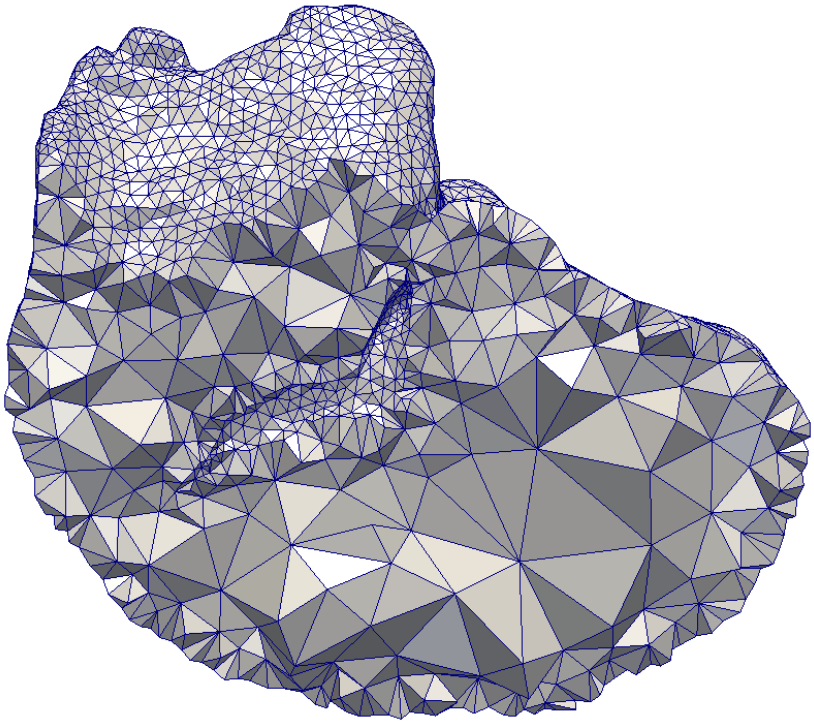
Let define a linear transformation **T** as

$$\mathbf{T} = \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{pmatrix} \tag{3.13}$$

This matrix relates the old coordinate system (x,y,z) with the new system (X,Y,Z) by



(a) Volume mesh generation with homogeneous cell size.



(b) Volume mesh with variable cell size.

Figure 3.16: Volume mesh generation from implicit surfaces.

$$\begin{matrix} x \\ y \\ z \end{matrix} = \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{3.14}$$

Applying this tranformation to tetrahedron $c_k = \{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$, an orthogonal unit tetrahedron $W_k = \{\mathbf{v}_0^{'}, \mathbf{v}_1^{'}, \mathbf{v}_2^{'}, \mathbf{v}_3^{'}\}$ is obtained.

then, the volume of C_k is:

$$Vol(C_k) = \int\int\int_{C_k} dv = \parallel T \parallel \int\int\int_{W_k} dv = \frac{\parallel T \parallel}{6} \tag{3.15}$$

where $\parallel T \parallel$ is the Jacobian equals to the absolute value of the determinant of matrix **T** (Lien and Kajiya, 1984).

For finite-element simulations the physical compatibility dictates a close correlation between size and shape of mesh cells (tetrahedrons) and the convergence of calculations. Thus, it may be necessary to describe appropriate element size not only on the boundary, but also inside regions.

The Jacobian ratio *(JR)* is one of the most used quality criterion in finite element meshes. The quality of an element *e* can be defined as

$$JR_e := \frac{\mid J \mid_{min}}{\mid J \mid_{max}} \tag{3.16}$$

and therefore, if any element presents a $JR_e \leq 0$, then not only is the element invalid, but the entire mesh is considered as not suitable for finite-element analysis.

In Figure 3.16a is shown a volume mesh with a homogeneous cell size discretization, whereas in Figure 3.16b a varying cell size of mesh elements is used to model the human liver with high accuracy. As can be seen from Figure 3.16 a lot of element nodes are eliminated when an adaptive cell size meshing method is used.

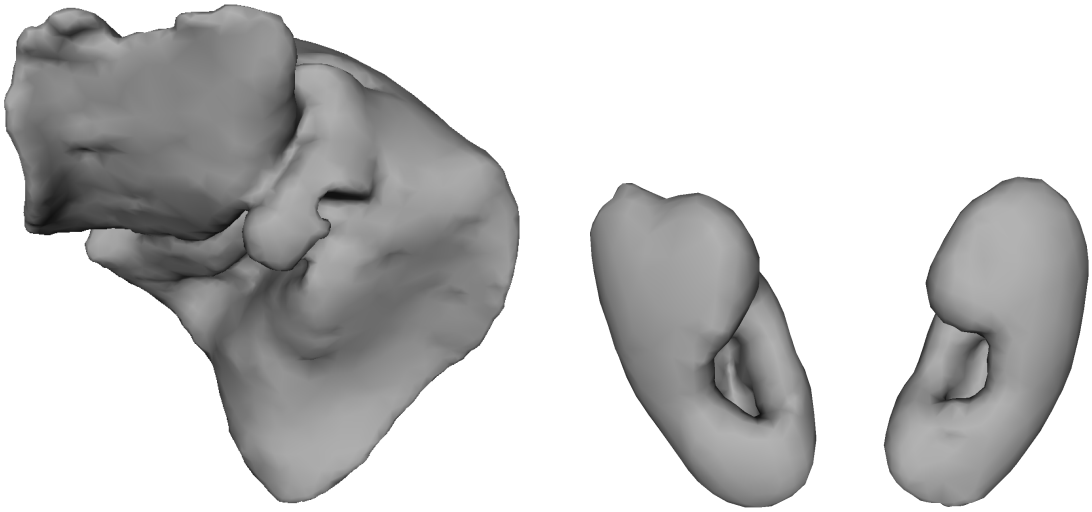
3.6 Examples

In this section, several examples using the *CAREM* method are shown to demonstrate the ability of this method to handle complex geometries, that anatomical regions and organs generally have. First, modeling three anatomical organs is presented in Subsection 3.6.1. Next, in Subsection 3.6.2 is showed the use of *CAREM* method for modeling more complicated structures such as the venous system.

3.6.1 modeling anatomical organs

Next three examples are related to the modeling of three anatomical organs whose voxel models are shown in Figure 3.10: a liver, a left kidney, and a right kidney.

First of all, in Figure 3.17 is shown the surface rendering of the implicit surfaces of the liver (see Figure 3.17a), a left kidney (see Figure 3.17b), and a right kidney (see Figure 3.17c).



(a) Back view of liver's surface.

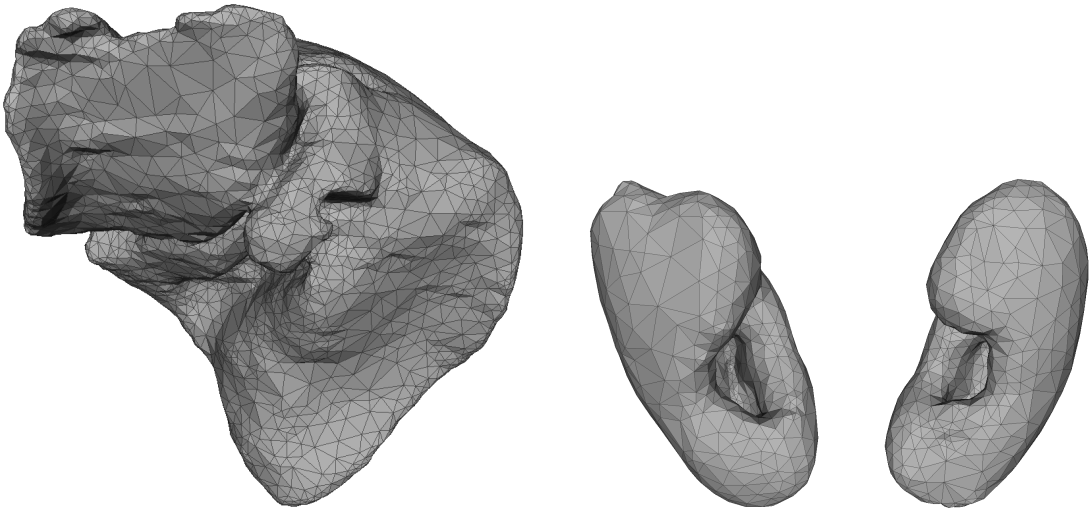
(b) Surface rendering of Left kidney.

(c) Surface rendering of right kidney.

Figure 3.17: Surface rendering of some 3D anatomical models obtained using the *CAREM* method.

It can be seen that the staircase effect appearing in voxel models has been re-

moved and instead of it, all organs are shown with smooth closed surfaces.



(a) Surface mesh of the liver.

(b) Surface mesh of the left kidney.

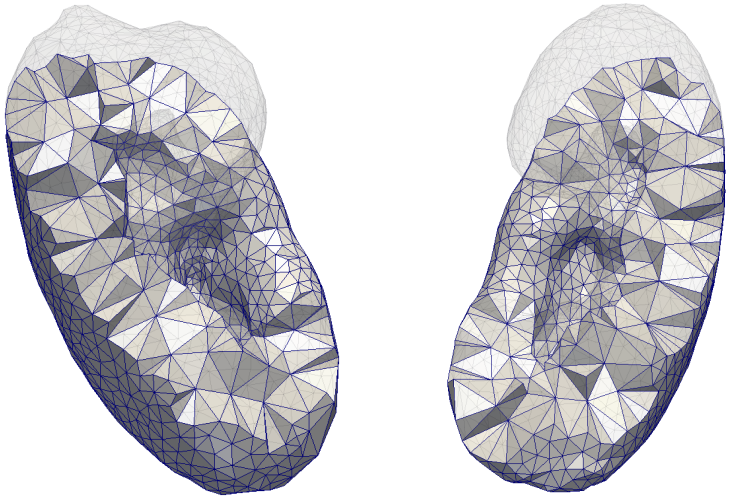
(c) Surface mesh of the right kidney.

Figure 3.18: BioCAD surface meshes fitting implicit surfaces of some anatomical organs.

An important feature of the *CAREM* method is the generation of bioCAD surface meshes. Figure 3.18 shows these surface meshes fitting the implicit surfaces of the liver (see Figure 3.18a), the left kidney (see Figure 3.18b), and the right kidney (see Figure 3.18c).

Another important feature of the *CAREM* method is the generation of bioCAD volume meshes (see Figure 3.19).

In Figure 3.19 a cross-section of volume meshes fitting implicit surfaces of two kidneys is shown: a left kidney (see Figure 3.19a), and a right kidney (see Figure 3.18c). The cell size of each tetrahedron is adapted to the organ size as it was shown in Figure 3.16.



(a) Visualization of a cross section of the left kidney's mesh.

(b) Vertical cut of the right kidney's mesh.

Figure 3.19: BioCAD volume meshes of some anatomical organs generated by the *CAREM* method.

3.6.2 Modeling anatomical structures

Above three examples have shown the ability of the *CAREM* method to generate three-dimensional bio-CAD models of any human organ. Next example is motivated to demonstrate the same ability in generating three-dimensional bioCAD models from more complicated structures, such as the venous system and other tubular structures. For this purpose, the venous system which interconnects the liver and the left and right kidneys was chosen in this case (see Figure 3.20). The voxel model which was segmented from the CT image is shown in Figure 3.20a whereas the surface rendering of the implicit surface of the venous system is depicted in Figure 3.20b.

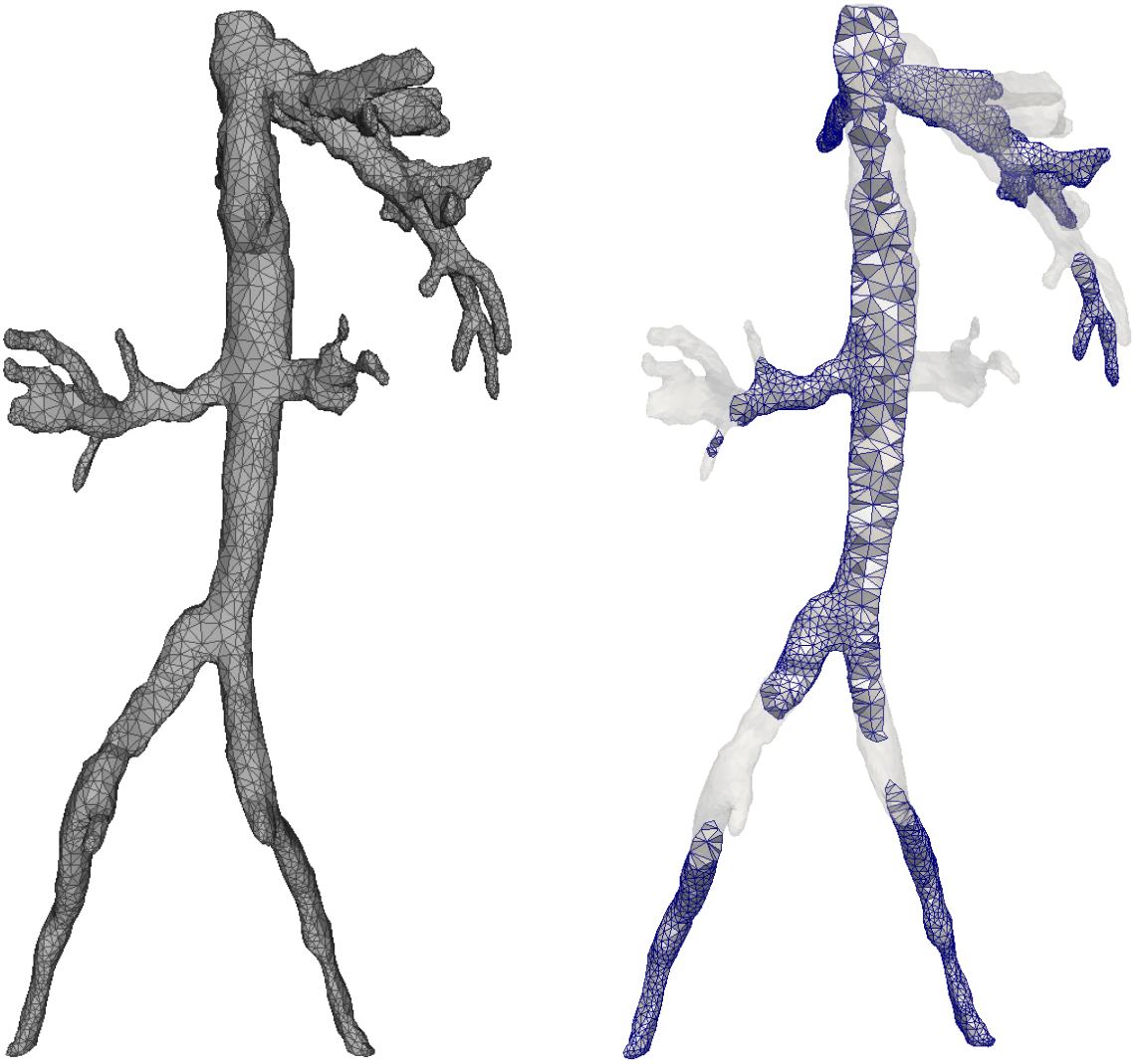
(a) A voxel model generated by the segmentation process.

(b) Surface rendering of the implicit surface representation.

Figure 3.20: Visualization of voxel model obtained from the venous system and surface mesh from the implicit representation.

Additionally, the bioCAD volume mesh of the venous system is shown in Figure 3.21. The physical compatibility dictates a close correlation between the size and shape of mesh cells (tetrahedrons) and the behavior of the finite element solution that is sought so a varying density of mesh elements can be used to model complex structures with high accuracy. In the *CAREM* method the appropriate element size distribution may be prescribed by the user.

Figure 3.21a depicts the surface view of the volume mesh whereas a vertical cross-section is shown in Figure 3.21b.



(a) BioCAD volume mesh of the venous system obtained from the implicit representation.

(b) A vertical cross-section of the bioCAD volume mesh.

Figure 3.21: BioCAD mesh generation from the venous system implicitly represented by the *CAREM* method.

3.7 Summary

This chapter was targeted for modeling specific regions composing natural heterogeneous objects, obtained using computer tomography scanners. To achieve this goal, the computer-aided reverse-engineering based modeling method (*CAREM*)

was proposed. The main idea is to construct specific three-dimensional computer models of different regions composing natural heterogeneous objects using a continuous implicit representation previous to the meshing process. In order to successfully build such three-dimensional models it is important to accurately identify and extract region of interest from medical images into voxel models through a process called image segmentation.

Given that the implicit surface reconstruction method takes as input a point cloud $\mathbf{P}$, the *CAREM* method is able to take a voxel model and input it into the Voxel-To-Point conversion process for extracting the boundary points and surface normals needed by the implicit surface reconstruction algorithm. So, the region's surface $\partial_\Omega = \mathbf{S}$ is represented by the zero level-set of an indicator function $\chi(\mathbf{P}) = 0$, which completely replaces the original voxel model during meshing operations.

The three-dimensional reconstruction process of the *CAREM* method finishes generating quality surface meshes approximating the boundary of specific regions. Moreover, volume meshes of such objects are also generated automatically by means of the implicit surface representation. This approach has a broad appeal to both theoreticians and engineers working in shape modeling and its application areas.

Some examples are given to demonstrate the ability of this method to produce quality models for anatomical organs and structures. Indeed, all examples presented in this chapter do not create a single degenerate triangle. The Poisson surface reconstruction method is computed using as solver the Eigen3, version 3.2.0. Surface meshing and volume meshing are computed using the Computational Geometry Algorithms Library (CGAL) library (Fabri et al., 1998). In Chapter 5 results of applying the *CAREM* method will be shown and discussed.

Chapter 4

Modeling natural heterogeneous objects

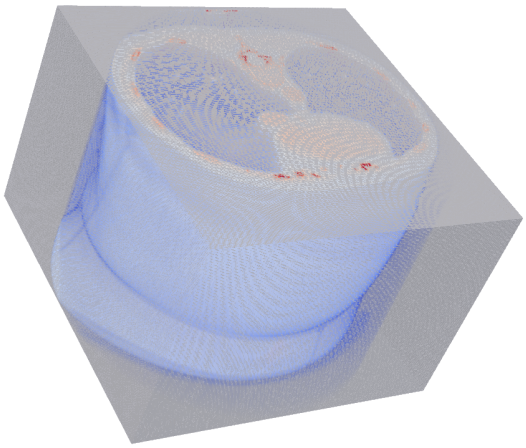
4.1 Introduction

MANY scientific modeling applications require the ability to model natural objects composed of multiple regions. In order to successfully build three-dimensional computer models from such objects, it is important to accurately identify and extract all regions comprising them. In Chapter 3, this is done by first isolating each region into a separate volume, extracting a point cloud from each volume, and then constructing smooth implicit representations from point clouds. However, these regions need to be considered as a whole, rather than loosely assembled parts. Additionally, the extraction of smooth multi-region meshes, that conform to the region boundaries comprising the multi-region natural heterogeneous object, is required.

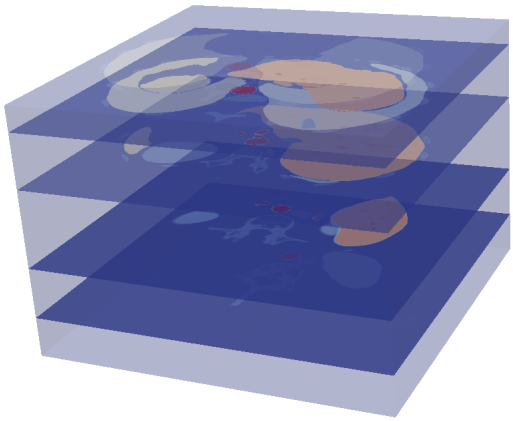
In this chapter, this issue is tackled through a region-aware method for heterogeneous object modeling named *RAM* method. In this method each region is called *feature-region* that makes reference to a closed volume of space represented by the geometry and the region material. The implicit surface representation is used to describe the geometry information of each feature-region, and the region material information is explicitly represented with a label $l_i : \ 1 \leq l_i \leq m$, where m is the total number of region materials, i.e., for each region material, there is one feature-region, containing this material (Feng et al., 2010) (Yuan et al.,

2012) (see Section 4.2). In Section 4.3 the Vector implicit function VIF structure is introduced.

Although different feature-regions composing the natural heterogeneous object can be modeled independently, they must eventually by stitched together. This stitching process requires a large amount of tedious manual intervention so an automate procedure is necessary. Based on both the vector implicit function F and the vector indicator function $\chi(F)$, a number of modeling operations are proposed in Section 4.4. The meshing process to obtain valid three-dimensional bioCAD models is explained in Section 4.5. In Section 4.6 the space and time analysis of the whole modeling process to construct three-dimensional computer models for natural heterogeneous objects is presented. Example of complex natural heterogeneous objects modeled with the proposed method are presented in Section 4.7. Finally, this chapter is summarized in Section 4.8.



(a) Abdominal CT image.



(b) 3D segmention into regions of interest from 4.1a.

Figure 4.1: Generation of feature-regions from a CT image.

4.2 RAM: A region-aware method for heterogeneous object modeling

In medical images multiple region intensities are typically present. For example, from a patient's abdominal CT image (see Figure 4.1a) about seven regions (organs and structures) have been segmented, each region with a specific intensity

(see Figure 4.1b). Thus, modeling natural heterogeneous objects begins with the segmentation of medical images into a set of voxel models, and generation of implicit functions that characterize the surface of each feature-region (see Figure 4.1).

4.2.1 Mathematical model

Let Ω be a natural heterogeneous object in a given geometric domain $D \in \mathbb{R}^3$ defined by a function $F(\mathbf{p})$ with $\mathbf{p} \in D$ (Wang and Wang, 2005). If there are a total of n feature-regions involved in Ω (see Figure 4.2), the value of $F(\mathbf{p})$ is an integral index of region materials between 0 and n (zero represents no material), which indicates the region material type in D at p.

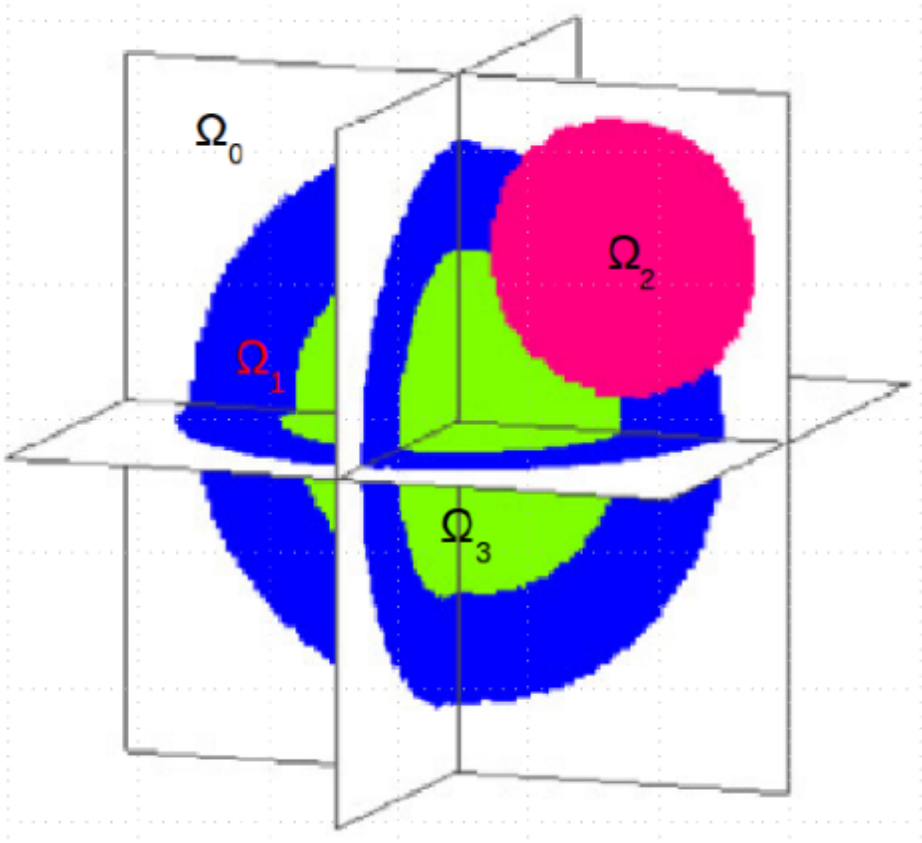


Figure 4.2: An artificial heterogeneous object composed of 3 regions. Adapted from (Wang, 2007)

Suppose that a region filled with the material i is denoted by Ω_i, then

$$\Omega = \bigcup_{i=1}^{n} \Omega_i \tag{4.1}$$

is the complement of Ω_0 in D. Thus, the natural heterogeneous object is partitioned by regions with unique material i. Additionally, as each feature-region Ω_i has a meaningful closed boundary surface $\Gamma_i = \delta\Omega_i$, the interface between two regions Ω_i and Ω_j can denoted by

$$\Omega_i \bigcap_{i \neq j} \Omega_j = \Gamma_i \bigcap \Gamma_j = \Gamma_{ij} = \Gamma_{ij} \tag{4.2}$$

meaning that the geometric domain D is a *partition*. The index $i,\ (i = 1, ..., n)$ is referred to as the feature-region id of Ω_i.

Now let

$$\chi_{\Omega_i}(\mathbf{p}) = \begin{cases} 1, \mathbf{p} \in \Omega_i \\ 0, \mathbf{p} \notin \Omega_i \end{cases} \tag{4.3}$$

be the characteristic function of the feature-region Ω_i. Suppose that $f_i(\mathbf{p})$ is an implicit function defined in $\mathbb{R}^3$ such that:

$$\begin{cases} f_i(\mathbf{p}) < 0 \quad if\ \mathbf{p} \in \Omega_i \\ f_i(\mathbf{p}) = 0 \quad if\ \mathbf{p} \in \partial\Omega_i \\ f_i(\mathbf{p}) > 0 \quad otherwise \end{cases} \tag{4.4}$$

Then, the result of multiplying function $\chi_{\Omega_i}(\mathbf{p})$ by function $f_i(\mathbf{p})$ has the following property:

$$\chi_{\Omega_i}(\mathbf{p}) f_i(\mathbf{p}) = \begin{cases} f_i(\mathbf{p}), & \mathbf{p} \in \Omega_i \\ 0, & \mathbf{p} \notin \Omega_i \end{cases} \tag{4.5}$$

For the function $f_i(\mathbf{p})$, once it is multiplied by the function$\chi_{\Omega_i}(\mathbf{p})$, the influence of function $f_i(\mathbf{p})$ outside the region Ω_i is diminished as the resulting function value will be zero for $\mathbf{p} \notin \Omega_i$, while the value of $f_i(\mathbf{p})$ for $\mathbf{p}$ inside the region Ω_i will not

be changed. Now, for a feature-region Ω_i represented implicitly by the equation $f_i(\mathbf{p}) = 0$, consider the following equation:

$$\chi_{\Omega_i}(\mathbf{p}) f_i(\mathbf{p}) + (1 - \chi_{\Omega_i}(\mathbf{p}))\tau = 0 \tag{4.6}$$

where $\tau > 0$. As can be seen from Equation (4.6), it will have exactly same solutions as the equation $f_i(\mathbf{p}) = 0$ when it is confined on the feature-region Ω_i. However, the part of the original shape defined by $f_i(\mathbf{p}) = 0$ outside the region Ω_i has been removed since when $\mathbf{p} \notin \Omega_i$

$$\chi_{\Omega_i}(\mathbf{p}) f_i(\mathbf{p}) + (1 - \chi_{\Omega_i}(\mathbf{p}))\tau \neq 0 \tag{4.7}$$

Then if the domain Ω is partitioned into *n* disjoint feature-regions $\{\Omega_1, \Omega_2, \ldots, \Omega_n\}$, that is $\mathbb{R}^3 = (\bigcup_{i=1}^{n} \Omega_i) \bigcup \Omega_0$, and the geometric shape over the feature-region Ω_i is represented implicitly as $f_i(\mathbf{p}) = 0$, then the overall geometric model of the natural heterogeneous object over $\mathbb{R}^3$ can be put in the form

$$F(\mathbf{p}) = \sum_{i=1}^{n} \chi_{\Omega_i}(\mathbf{p}) f_i(\mathbf{p}) \tag{4.8}$$

Definition 6. (Partition of a natural object): Let $\chi = \{\chi_{\Omega_i} : i = 1, 2, ..., n\}$ be a set of characteristics functions in a given domain $\Omega \in \mathbb{R}^3$. χ is said to be a partition of a natural heterogeneous object, if for each $\mathbf{p} \in \Omega$, then

$$\sum_{i=1}^{n} \chi_{\Omega_i}(\mathbf{p}) = 1 \tag{4.9}$$

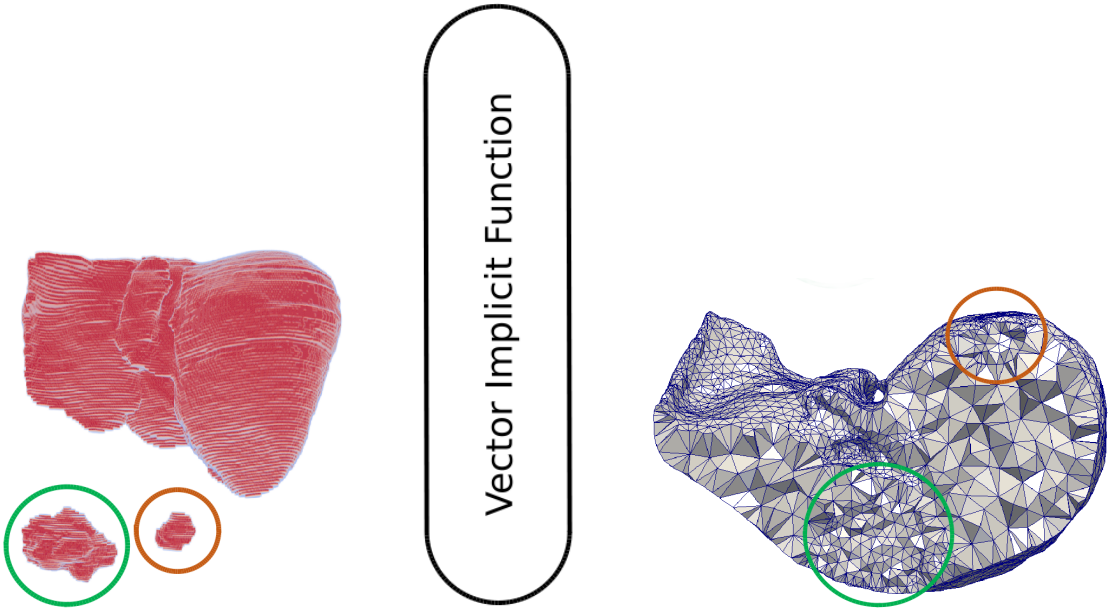
The successive steps are described in the following algorithm 4.1.

4.2.2 Algorithm

Each region composing the heterogeneous object is first segmented and reconstructed using an implicit based surface reconstruction approach (see Figure 4.3a). Then, a vector implicit function (VIF) is constructed based on the implicit surface fields (see Figure 4.3b). Finally, the three-dimensional model is constructed based on the VIF and additional user parameters (see Figure 4.3c).

Algorithm 4.1 The *RAM* method

Input: A set of regions Ω_i composing the natural heterogeneous object Ω.
Output: A closed three-dimensional model of the natural heterogeneous object Ω.
1: for all $\Omega_i \in \Omega$ do
$\mathbf{V}_i \leftarrow VoxelToPointCovertion(\Omega_i)$.
$\chi_i \leftarrow ImplicitSurfaceReconstruction(\mathbf{V}_i)$.
$VIF \leftarrow AddFunction(\chi_i)$
end for
2: Extract closed manifold surface model $SurfaceMesh(VIF)$.
3: Extract closed manifold volume model $VolumeMesh(VIF)$



(a) Region segmentation. (b) Vector implicit function. (c) Model generation.

Figure 4.3: An overview of the three-step algorithm for modeling natural heterogeneous objects.

4.3 Vector Implicit Function (VIF)

Following the idea of Vese and Chan (Vese and Chan, 2002) the concept of partition of a natural object (see Definition 6) can be best described with a vector

implicit function (VIF) F and a vector indicator function $\chi(F)$ corresponding to F:

$$\begin{aligned} F &= \{f_i \mid f_i : D \to \mathbb{R}\} \\ \chi(F) &= [\chi_i(F), \chi_2(F), ..., \chi_n(F)] \end{aligned} \tag{4.10}$$

Here, the vector indicator function $\chi(F)$ for each implicit function has a value of only 1 or 0. The derivative of the vector indicator function can be seen as the inward normal derivative at the boundary of the domain given by the positive half-line. Thus, feature-regions Ω_i $(i = 1, ..., n)$ of n different regions can be defined in the design domain D (including the background phase) by the vector indicator function $\chi(F)$ such that

$$\Omega_i = \{\mathbf{p} : \ \chi(F(\mathbf{p})) = \text{constant VIF}, \ \mathbf{p} \in D\} \tag{4.11}$$

Thus, the set of n implicit functions f_i is organized on a vector-valued continuous function such that:

$$F(\mathbf{p}) = [f_1(\mathbf{p}), f_2(\mathbf{p}), ..., f_n(\mathbf{p})], \ \mathbf{p} \in \Omega \tag{4.12}$$

where n is the number of regions. A label i is assigned to a point $\mathbf{p} \in \Omega$ if (and only if) $f_i(\mathbf{p}) > f_j(\mathbf{p}) \, \forall \, j \neq i$. For any single feature-region i, a continuous, inside-outside function can be constructed as $f_i(\mathbf{p}) = f_i(\mathbf{p}) - \min_{j \neq i}(f_j(\mathbf{p}))$, and the zero functions of various feature-regions will coincide at shared boundaries.

The VIF must be minimal, i.e. each dimension in F represents a unique feature-region and so, a component of VIF in a particular dimension would represent a distinct feature-region made of that corresponding material. The material label i is assigned to a point $\mathbf{p} \in \mathbb{R}^3$ if (and only if) $f_i(\mathbf{p}) > f_j(\mathbf{p}), \ \forall i \neq j$. Additionally, if $\mathbf{p}$ belongs to the interior of Ω_i, there does not exist $j(\neq i)$ such that $f_i(\mathbf{p}) = f_j(\mathbf{p})$, that emphasizes that the scalar component must have a unique maximum at the interior of a feature-region.

For the purpose of illustration, let us consider a case of three regions (n=3). It is needed three implicit functions as illustrated in Figure 4.2. In this case, the vector indicator function $\chi(F)$ has three possible vector values: $\chi(F) = \{[1, 0, 0], [0, 1, 0], [0, 0, 1]\}$, corresponding to each of the three distinct regions with the background phase (or void) indicated by [0,0,0].

The main purpose of the VIF is to provide a data structure for representing regions of a natural heterogeneous object and the incidence relation between them. VIF is an organized structure composed of a collection of vector elements, where each element has the information related to a feature-region. VIF maintains geometric variation dependencies among all feature-regions. The geometry of the whole natural heterogeneous object is dependent on (or determined by) its embedded feature-region geometries.

4.4 Modeling operations

Although a feature-region (component) in the heterogeneous object model could be considered as a primitive in a constructive solid geometry sense, the result from an operation does not yield new feature-region, instead of that it drives specific actions over the geometric model $F(\mathbf{p})$ of the natural heterogeneous object.

4.4.1 Boolean operations

The VIF structure may contain implicit surfaces that intersect each other. A combination of boolean operations such union, difference and subtraction are used to resolve such surface overlaps. Analytical definitions of the set-theoretical operations on functions describing objects have been introduced and studied by Rvachev for solving problems of mathematical physics in areas of complex shapes (Rvachev et al., 2001). In such definitions f_i and f_j are implicit functions of representing feature-regions Ω_i and Ω_j:

Union_Regions:

$$Union(\Omega_i, \Omega_j) = Union(f_i, f_j) = \Omega_i \bigcup \Omega_j = \max(f_i, f_j) \tag{4.13}$$

Intersection_Regions:

$$Intersection(\Omega_i, \Omega_j) = Intersection(f_i, f_j) = \Omega_i \bigcap \Omega_j = \min(f_i, f_j) \tag{4.14}$$

Difference_Regions:

$$Difference(\Omega_i, \Omega_j) = Subtraction(f_i, f_j) = \Omega_i \setminus \Omega_j = \min(f_i, -f_j) \tag{4.15}$$

4.4.2 Building operations

Building heterogeneous object models consist in incrementally add and remove feature-regions to/from the VIF. The order in which operations are executed determines the resulting heterogeneous object model configuration. The feature-region that is processed first will most likely lose the most of its geometry, and will have no influence on the other feature-regions. The feature-region that is processed last will not be influenced at all by other feature-regions.

Add_Feature_Region():

$$FeatureRegionAddition(\Omega_i, F) = Append(\Omega_i) \tag{4.16}$$

Delete_Feature_Region:

$$FeatureRegionRemoving(\Omega_i, F) = Delete(\Omega_i) \tag{4.17}$$

4.4.3 Querying operations

In many situations, being able to test whether or not a given point belongs to a given feature-region is a key operation. This is the point decision problem. Moreover, it is also interesting to determine the volume of a feature-region and its surface area. For example, one could ask in which region of the natural heterogeneous object a particular point of space is located, and what is the distance from that point to the surface of its enclosed region.

A point p inside a heterogeneous object model is generally represented as

$$\mathbf{p}(x, y, z, M) \tag{4.18}$$

where *(x,y,z)* denotes its location in the three-dimensional space and *M* represents its material composition at *(x,y,z)*. In some circumstances, a heterogeneous point may be also separately represented as $(\mathbf{p}, M(\mathbf{p}))$, where p denotes a geometric point and $M(\mathbf{p})$ refers to the material composition defined at that location.

Distance_Querying:

$$Distance(\mathbf{p}, \Omega_i) = \min(\mathbf{p}, \Omega_i) \tag{4.19}$$

Material_Querying:

$$Material(\mathbf{p},\Omega) = F[i] \tag{4.20}$$

Feature_Region_Geometry: Returns the geometry of the feature-region Ω_i.

$$FeatureRegionGeometry(\Omega_i) = \{\mathbf{p} : \ f_i \leq 0\} \tag{4.21}$$

Heterogeneous_Object_Geometry: Returns the geometry of all feature-regions that form the heterogeneous object model:

$$HeterogeneousObjectGeometry(\Omega) = F = \bigcup_{i=1}^{n} FeatureRegionGeometry(\Omega_i) \tag{4.22}$$

Volume_Feature_Region: Returns the volume of the feature-region Ω_i.

$$FeatureRegionVolume(\Omega_i) = \sum_{k=1}^{k=m} Volume(cell_{i,k}) \tag{4.23}$$

Volume_Heterogeneous_Object: Returns the volume of the whole heterogeneous object model

$$HeterogeneousObjectVolume(\Omega) = \sum_{i=1}^{n} FeatureRegionVolume(\Omega_i) \tag{4.24}$$

Feature_Region_Area: Returns the area of the feature-region Ω_i.

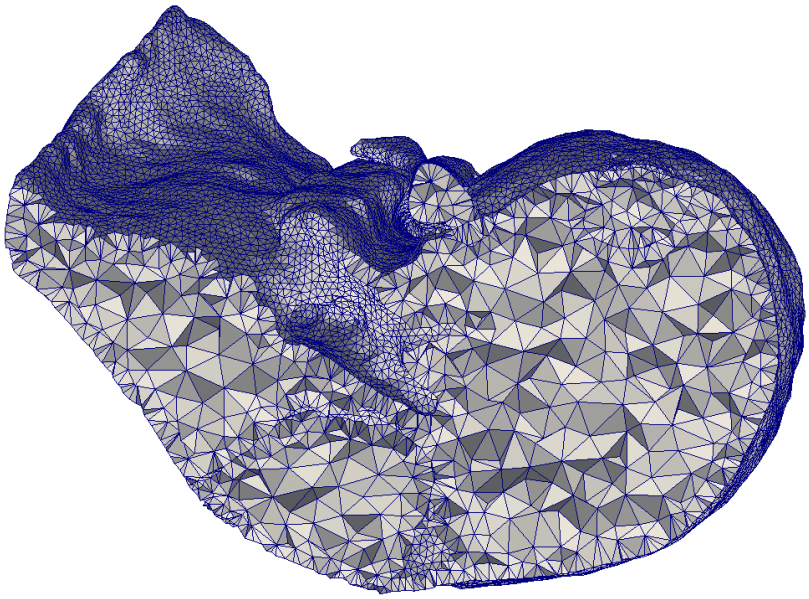
$$FeatureRegionArea(\Omega_i) = \sum_{k=1}^{k=m} Area(facet_{i,k}) \tag{4.25}$$

Once the implicit heterogeneous object model is available it could be rendered directly using a raytracing engine POV-Ray. However, In finite element analysis, multiple feature-regions correspond to various material regions of different physical phenomena. For this purpose, meshes with appropriate quality and conforming boundaries are required, i.e. interfaces separating different feature-regions, as well as boundary surfaces should be smooth and conform to tissue boundaries (see Figure 4.4).

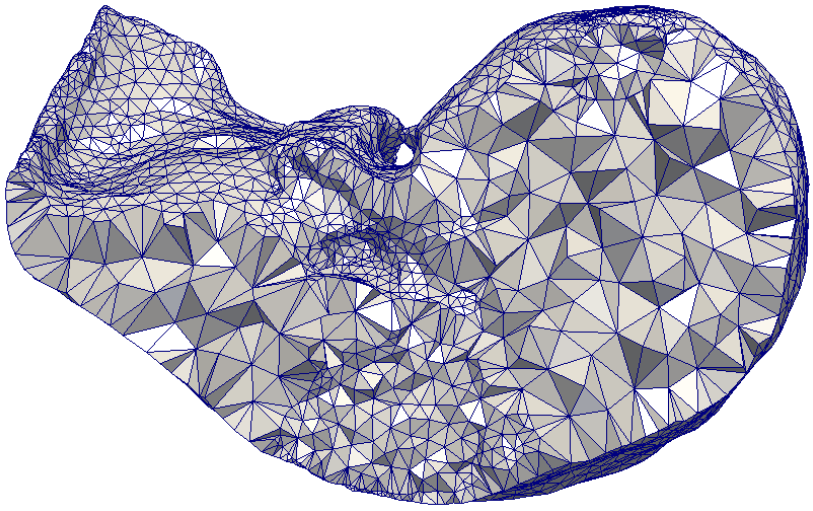
4.5 Meshing process

In the meshing process an appropriate element size needs to be fitted on the boundary and inside the heterogeneous object. One of the techniques commonly used for these purposes is based on so-called sources (Löhner, 1997). Here sources could be considered as feature-regions comprising the heterogeneous object. Typically, a small element size is desired close to the feature-region surface, and a large element size is more preferable far from it (see Figure 4.4a). In other cases, the element size for a feature-region Ω_i in the domain Ω is given as a function of the closest distance to feature-region Ω_i (see Figure 4.4b).

Suppose that an attribute $A(\Omega)$ is defined everywhere in the domain Ω and an arbitrary value $A_i = A(\Omega_i)$ is interpreted as the desired mesh element size at a point $\mathbf{p}_i$ of the feature-region Ω_i. Geometry of the feature-region is specified by the implicit function put in the VIF.



(a) A cross section of a bioCAD volume mesh with same cell size for all feature-regions.



(b) A cut of a bioCAD volume mesh with different cell size for each feature-region.

Figure 4.4: 3D bioCAD meshes from a volume model with embedded regions.

A value of the function χ_i describing the feature-region at a point $\mathbf{p}_i$ can be used as a measure of the closest distance from $\mathbf{p}_i$ to the feature-region Ω_i, $d(\mathbf{p}_i) = d(F(\mathbf{p}_i))$. The element size attribute A_i generated by the i^{th} feature-region is defined such that:

$$A_i(\mathbf{p}_i) = \begin{cases} h_{\min} & if\ F_i(\mathbf{p}_i) \leq 0 \\ \min(h_{\max}, (F_i(\mathbf{p}_i) * (k_i - 1) + h_{\min})/k_i) & if\ F_i(\mathbf{p}_i) > 0 \end{cases} \tag{4.26}$$

This formula provides the geometrical progression law of the element size increase. Here $h_{\min}$, and $h_{\max}$ are the minimal and maximal admissible sizes of the elements and k_i is the coefficient of the progression $k_i \geq l$, $F_i(X)$ is the functional description of the i^{th} feature-region (see Figure 4.4).

4.6 Space and time analysis

The idea that natural heterogeneous objects Ω can be decomposed into simpler regions Ω_i is fundamental to design a three-dimensional computer approach for modeling such objects. All the efficient divide-and-conquer algorithms divide the problems into subproblems, each of which is some fraction of the original problem, and then perform some additional work to compute the final answer. The *RAM* method begins with the generation of implicit functions that characterize the surface of each region composing the anatomical structure.

To compute the implicit surface reconstruction method the oriented point cloud $\mathbf{V}$ is organized on an full octree structure of cube dimensions $2^h x 2^h x 2^h$. The total number of cubes in the octree is 2^{3h}. the number of nodes in the octree is $2^{3h} + 2^{3h}/8 + 2^{3h}/8^2 + \ldots + 8^2 + 8 + 1 < (8/7)2^{3h}$. Thus the space used by the octree is $O(N)$ where $N = 2^{3h}$is the number of cubes in the octree. The height of the full octree is $h = \log_8(N)$. The time to build the octree is proportional to the size of the octree, i.e, $O(N)$.

The basic idea of the Poisson reconstruction process is to reconstruct the surface from $\mathbf{V}$, estimating the indicator function χ by ensuring its gradient as-close-as-possible to the normal field of $\mathbf{V}$ in terms of least-squares using $\| \nabla\chi - \mathbf{V} \|_2^2$. Thus, the solution of the Poisson equation is obtained comparing a finite set of n distinct points and taking a positive integer $k \leq \binom{n}{2}$, which reports the smallest distance between n pairs of points. Each implicit function is then added to the vector implicit function *VIF*.

Once the vector implicit function *VIF* is computed the *RAM* method outputs a

set of triangles in the Delaunay tetrahedrization of Ω, that forms a surface that approximates and is also topologically equivalent to Ω. Since computing the Delaunay tetrahedrization of Ω requires $O(N^2)$ time in the worst case, the surface reconstruction algorithm has also this quadratic worst time behavior.

It can be observed that the overall worst-case running time is $O(N^2 \log N)$ when k is $O(N^2)$.

4.7 Examples

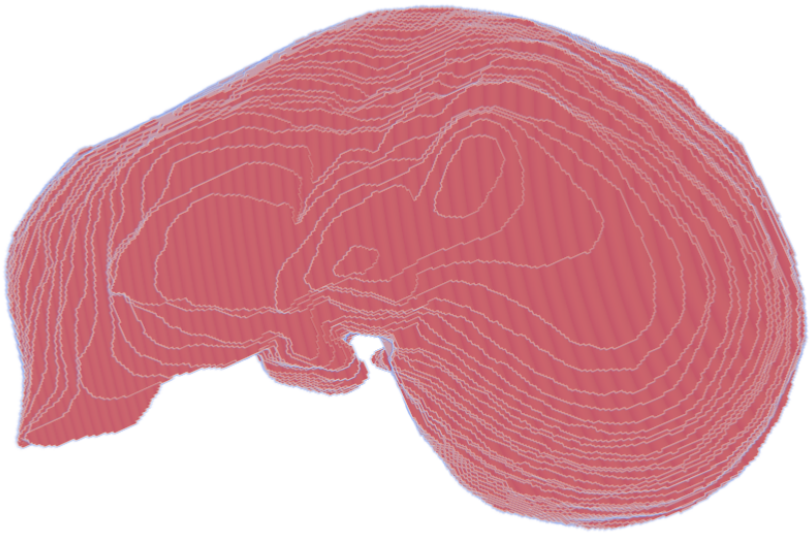
This section underlines the fact that natural heterogeneous object models can be made with different configurations based on both the mathematical model and the vector implicit function VIF structure. First, an example of modeling a natural heterogeneous object composed of a liver with 2 liver-tumors is considered (see Subsection 4.7.1). Next, it is showed an example of using the *RAM* method for modeling a multi-material heterogeneous object composed of a liver, a left kidney, a right kidney, and a venous system (see Subsection 4.7.2). Additionally, a third example with a more complex configuration is shown where the previous two examples are combined in a general heterogeneous object model (see Subsection 4.7.3).

4.7.1 Modeling NHOs with embedded feature-regions

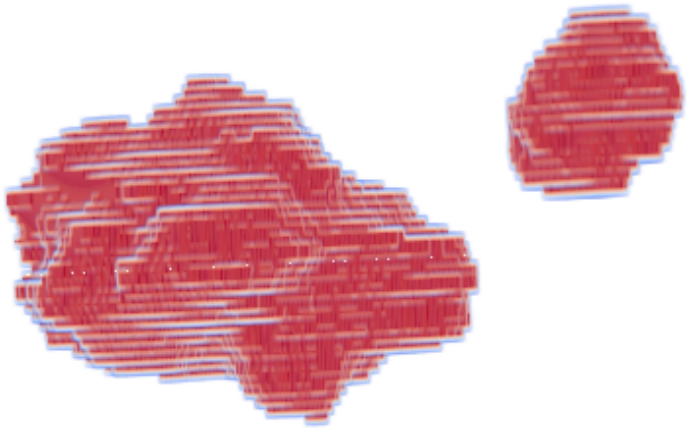
This example is related to modeling of natural heterogeneous objects with embedded regions. Figure 4.5 depicts three voxel models generated from an abdominal CT image: a liver (see Figure 4.5a), and two tumors inside the liver (see Figure 4.5b). From each voxel model a feature-region has been constructed using the *CAREM* method described in Chapter 3. Then these feature-regions are added to the VIF structure from which three-dimensional computer models of heterogeneous objects are generated. In Figure 4.4 the domain geometry is described by the VIF $F(\mathbf{p}) = [f_1(\mathbf{p}), f_2(\mathbf{p}), f_3(\mathbf{p})]$, where $f_1(\mathbf{p})$ describes the liver, $f_2(\mathbf{p})$ and $f_3(\mathbf{p})$ define liver-tumors (1 and 2).

In this case, the overall surface geometry of the heterogeneous object model is the geometry of the feature-region that embeds other feature-regions. Thus, the

bioCAD surface mesh of the heterogeneous object model is the same bioCAD surface mesh of the liver that is shown in Figure 3.15b. But the case is different when bioCAD volume models need to be generated. As there are various feature-regions involved in the natural object the cell size is an important feature of the bioCAD volume mesh.



(a) A top view of a liver voxel model.



(b) Voxel models of two liver-tumors.

Figure 4.5: Voxel models for volume modeling with embedded regions.

Figure 4.4 shows two bioCAD volume models generated with different cell size strategy. In Figure 4.4a the cell size is the same for all feature-regions forming the heterogeneous object model, while in Figure 4.4b the cell size is different for

each feature-region, however this size does not change inside the feature-region.

4.7.2 Modeling NHOs with multiple regions

In this example, a multi-material heterogeneous object composed of four feature-regions is modeled. In Figure 3.10 and Figure 3.20 voxel models of a liver, a left kidney, a right kidney, and a venous system are shown generated from another abdominal CT image.

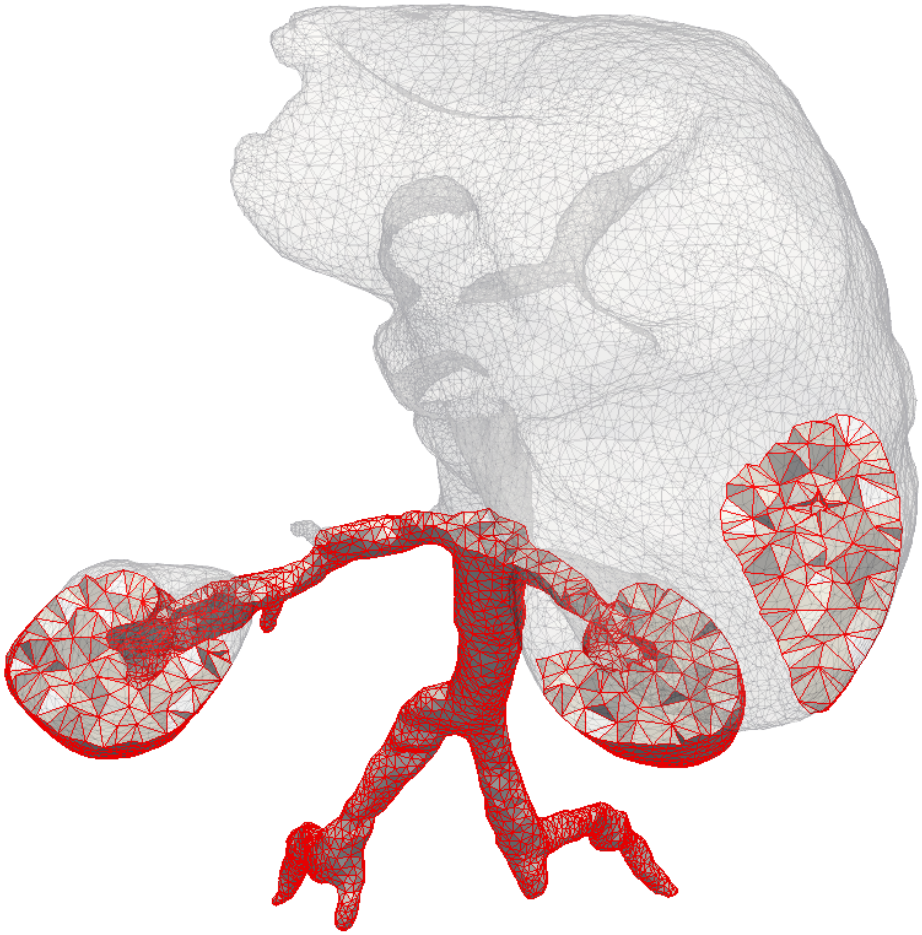


Figure 4.6: A cross-section of the tetrahedral mesh for all four feature-regions

From each voxel model a feature-region is built using the *CAREM* method, and put into a VIF structure. Then a bioCAD surface mesh (see Figure 4.7) or a bioCAD volume mesh (see Figure 4.6) from the heterogeneous object model is generated.

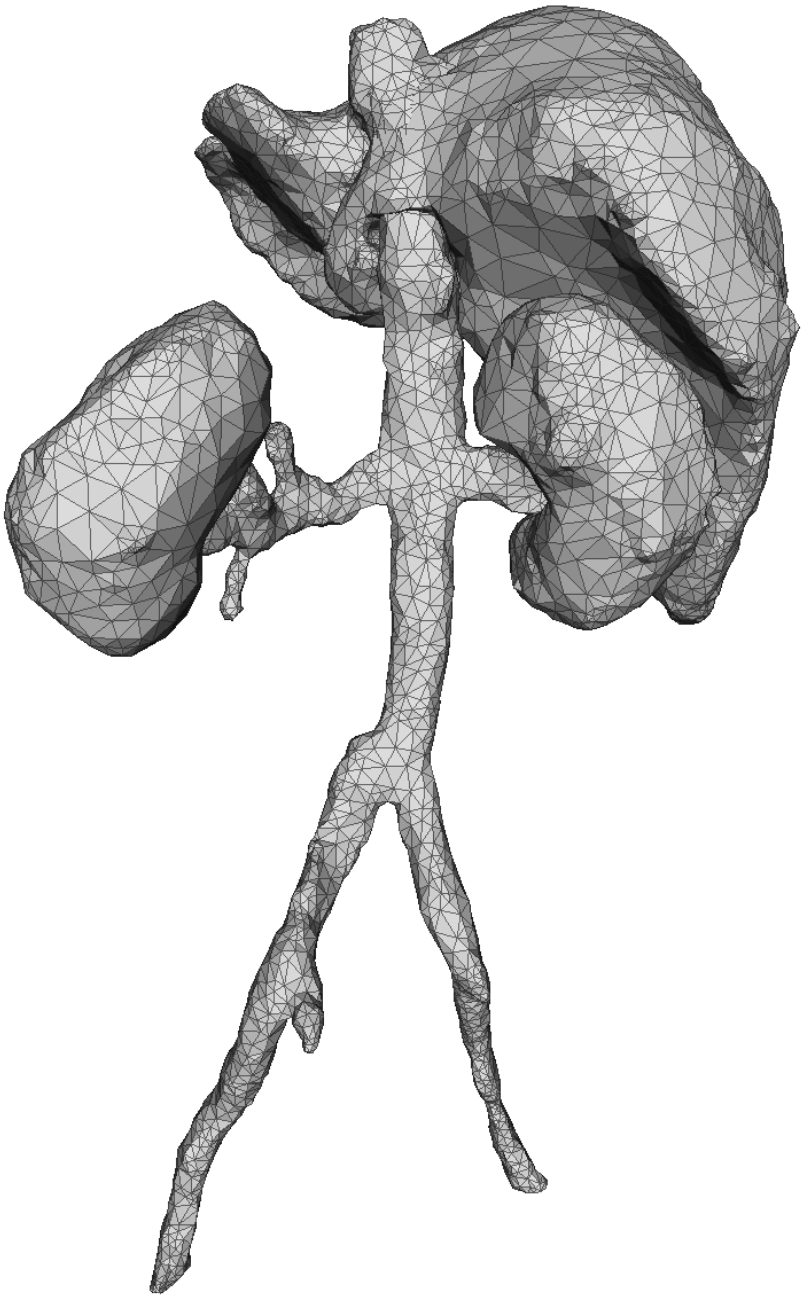


Figure 4.7: Surface modeling of a multi-material natural heterogeneous object.

Unlike previous cases, the surface mesh generated is not exactly the union of the feature-region surfaces, but the surface mesh of the heterogeneous object model generated from the VIF structure. A wireframe model of this mesh is shown in Figure 4.8

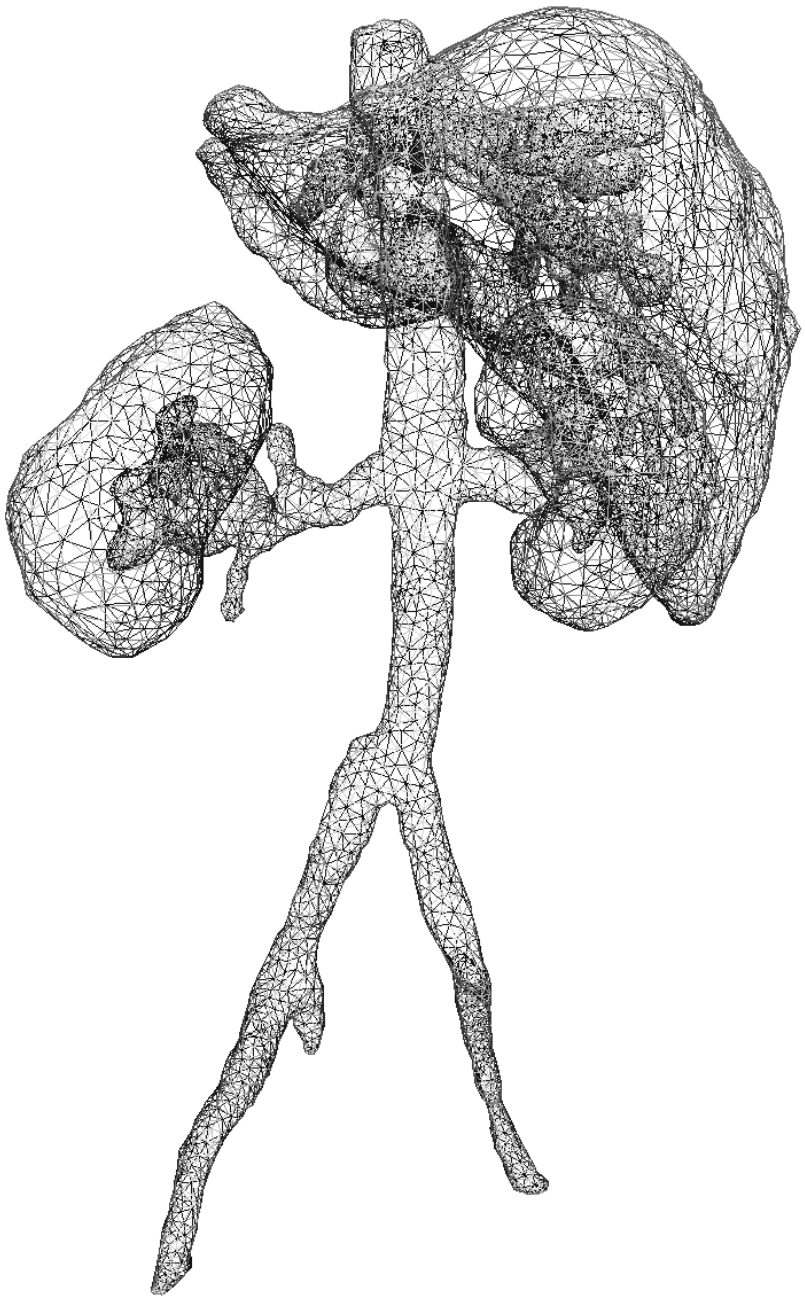


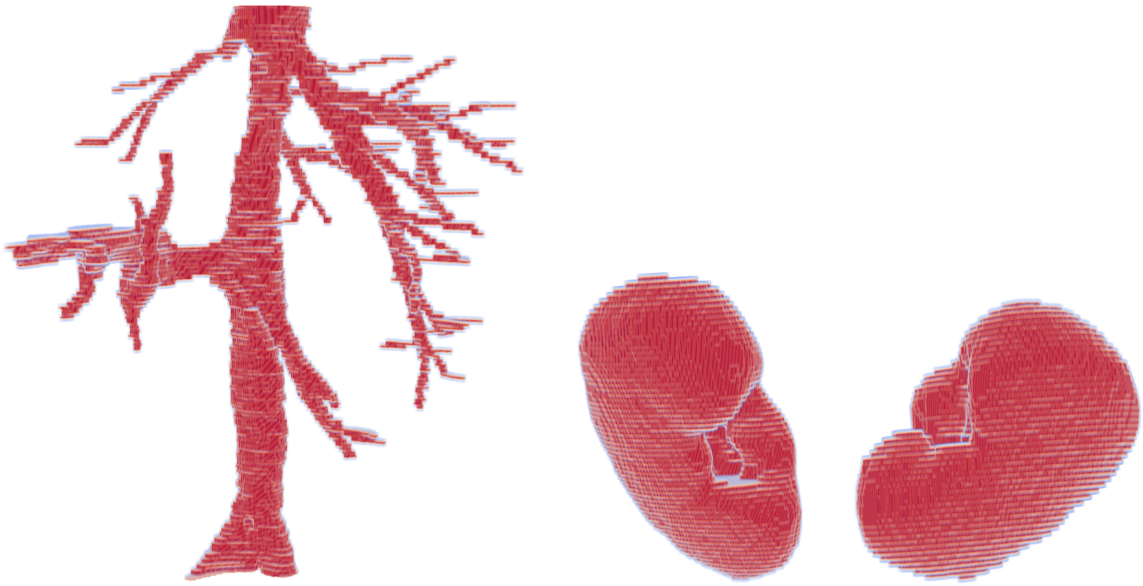
Figure 4.8: Wireframe visualization of a multi-material natural heterogeneous object

The bioCAD volume mesh from the heterogeneous object model is obtained applying the same strategy for the cell size as in the previous case, i.e., set up the cell size equal for all feature-regions or establish a different cell size for each feature-region. Figure 4.6 shows one cross-section of the bioCAD volume mesh

where each feature-region was configured with a different cell (tetrahedron) size, in this case, kidneys=6, venous system = 4, and liver=8.

4.7.3 Modeling general heterogeneous objects

In this example, a more general heterogeneous object model is obtained combining two previous examples, i.e., a multi-region heterogeneous object with embedded regions is modeled.

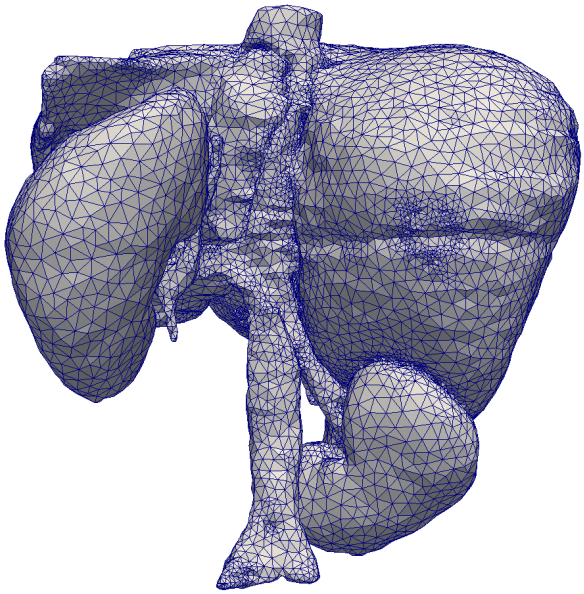


(a) Venous system. (b) Left kidney. (c) Right kidney.

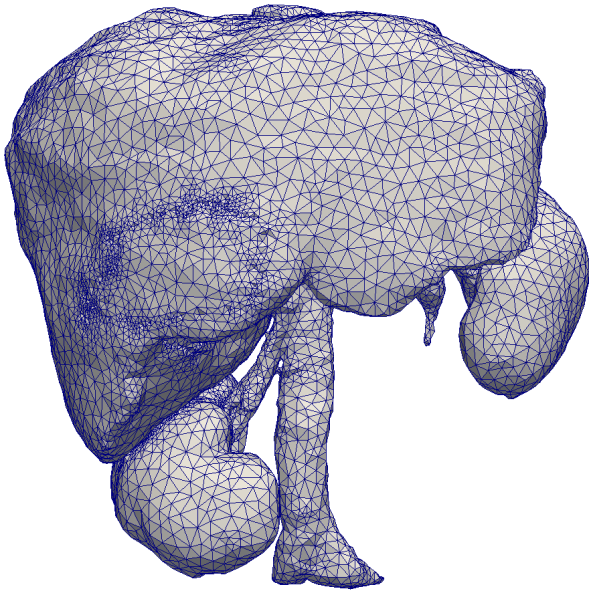
Figure 4.9: Voxel models for volume modeling of a natural heterogeneous object.

Figure 4.9 depicts voxel models that form the multi-region part of the heterogeneous object model. They are a venous system (see Figure 4.9a), a left kidney (see Figure 4.9b), and a right kidney (see Figure 4.9c). The embedded part is taken from Figure 4.5.

As the cell size of liver-tumors is smaller than the cell size of the liver, the geometry of heterogeneous object model reflects this aspect (see Figure 4.10). For instance, Figure 4.10a and Figure 4.10b show the anterior view and posterior view respectively of the heterogeneous object model of a liver with two liver-tumors embedded in it.



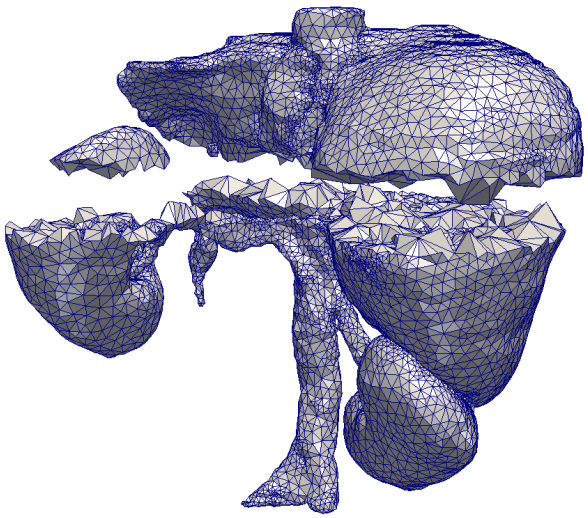
(a) An anterior view of the bioCAD volume mesh.



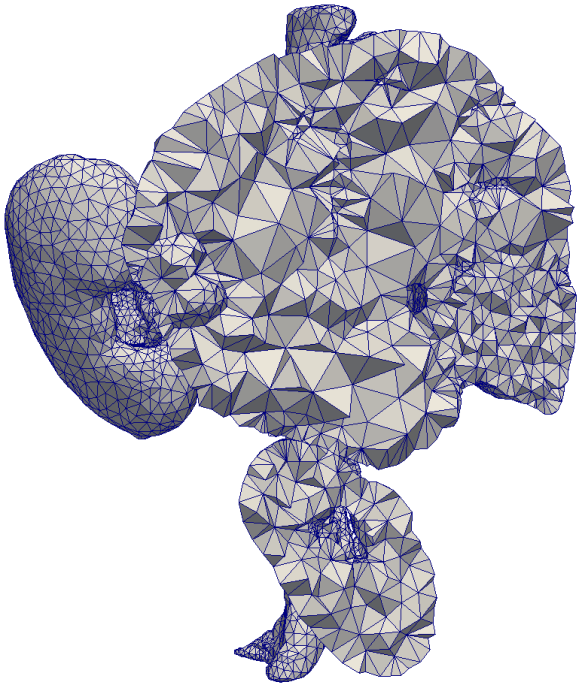
(b) A posterior view.

Figure 4.10: Rendering of 3D bioCAD meshes with embedded regions.

Although the cell size is different for each feature-region (see Figure 4.11), the size of the surface triangles (faces) is the same for all the heterogeneous object model (see Figure 4.10). This is consistent with the definition that all feature-regions form an heterogeneous object model (see Definition 6).



(a) A cross section of the multi-region volume mesh with embedded regions.



(b) A cut of a bioCAD volume mesh with different cell size for each feature-region.

Figure 4.11: Sections of a multi-region 3D bioCAD mesh with embedded regions.

Figure 4.11a shows a cross section of the multi-region bioCAD mesh with embedded regions where tetrahedrons of the heterogeneous object model are depicted. Figure 4.11b shows a vertical cut of the bioCAD volume mesh where a different

cell size is evident.

4.8 Summary

In this chapter, a region-aware heterogeneous object modeling method named *RAM* method was proposed to achieve such a goal. Due to the flexible and versatile divide-and-conquer approach, the *RAM* method can handle multi-material heterogeneous objects as well as heterogeneous objects with embedded feature-regions. Additionally, modeling operations are defined and implemented, with which bioCAD models can be constructed from simple feature-regions.

There are two key properties of the proposed *RAM* method making it suitable for modeling natural heterogeneous object. First, the *independence* of feature-regions. This means that feature-regions are self-contained; they do not depend on any external geometric information. Second, the ability to organize feature-regions into a meaningful implicit vector function VIF structure. Such structure is beneficial because it allows complexity to be encapsulated, which is important for both usability and computational efficiency.

Thus, natural heterogeneous objects are modeled with a set of implicitly represented regions assuming to have homogeneous material distributions. Note that there are no special restrictions on decompositions, as long as they altogether form an assembly (partition) of the final object model. Although each feature-region is independently reconstructed and put in an independent VIF element, the heterogeneous object model is considered as a whole, rather than loosely *assembled* parts. Furthermore, diverse decompositions can be possible for the same natural object depending of different possible segmentations.

Some examples are given to demonstrate the ability of this method to model natural heterogeneous objects composed of regions within different configurations. This is expected to make *RAM* method transcend the traditional boundary geometric modeling and be of benefit to data visualization and finite element-based analysis. In Chapter 5 results of applying the *RAM* method will be shown and discussed.

Part III

Results and conclusions

Chapter 5

Results and discussion

5.1 Introduction

In this chapter, the results of evaluating three-dimensional models produced by applying both the *CAREM* method and the *RAM* method described in Part II are presented. These results were obtained through a modeling framework implemented in C++ on a PC with a 3.30GHz Pentium-i7 CPU and 16Giga RAM. However, there is not ground truth from real patient data to compare the results with.

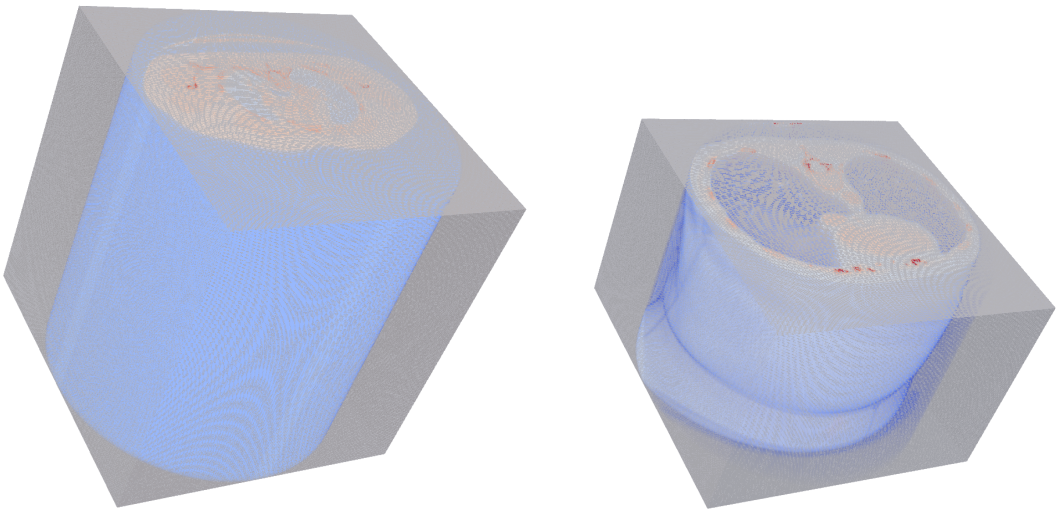
In the absence of ground truth, some measure of the accuracy can be obtained in two ways. The first is to compare results to a gold standard, which has been generated and reviewed by a panel of experts. The drawback of this approach is that the gold standard could vary depending on the composition of the panel of experts and the modeling method used to generate the gold standard. Moreover, creating a panel of experts in the field of three-dimensional modeling of natural heterogeneous objects is a difficult task due to the small number of those experts and the limited time they have. A second option is to conduct an evaluation, in which the accuracy of three-dimensional models depends solely on quality of input voxel models, i.e. of the segmented images, and of the function used to fit the implicit surface to extracted point sets.

Two datasets obtained from IRCAD (IRCAD, 2015) are considered for applying the *CAREM* method presented in Chapter 3 and the *RAM* method presented in

Chapter 4. From the dataset-1 four feature-regions were segmented, and from dataset-2 were segmented six feature-regions (see Figure 5.1).

Dataset-1 was acquired using a CT scan in Dicom format with dimensions: 512 x 512 x 219; spacing: x=0.96, y=0.96, z (space between slices) =2.39; Data ranges are between -2048 and 2890. All voxel models (segmented organs and structures) have same dimensions as the original patient Dicom image, i.e., 57409536 voxels that are equivalent to 110 MB (see Figure 5.1a).

Dataset-2 was also acquired using a CT scan in Dicom format with dimensions: 512 x 512 x 129; but smaller spacing: x=0.57, y=0.57, z (space between slices) =1.59. Data ranges are between -1024 and 1023. All voxel models (segmented organs and structures) have same dimensions as the original patient Dicom image, i.e., 33816576 voxels that are equivalent to 66 MB (see Figure 5.1b).



(a) Dataset-1 in Dicom format containing 21 organs and structures.

(b) Dataset-2 containing 19 organs and structures including 7 liver tumors.

Figure 5.1: Datasets obtained from IRCAD IRCAD (2015) for testing both the *CAREM* method and the *RAM* method.

The chapter starts presenting the results from modeling anatomical organs and structures using the *CAREM* method (see Section 5.2). Then, the results of applying the *RAM* method to model natural heterogeneous objects are presented (see Section 5.3). Finally, some factors that affect the modeling approach are discussed in Section 5.4.

5.2 Modeling regions using the *CAREM* method

The *CAREM* method can produce three-dimensional bioCAD models both surface and volume of any region or object composed of only one region. Next, different processes involved in the *CAREM* method are brief summarized. As a first step, regions of interest from each dataset are identified and voxel models of different organs and structures are extracted (see Section 3.3). In a second step, voxel models are used as input to the Voxel-To-Point conversion process to obtain from each voxel model a point cloud along with a set of normal vectors (see Section 3.4). Finally, each point cloud is loaded into the three-dimensional reconstruction process (see Section 3.5).

In this section, quality of different three-dimensional bioCAD models obtained when using the *CAREM* method is evaluated. Accuracy (see Subsection 5.2.1), efficiency (see Subsection 5.2.2), and volume preservation (see Subsection 5.2.3) are the three criteria chosen for evaluating the performance of the *CAREM* method. Here, each of these terms are defined, and how they can be measured is described. Finally, a sensitivity analysis is presented in Subsection 5.2.4.

5.2.1 Accuracy

Accuracy is the degree to which the resulting three-dimensional bioCAD model matches the truth. Dice's coefficient and Hausdorff distance are two measures used to evaluated the accuracy of three-dimensional bioCAD models.

Dice's coefficient

Dice's coefficient C_{Dice} is used to measure the volume of the region shared by two models as a percentage of the total volume occupied by both models (Dice, 1945). It is calculated as twice the ratio of the number of voxels contained in the intersection of two models to the total number of voxels contained in each solid model.

$$C_{Dice} = \frac{2 \mid A \cap B \mid}{\mid A \mid + \mid B \mid} \tag{5.1}$$

Table 5.1 summarizes the results from computing the Dice's coefficient using the MeshValmet tool (https://www.nitrc.org/projects/meshvalmet), between the

known Voxel model (A) and the bioCAD surface mesh (B) obtained from the implicit representation.

Table 5.1: Dice's coefficient between Voxel models obtained from dataset-1 (left) and dataset-2 (right), and bioCAD surface meshes obtained applying the *CAREM* method.

Dataset-1	Dice coefficient
liver	0.9918
left kidney	0.9833
right kidney	0.9892
venous system	0.9578

Dataset-2	Dice coefficient
liver	0.9946
left kidney	0.9884
right kidney	0.9876
venous system	0.9425
liver-tumor1	0.9870
liver-tumor2	0.9519

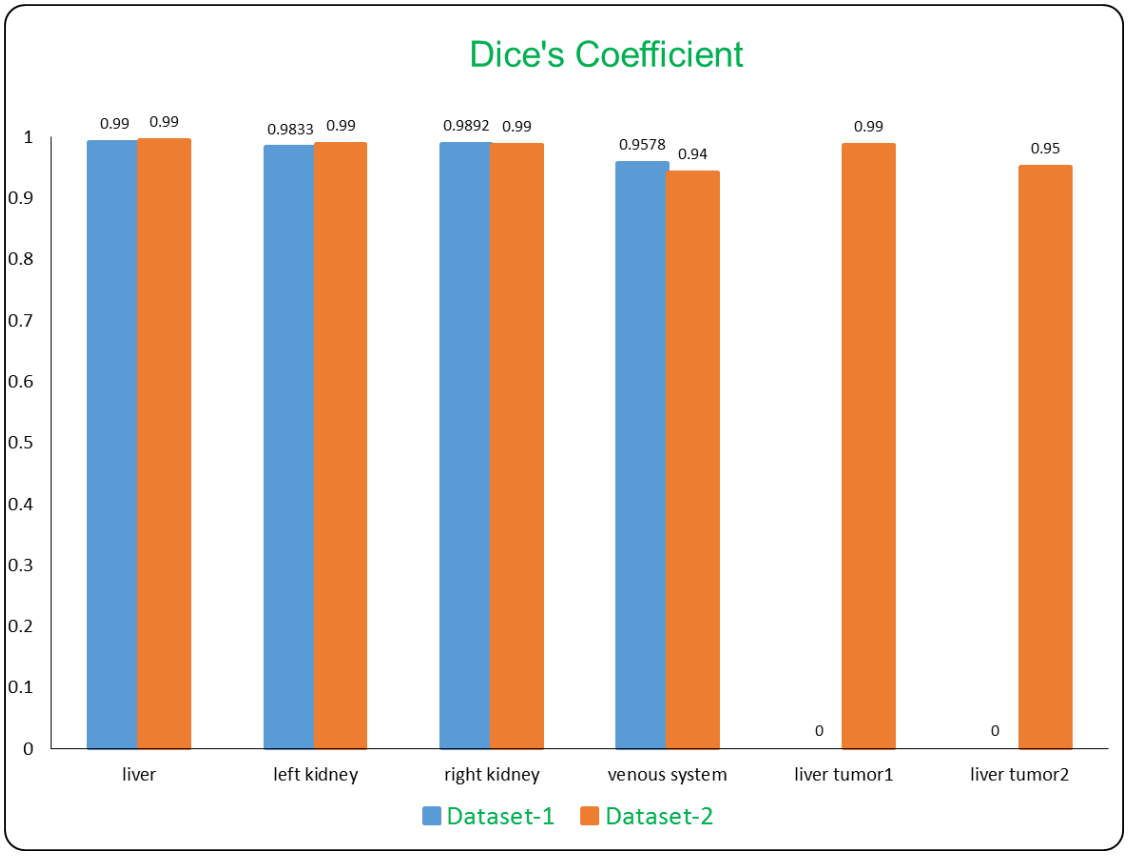


Figure 5.2: Dice's coefficient of 3D bioCAD models from Table 5.1.

It can be seen from Table 5.1 that results are more than satisfactory by noting

that values of the Dice's coefficient are above 0.94 for all cases. For instance, the Dice's coefficient for the dataset-1 is above 0.95 (see Table 5.1) left, while for the dataset-2 is above 0.94 (see Table 5.1) right.

Lowest values in both datasets corresponds to the venous system (see Figure 5.2), which is a structure with a very complicated geometry due to the number of branches it has, and the limited number of points each of these branches is represented by.

Indeed, Figure 5.2 shows how the more irregular the feature-region is, the smaller the Dice coefficient is. This coefficient is also affected when the feature-region is represented by a small number of points; see the case of the liver-tumor2 in Figure 5.2, yellow line. Although this feature-region has a rounded shape the number of points that represents it is small (see Table 5.4).

Hausdorff distance.

The Hausdorff distance $D_{Hausdorff}$ measures the difference between two object models in terms of the distances between point samples on the surfaces of both models. Given two finite point sets $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$, for each point on surface A, the distance to surface B is defined as the Euclidean distance to the nearest point on surface B.

$$D_{Hausdorff}(A, B) = \max(d(A, B), d(B, A)) \tag{5.2}$$

where

$$d(A, B) = \max_{a \in A,\ b \in B} \min \parallel a - b \parallel \tag{5.3}$$

and $\| \cdot \|$ is some underlying norm on the points a and b, for instance, the Euclidean norm. Table 5.2 summarizes the results of computing the Hausdorff distance $D_{Hausdorff}$ between the known Voxel model (A) and the bio-CAD surface model (B), using the standard tool METRO (Cignoni et al., 1998).

Table 5.2: Hausdorff distance between Voxel models obtained from dataset-1 (top) and dataset-2 (button), and 3D bioCAD models obtained applying the *CAREM* method.

Dataset-1	Forward distance $d(A, B)$	Backward distance $d(B, A)$	Hausdorff distance
liver	2.8193	1.5545	2.8193
left kidney	3.5515	1.3670	3.5515
right kidney	2.9204	1.3581	2.9504
venous system	4.3665	4.3143	4.3665

Dataset-2	Forward distance $d(A, B)$	Backward distance $d(B, A)$	Hausdorff distance
liver	4.3309	0.9878	4.3309
left kidney	1.5016	0.9024	1.5016
right kidney	1.3785	0.8574	1.3785
venous system	2.0587	8.6911	8.6911
liver-tumor1	1.2365	0.9236	1.2365
liver-tumor2	1.0504	0.7186	1.0504

It is important to note that this distance is in general not symmetrical, i.e. $d(A, B) \neq d(B, A)$ as shown in Table 5.2. The distance $d(A, B)$ is referred to as forward distance, and distance $d(B, A)$ as backward distance.

According to Table 5.2-top forward distance is taken as the Hausdorff distance for bioCAD models obtained from dataset-1 because it is the biggest distance between both forward distance and backward distance. This is not the case for three-dimensional bioCAD models obtained from dataset-2 (see Table 5.2-bottom), where for the venous system the backward distance has been chosen.

Figure 5.3 depicts the tendency of the Hausdorff distance. It shows how the more irregular the feature-region is, the bigger the Hausdorff distance is. For instance, the Hausdorff distance of the venous system model is almost the double than the Hausdorff distance of the second bigger bioCAD model- the liver, as it is shown in Figure 5.3 yellow line. This seems to be related to the evident lost of surface area and volume percentage that present each feature-region once the bioCAD model is generated (see Table 5.4 and Table 5.6) respectively.

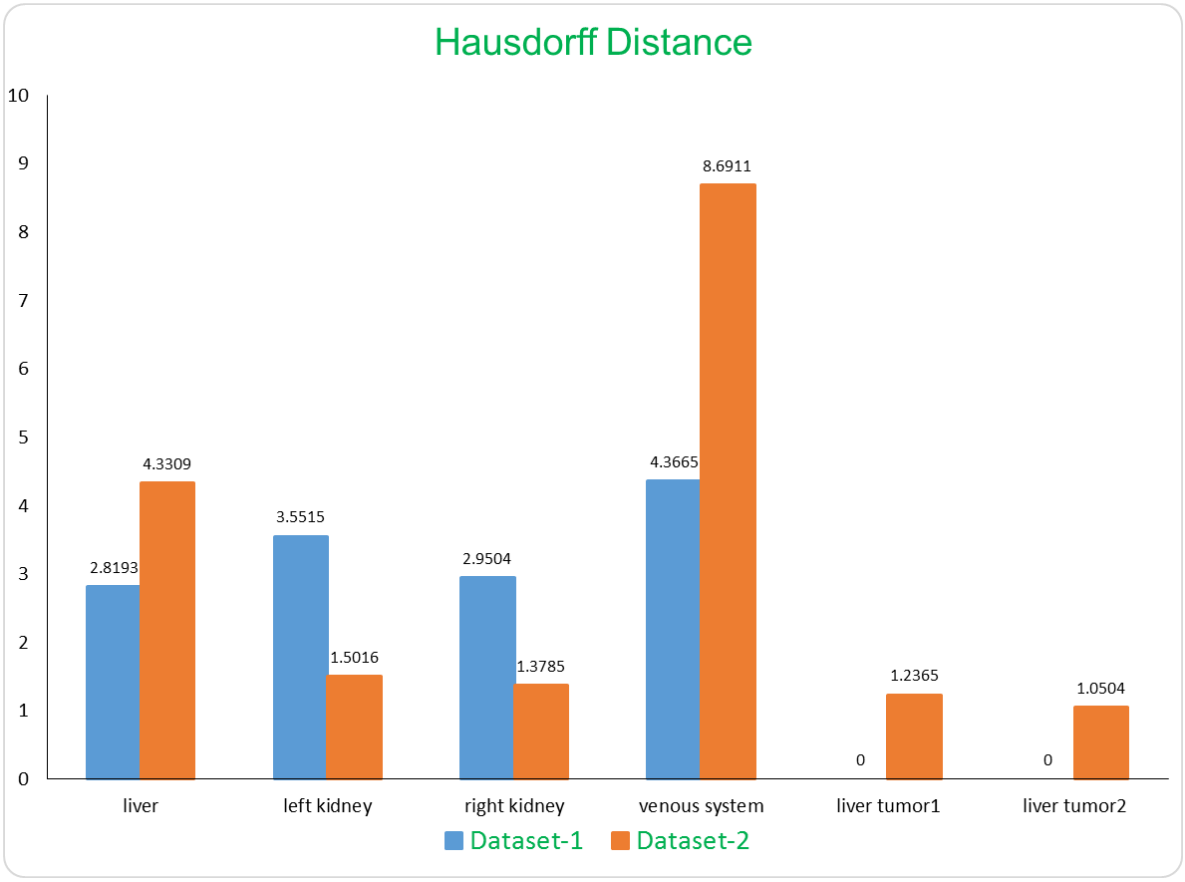


Figure 5.3: Hausdorff distance of 3D bioCAD models from Table Table 5.2.

5.2.2 Efficiency

In this subsection are summarized the temporal and spatial efficiency of the *CAREM* method. In Table 5.3 are listed some time statistics for all processes involved in the *CAREM* modeling method: point cloud generation, implicit surface reconstruction, surface meshing, and volume meshing for modeling feature-regions obtained from both datasets.

Table 5.3: Time statistics (in seconds) for all processes involved in the *CAREM* method for modeling feature-regions from dataset-1 (top) and dataset-2 (button).

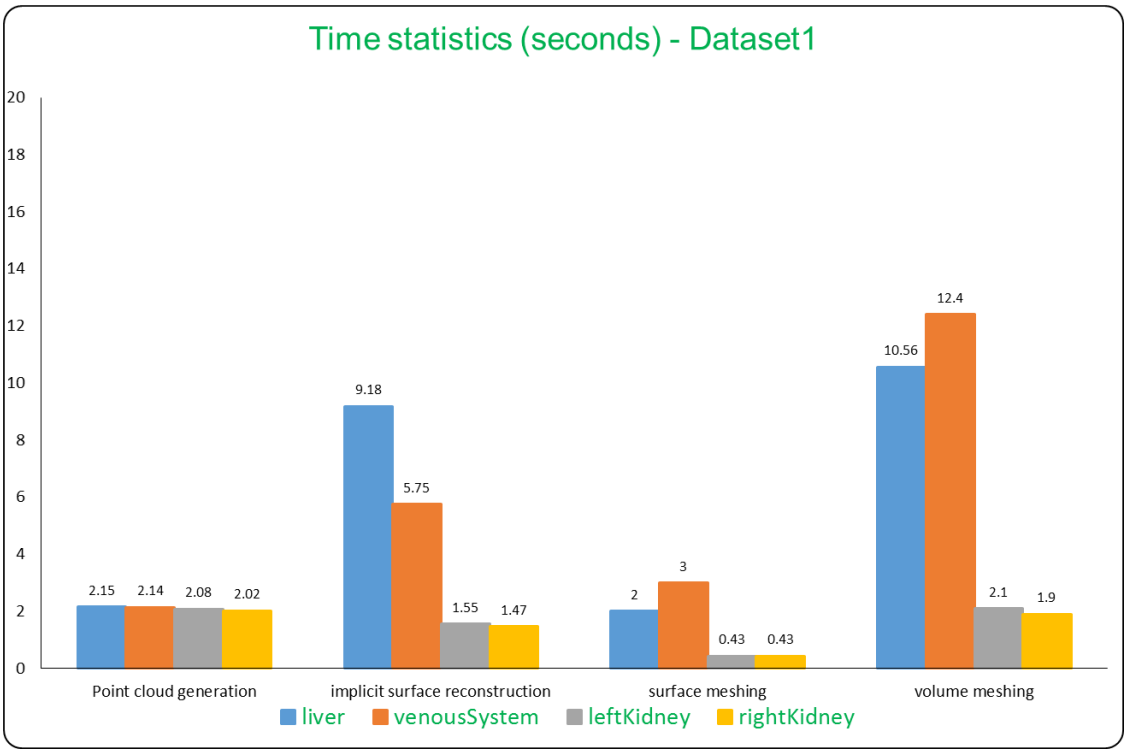
Dataset-1	Point cloud generation	Implicit surface reconstruction	Surface meshing	Volume meshing
liver	2.15	9.18	2.0	10.56
venous system	2.14	5.75	3.0	12.40
left kidney	2.08	1.55	0.43	2.10
right kidney	2.12	1.47	0.43	1.9

Dataset-2	Point cloud generation	Implicit surface reconstruction	Surface meshing	Volume meshing
liver	1.32	18.9	4.52	7.92
venous system	1.28	7.7	5.55	9.43
left kidney	1.26	3.67	0.83	1.73
right kidney	1.26	2.90	0.76	1.45
liver-tumor1	1.27	2.05	0.81	1.63
liver-tumor2	1.23	0.42	0.12	0.35

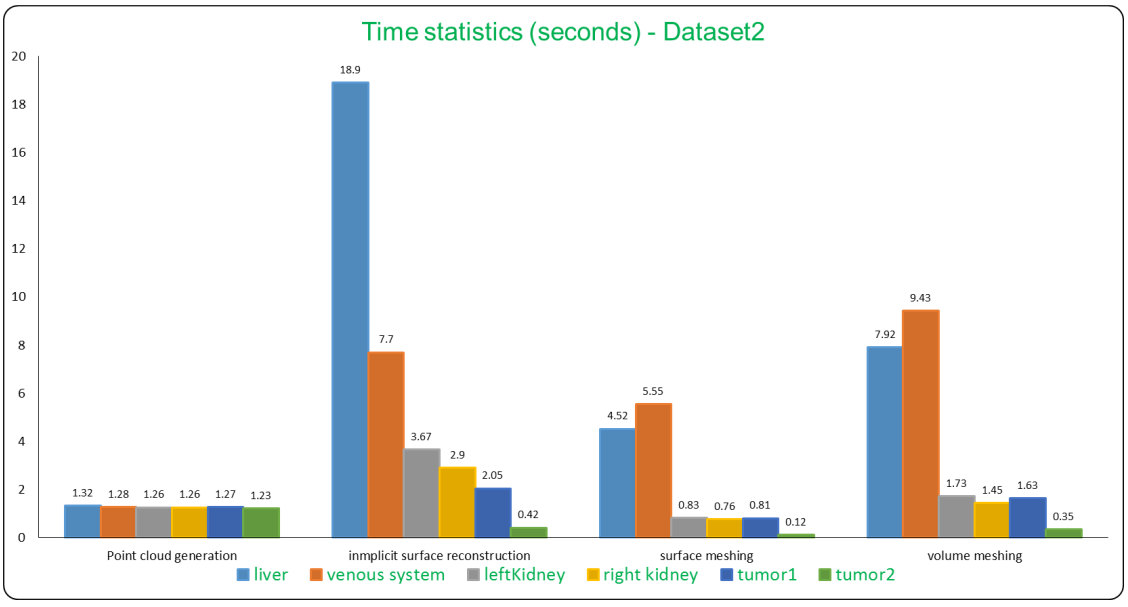
Figure 5.4a shows the time required for the same process by all feature-regions belonging to dataset-1 and dataset-2. It can be observed that the time associated with the point cloud generation from voxel models belonging to the same dataset is almost the same: 2.10 and 1.27 respectively. This is because the size of voxel models from each feature-region are the same no matter the size of the feature-region as long as they come from the same dataset. Due to the fact that Voxel-To-Point conversion process has to check all voxels in the voxel model to obtain the cloud point, it takes the same time to generate point clouds no matter which voxel model is inputed to.

The time for the implicit surface reconstruction process, for dataset-2, is the greatest value with the exception of the venous system (see Figure 5.4b), whereas for dataset-1 the volume meshing is the greatest value for all cases (see Figure 5.4a).

It can be inferred that the time for the implicit surface reconstruction process depends more of the number of points generated in the Voxel-To-Point procedure than of geometry complexities of feature-regions.



(a) Time required for all processes involved in the *CAREM* method by each feature-region belonging to dataset-1.



(b) Time required for all processes involved in the *CAREM* method by each feature-region belonging to dataset-2.

Figure 5.4: Time statistics for all processes involved in modeling feature-regions applying the *CAREM* method.

The second more time consuming process is the volume meshing which in some times is higher than the implicit surface reconstruction. This is the case of the venous system of the dataset-1 whose volume meshing takes almost the double than the implicit surface reconstruction process (see Figure 5.4a). But when the number of points increases only the time needed for the implicit surface reconstruction process increases (see Figure 5.4b).

Additionally, in Table 5.4 summarizes some statistics related to the surface meshing using the MC-based method which makes the triangulation directly from the voxel models, and the bioCAD surface meshing process using the *CAREM* method. This includes surface area in mm2, number of points, and number of faces (triangles) generated by both the MC-based method and the *CAREM* method.

Table 5.4: Some statistics for surface mesh generation from dataset-1 (top) and dataset-2 (button) using the *CAREM* method.

Dataset-1	Points MC	Triangles (faces)	Surface area	Points *CAREM*	Triangles (faces)	Surface area
liver	114026	228052	133799	3687	7374	116039
left kidney	20684	41364	25014	820	1636	21718.4
right kidney	20224	40444	25088	863	1722	21918.5
venous system	63390	126666	83938	5822	11644	73419.5

Dataset-2	Points MC	Triangles (faces)	Surface area	Points *CAREM*	Triangles (faces)	Surface area
liver	221402	442788	99179.6	7681	15358	85677.6
left kidney	46822	93640	21492	1599	3194	18697.5
right kidney	38582	77160	17121	1466	2928	14620
venous system	93714	186992	41712	10864	21740	35523
liver-tumor1	29510	59016	12919	1531	3058	11048.4
liver-tumor2	5228	10452	2325	291	578	1998.7

An important feature of the *CAREM* method is the space reduction after the surface meshing process is performed mantaining a good value of the Dice's coefficient (see Subsection 5.2.4). In all experiments the space reduction is above 88 per cent in comparison with the space needed by the MC triangulation (see Table 5.5).

Table 5.5: Space reduction and surface area keeping by the *CAREM* method.

Dataset-1	Space reduction	Surface area keeping
liver	96.77	86.73
left kidney	96.04	86.82
right kidney	95.73	87.37
venous system	90.82	87.47
Dataset-2	Space reduction	Surface area keeping
liver	96.53	86.39
left kidney	96.58	87.00
right kidney	96.20	85.39
venous system	88.41	85.16
liver-tumor1	94.81	85.52
liver-tumor2	94.46	85.97

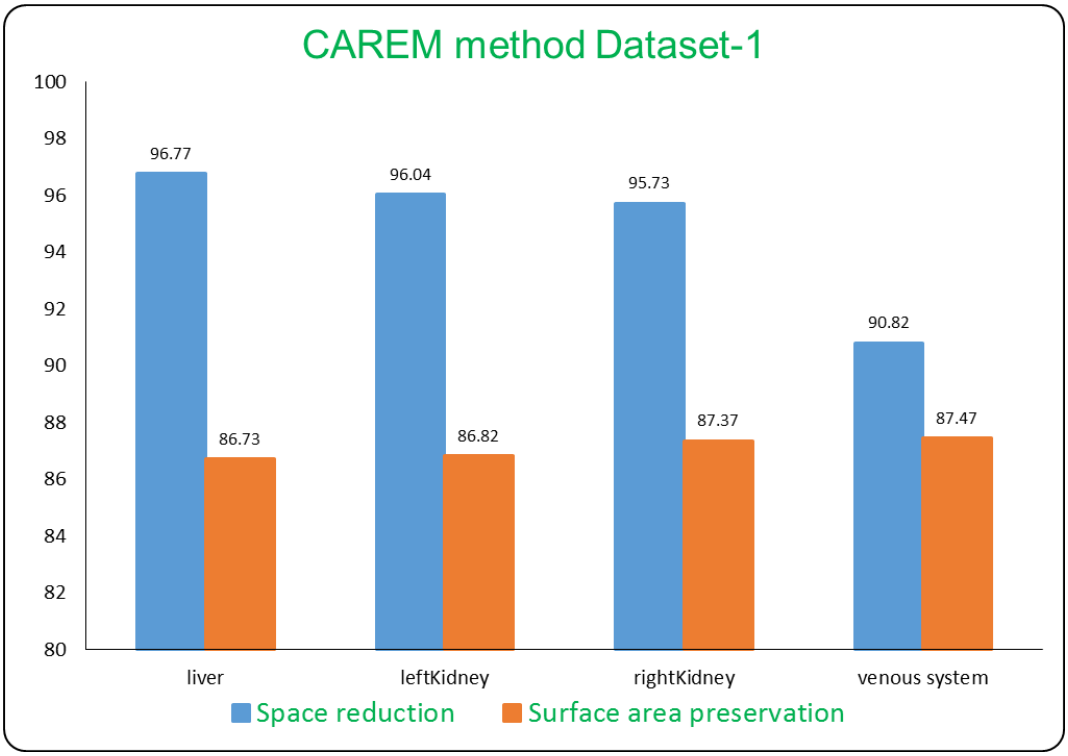


Figure 5.5: Memory space reduction by the *CAREM* method after the surface meshing process.

Although the space reduction is very similar for both datasets (see Figure 5.5), it

should be noted that the area involved in each feature-region is much higher in dataset-1 than in dataset-2, but the number of initial points is the opposite, i.e., point clouds of dataset-1 have fewest points (almost 50%) than those of dataset-2 (see Table 5.4).

Another feature of the *CAREM* method is the surface area that is kept during the meshing process in comparison with the surface obtained from the MC method (see Figure 5.6). This preservation, even in the worst case is over the 85 percent (see Table 5.4).

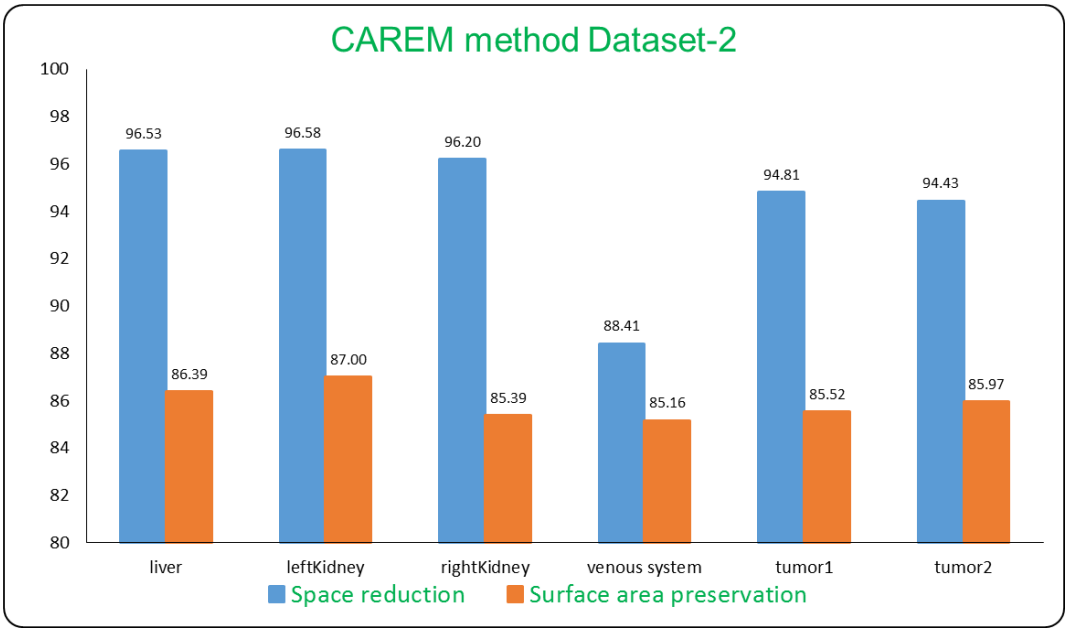


Figure 5.6: Surface area preservation by the *CAREM* method.

The reduction in surface area could be due to the smoothness required in three-dimensional bioCAD models. It is well known that a simple procedure, such as Laplacian smoothing shrinks significantly the feature-region domain (Taubin, 1995). Thus, to achieve a smooth surface for different feature-regions, the algorithm for smoothing reduces parts that exhibit large variation.

Table 5.6: Statistics for volume mesh generation from dataset-1 (top) and dataset-2 (button) applying the *CAREM* method.

Dataset-1	Voxels	Volume (mm3)	Points MC	Volume (mm3)	Points 3D mesh	Volume (mm3)
liver	962705	2133790	114026	2133391	12602	2131250
left kidney	82996	183957	20684	183841	1973	182798
right kidney	82902	183748	20224	183631	1276	181418
venous system	111763	247717	63390	245835	8338	239241

Dataset-2	Voxels	Volume (mm3)	Points MC	Volume (mm3)	Points 3D mesh	Volume (mm3)
liver	2865131	1489410	221402	1489300	6346	1487060
left kidney	280405	145766	46822	145727	1308	143265
right kidney	215358	111952	38582	111919	1057	110442
venous system	143180	74431	93714	73069	6926	70723.4
liver-tumor1	162337	84389	29510	84337	1144	83745.4
liver-tumor2	14464	7519	5228	7500	172	7108.93

5.2.3 Volume Preservation

A desirable feature of a three-dimensional bioCAD model is the volume preservation. The preservation of volume is an important feature required in different fields from Computer Graphics for making deformations look more natural, to dynamic systems for achieving more stable simulations. In this context, volume preservation has been studied for many jobs in recent years (Zhang et al., 2009), (Zhou et al., 2005), and several methods were proposed to reduce errors, such as volume boundaries or vertexes movement constrained to parameterized spaces (Frey and Borouchaki, 1998). Even with these restrictions, if smoothing is frequently applied within an adaptive process it tends to accumulate errors.

In Table 5.6 are listed some statistics related to the three-dimensional mesh generation process. These statistics include number of voxels and volume in mm3 of voxel models (medical segmentations) as well as number of points and volume of both surface meshes using the MC-based method, and three-dimensional bioCAD meshes obtained using the *CAREM* method.

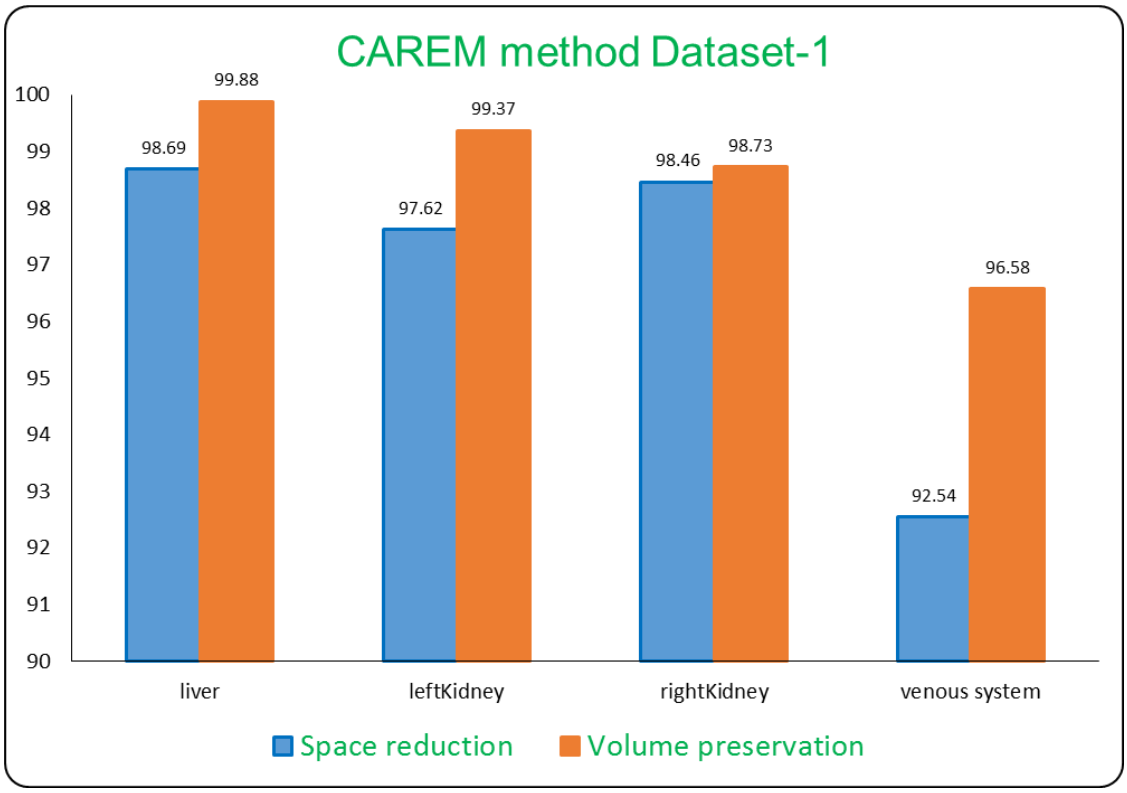


Figure 5.7: Space reduction and volume preservation obtained from voxel models to 3D bioCAD models of dataset-1 by the *CAREM* method.

In Figure 5.7 are shown the space reduction of each feature-region from dataset-1 between the MC model and the voxel model (MC-Voxel), between the three-dimensional bioCAD mesh and the MC model (3Dmesh-MC), and finally, between the three-dimensional bioCAD model and the voxel model (3Dmesh-Voxel). It can be observed from Figure 5.7 that surface reduction increases from the MC model to the bioCAD model ranging from 43% for the venous system, to 98.69% for the liver.

Figure 5.8 depicts the volume conservation of each feature-region from dataset-2 between the MC model and the voxel model (MC-Voxel); between the three-dimensional bioCAD mesh and the MC model (3Dmesh-MC), and finally, between the three-dimensional bioCAD model and the voxel model (3Dmesh-Voxel). It has been taken only dataset-2 because the volume loss ratio is almost the same as for dataset-1 which can be easily calculated from Table 5.6.

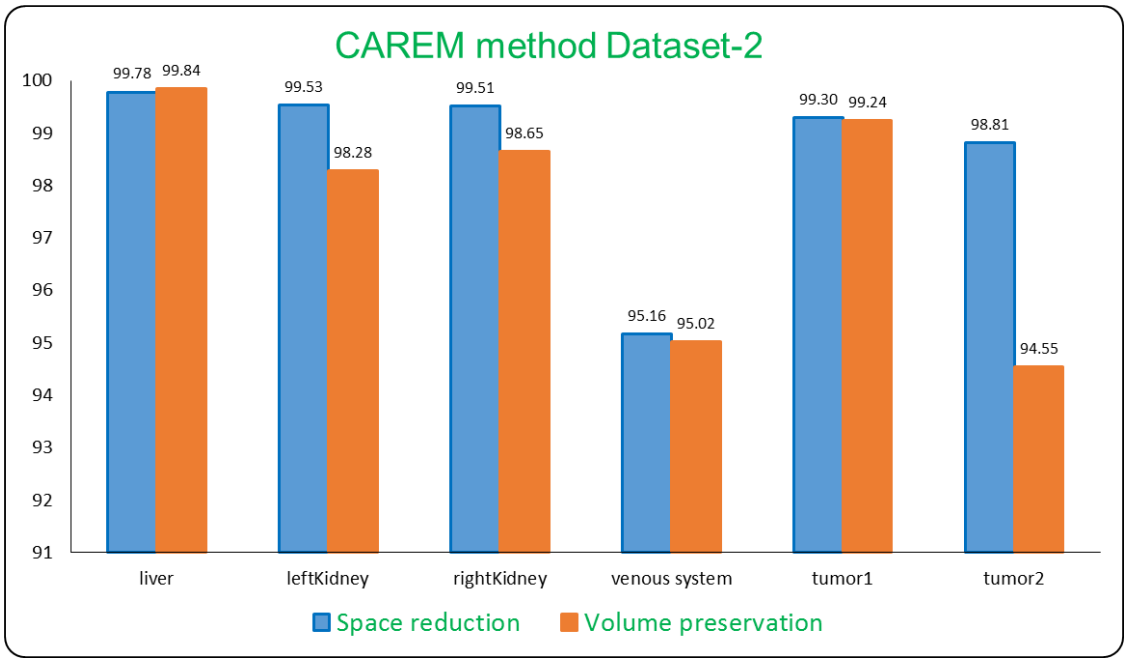


Figure 5.8: Space reduction and volume preservation obtained from feature-regions of dataset-2 by the *CAREM* method.

It can be observed from Figure 5.8 that the difference in volume between the three-dimensional bioCAD mesh and the MC model (3Dmesh-MC) ranges from 0.15 for the liver to 3.21 for the venous system. However, an remarkable case is the tumor2 that despite of being the most small object of all analyzed is which the greatest loss of volume presents, about 5.2%. This could be due to small number of voxels that represent the liver tumor2 and the size of these voxels. The same different applies for volume keeping between the three-dimensional bioCAD model and the voxel model (3Dmesh-Voxel). The only exception is the venous system which starts with a volume reduction of the 1.9% between the MC model and the voxel model (MC-Voxel), continues with a 3.2 when the volume mesh is generated. However, the total of volume keeping is about 95% that coincides with the space reduction.

Let us remember that the venous system was chosen because it is a structure with a very complex geometry that makes three-dimensional bioCAD modeling a challenging task due to multiple bifurcations (branches) that contains this type of structure. Additionally, the reconstruction algorithm can skip or cut some branches of the structure due to low number of points presented in those places.

5.2.4 Sensitivity Analysis

The *CAREM* method can produce accurate geometric models of anatomical organs and structures, however, parameters such as space reduction and Dice's coefficient are more-or-less uncertain. Moreover, both the Dice's coefficient and the Hausdorff distance are affected by complex geometries of feature-regions as well as by small number of points used to represent such regions. If parameters are uncertain, sensitivity analysis can give information such as: how accurate the final mode is in the face of different parameter values; under what circumstances the final model would change; and how the final model changes in different circumstances.

Although the liver of dataset-2 was taken for the sensitivity analysis the same analysis could be applied to any region. Table 5.7 depicts seven different experiments on the original liver and shows the number of points obtained from each experiment, the space reduction, and the Dice's coefficient.

Table 5.7: Sensitivity analysis applied on the liver organ model.

Experiment	Number of points	Space reduction	Dice's coefficient
Liver-original	221402	0	1
liver-1	138984	37,23	0,9953
liver-2	82903	62,56	0,9931
liver-3	33759	84,75	0,9911
LIVER-4	*7681*	*96,53*	*0,9946*
liver-5	5779	97,39	0,9805
liver-6	2982	98,65	0,9709
liver-7	1814	99,18	0,9578

Figure 5.9 shows the impact of changing the space reduction on the Dice's coefficient and therefore on the accuracy of the final model. The forth experiment labeled LIVER-4 is the best result, i.e, the higher space reduction (96,75 %) with a good Dice's coefficient value over 0.99 (0,9946). This means that when approximating the surface with fewer points the approximation error increases. This best result is given directly by the *CAREM* method (see Table 5.4).

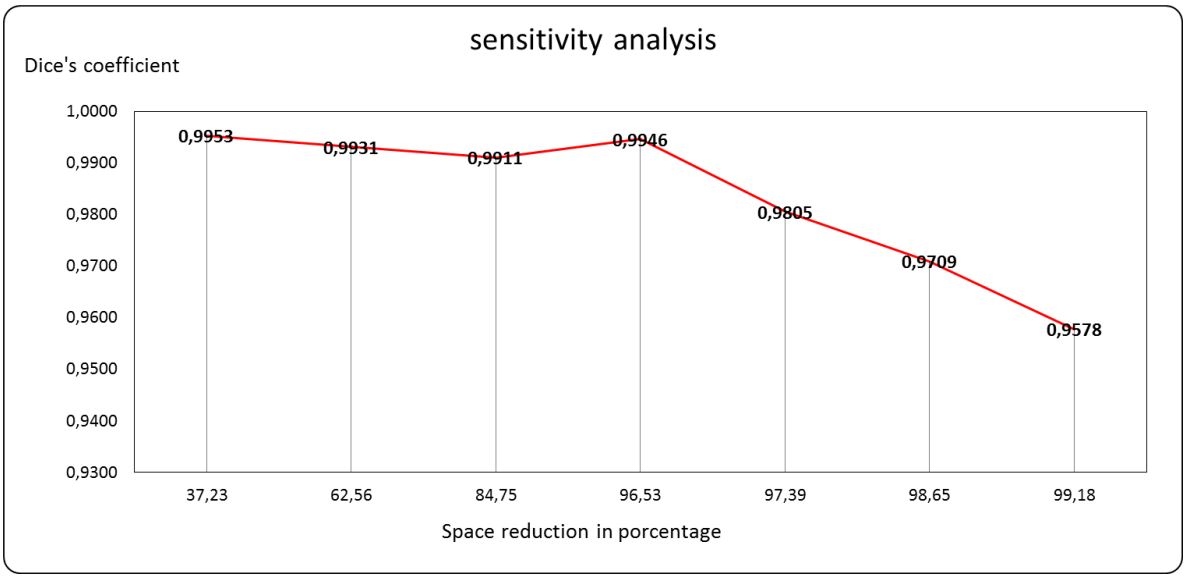


Figure 5.9: Graph sensitivity analysis applied between reducing disk space and the Dice coefficient of the liver organ model.

5.3 Modeling natural heterogeneous objects using the *RAM* method

The region-aware modeling method *(RAM)* lets model any object (live or inanimate) as it were a natural heterogeneous object composed of multiple regions. Once the heterogeneous object model is created three-dimensional bioCAD meshes are generated not only for visualization purposes, but for accurate finite element-based solutions. Next, different stages involved in the *RAM* method are brief summarized (see Chapter 4).

First of all, each feature-region that is part of the natural heterogeneous object is represented with an implicit function by using the *CAREM* method (see Chapter 3). Then, the set of implicit functions is organized in the vector implicit function structure through the Add_Featute_Region() operation (see Section 4.2.2). Finally, a three-dimensional bioCAD mesh is obtained using the Heterogeneous_Object_Geometry() operation.

In this section, quality of three-dimensional bioCAD meshes obtained when using the *RAM* method is evaluated. The same three criteria chosen for evalu-

ating the *CAREM* method, were taken for evaluating the performance of the *RAM* method: Accuracy (see Subsection 5.3.1), efficiency (see Subsection 5.3.2), and volume preservation (see Subsection 5.3.3). These criteria were applied on a multi-material volume model composed of four feature-regions created from dataset-1, and on a heterogeneous object model of the liver with two tumors embedded in it from dataset-2. Additionally, another heterogeneous object model called ***general*** was created as combination of a multi-material heterogeneous object with a heterogeneous object with embedded regions, also from dataset-2 (see Section 4.7).

5.3.1 Accuracy

Table 5.8 summarizes the results from computing the Dice's coefficient C_{Dice} using the MeshValmet tool (https://www.nitrc.org/projects/meshvalmet), between the known Voxel model (A) and the surface mesh (B) obtained from the bioCAD volume model. Let's remain that the *RAM* method does not generate surface meshes directly but only volume meshes. So two functionalities of the ParaView software (Ayachit, 2015) have been used for generating both the point cloud, and the surface mesh from the three-dimensional mesh: AppendGeometry and ExtractSurface.

It can be observed from Table 5.8 that the value of the Dice's coefficient is over 0.99 for all three cases. This means that the *RAM* method produces accurate heterogeneous object models of natural heterogeneous objects.

Table 5.8: Dice's coefficient between Voxel models obtained from dataset-1 and dataset-2, and surface meshes obtained with the *RAM* method.

Heterogeneous object model	Dice coefficient
Embedded	0.994090
Multi-material	0.992253
General	0.990395

Additionally, Table 5.9 summarizes the results from computing the Hausdorff distance $D_{Hausdorff}$ between the known Voxel model (A) and the surface reconstructed (B), using the standard tool METRO (Cignoni et al., 1998).

Table 5.9: Hausdorff distance between Voxel models obtained from dataset-1 and dataset-2, and surface model obtained with the *RAM* method.

Heterogeneous object model	Forward distance $d(A, B)$	Backward distance $d(B, A)$	Hausdorff distance
Embedded	4.432796	4.432796	4.432796
Multi-material	3.880045	3.424728	3.880045
General	13.910194	8.580923	13.910194

It is observed from Table 5.9 that the forward distance is taken as the Hausdorff distance. Moreover, for the heterogeneous object model with embedded regions the Hausdorff distance coincides with the Hausdorff distance of the region that embeds the other regions, that is what was expected. For instance, the Hausdorff distance of the embedded heterogeneous object is almost the same than the Hausdorff distance of the liver from dataset-2 (see Table 5.2-bottom).

5.3.2 Efficiency

In Table 5.10 are listed some time statistics making reference to point cloud generation, implicit surface reconstruction, and volume meshing of all three heterogeneous object models. However, it is important to clarify that the time for volume meshing is the only statistic that depends on the *RAM* method. The time related to the point cloud generation process and the implicit surface reconstruction process was calculated as the sum of the time required for each individual feature-region that is part of the heterogeneous object model which depends on the *CAREM* method.

Table 5.10: Time statistics (in seconds) for all processes involved in the *RAM* method for modeling feature-regions from dataset-1 (top) and dataset-2 (button).

Heterogeneous object model	Feature regions	Point cloud generation	Implicit reconstruction	Volume meshing
Embedded	3	3.82	21.37	15.91
Multi-material	4	8.49	17.95	29.35
General	6	7.62	35.64	41.75

Additionally, in Table 5.11 are listed some statistics related to the number of

points, the number of faces (triangles), and surface area in mm2, generated by both the MC-based method and the *RAM* method.

Table 5.11: Statistics for surface mesh generation using the *RAM* method.

Heterogeneous object model	Points MC	Triangles (faces)	Surface area	Points *RAM*	Triangles (faces)	Surface area
Embedded	256140	512256	99174	7051	14100	85505
Multi-material	218324	436526	227505	14067	28178	199425
General	435258	870048	151221	13028	26176	130887

Like for the *CAREM* method space reduction and surface area preservation are also two important features of the *RAM* method. Space reduction is in order 97.25%, 93.56% and 97% for the embedded, multi-material, and general heterogeneous objects respectively. Additionally, surface area keeping is of 86.22%, 87.66%, and 86.55% respectively.

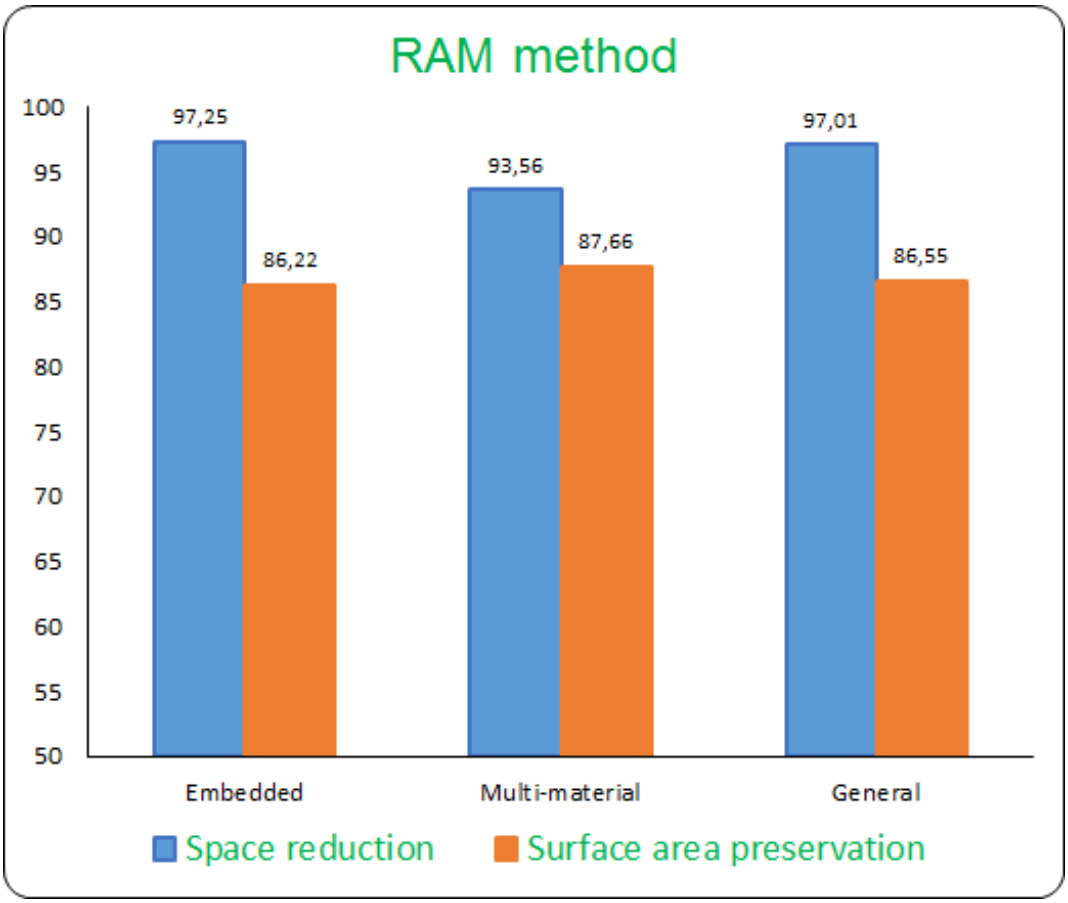


Figure 5.10: Statistics for surface mesh generation using the *RAM* method.

Figure 5.10 shows how the higher space reduction is less surface area preservation is. This is perfectly understandable since area calculation is based on triangles of the mesh surface, meaning that the less points you have (greater space reduction) lower surface to be covered from them (less surface area).

5.3.3 Volume preservation

In Table 5.12 are listed some statistics related to volume mesh generation process.

Table 5.12: Statistics for volumetric mesh generation from dataset-1 (top) and dataset-2 (button) using the *RAM* method..

Heterogeneous object model	Feature regions	Points of MC	Volume (mm3)	Points of 3D mesh	Volume (mm3)
Embedded with different cell-size	3	256140	1489303	10594	1487120
Embedded with same cell-size	3	256140	1489303	9110	1487110
Multi-material	4	218324	2658253	22480	2648500
General	6	435258	1803347	23580	1798760

These statistics include number of points along with the volume of both the MC model and the three-dimensional mesh obtained from the *RAM* method. An additional embedded heterogeneous object is included - one with different cell-size - to see differences in number of points and volume generated by the *RAM* method. However, those differences are minimal even improving volume conservation of the natural heterogeneous object.

Figure 5.11 shows how the higher space reduction is less volume preservation is. This is the same as for the surface preservation (see Figure 5.10), however, here the percentage of volume preservation is higher than the percentage of surface preservation.

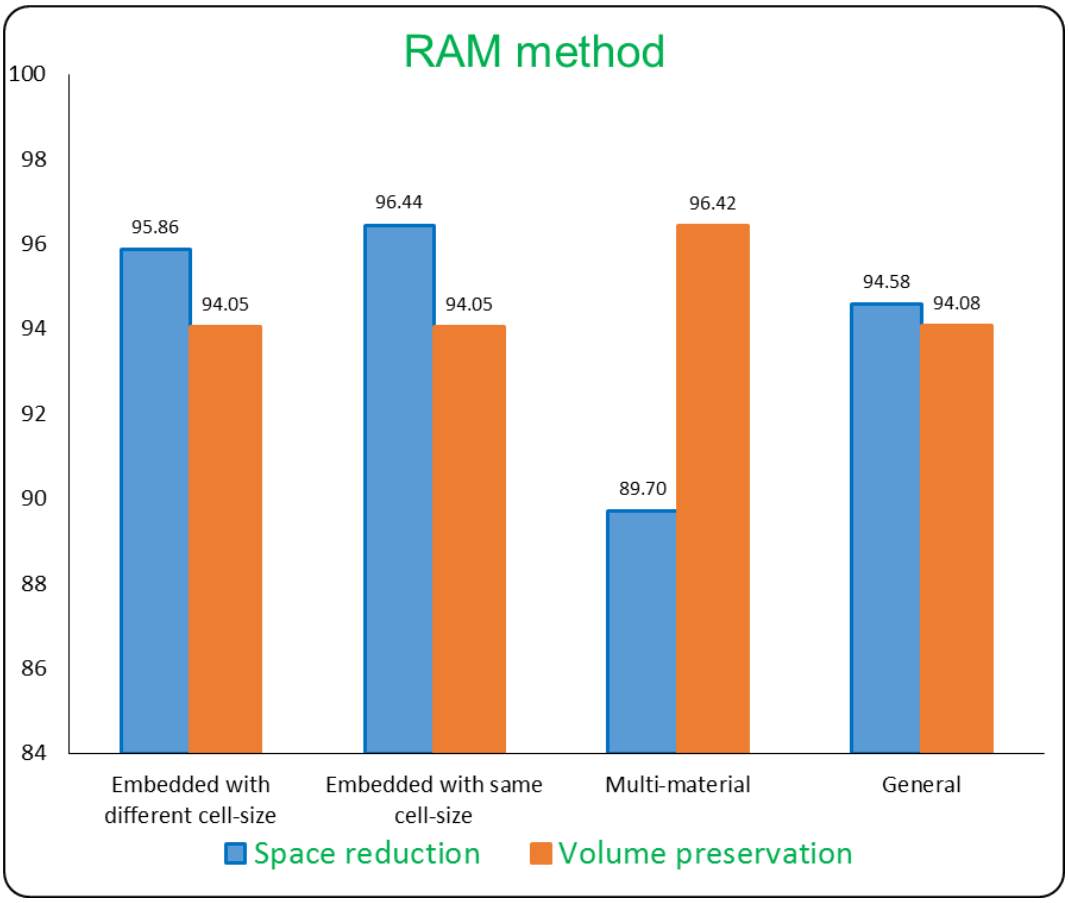


Figure 5.11: Volume statistics by the *RAM* method.

5.4 Discussion

Results presented in this chapter show that the proposed approach can work effectively in modeling natural heterogeneous objects. However, it can be observed that both the Dice's coefficient and the Hausdorff distance are affected by complex geometries of feature-regions as well as by small number of points used to represent such regions. Additionally, the time associated with the implicit surface reconstruction, the surface meshing, and the volume meshing, depends directly on the number of points, obtained from the Voxel model, and on the geometry complexity of feature-regions to be modeled. For example, the time for the implicit surface reconstruction of the liver from dataset-2 takes twice the time than for the liver from dataset-1 (see Figure 5.4), but the amount of points

used to represent that liver is also almost the double (see Table 5.4). This tendency is also repeated for the surface meshing process which is almost the double in the case of dataset-2 compared to dataset-1. But this tendency is different for the volume meshing process where the time needed for obtaining the volume model from the dataset-2 is higher than that from dataset-1, being the case of the venous system the most relevant (see Figure 5.4).

Although the overall modeling approach makes emphasis on physical fidelity, next factors affect accuracy specially of the *CAREM* method:

- Implicit surface reconstruction algorithm: In Subsection 2.3.2 three different algorithms were reviewed. The choice of the algorithm for generating the implicit surface from the input point cloud makes a difference. Although all algorithms should produce an implicit surface that partitions space into inside and outside regions, how points get interpolated or approximated will vary from one algorithm to the next.

- Meshing algorithm parameters: As for the implicit surface reconstruction the choice of the meshing algorithm for generating both surface meshes and volume meshes from the implicit representation makes a difference. Even if the same meshing algorithm is used, such algorithms often have input parameters. For example, in the *CAREM* method, three input parameters control the surface meshing procedure, and five parameters the volume meshing procedure. Changing parameter values will alter the results.

- Boundary coincidences: Boundaries share by more than two regions will take longer to determine where interfaces should be adjusted (fitted).

- With any choice of a medical image modality, the space is sampled and reconstructed mathematically, and as a result, accuracy of the reconstructed image is limited by accuracy of the reconstruction algorithm and the imaging modality resolution, i.e. three-dimensional bioCAD models can be obtained on a data set of higher quality. Higher resolution of CT data with less noise can be obtained by using a scanner with a small spacing field. The resulting segmentation would have less artifacts and the resulting finite element ready meshes would be more accurate.

Furthermore, next factors affect the efficiency of the *RAM* method:

- Number of regions: There is an overhead associated with each additional region that has to be added to the heterogeneous object model. The *RAM* method must move through the set of implicit regions and then determine the location of the boundary on the heterogeneous object model.
- Point cloud length: Longer point clouds require more time to obtain the implicit surface reconstruction. As the size of the segmented structures can vary greatly, this translates into a large difference in the number of acquired slices in which the structure appears.
- Shape complexity: More complex structures require more iterations of the algorithm to capture the details of the boundary and it will take longer for the modeler to determine if the generated model matches that three-dimensional shape. Intuitively, more complex shapes will be more difficult to contour than simpler ones.

Nevertheless, the modeling approach presented in this dissertation has important features which can be summarized as:

- Shape-preserving. The *CAREM* method minimizes geometric approximation errors so that resulting details appear similar to their original counterparts. The *RAM* method also generates a continuous smooth transition between interior regions and the exterior region. Study of cases for modeling feature-regions and heterogeneous objects are proposed in Section 3.6 and Section 4.7. In this chapter quantitative comparisons are given to show the efficacy of the proposed approach.
- Robustness. The present approach enables the user to model arbitrary number/shape of multiple and complex feature-regions to effectively display the entire heterogeneous object for visualization. It also allows the user to interactively specify geometric metrics for various simulation purposes. One of the most important quality metrics for surface meshes is the minimal angle that avoids having flat triangles. In all experiments both surface meshes and volume meshes were constrained to have triangles with minimum angle above 30 degrees as shown in Figure 3.15. The

experimental results show that the proposed approach can work effectively in modeling natural heterogeneous objects in different configurations.

- Efficiency. The computation is very efficient due to the implicit-based representation. The experimental results have demonstrated that this approach has great potentials in many biomedical applications. The present approach is well adapted to the large datasets encountered in medical imaging as it uses a recursive octree subdivision of the space in order to adapt to local surface details. Hence, computational time depends on surface complexity rather than image size. Moreover, the geometric approximation error can be user-controlled and bounded.

Chapter 6

Conclusions and future work

In this dissertation the research problem of three-dimensional modeling for natural heterogeneous objects was tackled applying a three-dimensional modeling approach based on reverse engineering from medical images. Since most natural heterogeneous objects are composed of many regions it is clearly desirable that the modeling approach incorporates this region-related information. Solid modeling techniques based on set-theoretic volume composition intrinsically support hierarchical part-based shape descriptions, however, natural objects such as anatomical organs and structures are more efficiently represented by surface modeling techniques. Additionally, in many engineering fields such as biomedical engineering, medicine or virtual reality, surface modeling is not enough for several applications where information about the interior of such natural objects is necessary and the generation of volume models is required.

In order to address the above problem, in this dissertation a systematic modeling approach consisting on two methods the *CAREM* method and the *RAM* method has been presented. This modeling approach starts from the base level, i.e., discrete voxel data, and allows to construct, visualize, and manipulate three-dimensional computer models using a representation based on implicit functions based on indicator functions. Experiments show that using a compact and consistent implicit representation greatly reduces the computer memory requirement. This is due to the fact that the indicator function is constant almost everywhere, and both the indicator function and its gradient were represented within an adaptive, hierarchical function space defined over an octree. This adaptive

representation have a spatial complexity of $O(n^2)$ and temporal complexity of $O(n^2 \log n)$.

This research has focused on modeling natural heterogeneous objects making emphasis on Accuracy (geometric quality and shape fidelity), Efficiency in terms of time involved from the beginning of the modeling process until the final three-dimensional model is obtained, and surface area preservation and volume preservation of feature-regions and of the whole natural heterogeneous object. In addition, quality surface meshes and tetrahedral meshes are generated automatically with conforming boundaries. These successful results suggest the validity of the *CAREM* method as well as of the *RAM* method, and of the whole approach in general for modeling natural heterogeneous objects.

It can be concluded that using a reverse engineering approach for modeling natural heterogeneous objects from segmented images through an implicit approximation seems preferable than reconstructing triangle meshes directly from such images. It has been shown that three-dimensional computer models defined this way enjoy a number of properties:

- Shape preserving so that resulting details of final three-dimensional models appear similar to their original counterparts.
- Robustness in modeling natural heterogeneous objects with arbitrary number of complex feature-regions.
- Computational efficiency due to the representation based on a vector implicit function.

Such three-dimensional models are capable of depicting accurately the interior of selecting organs and structures. Additionally, these models are suitable for significant data reduction and have potential applications in different fields of finite element analysis of natural heterogeneous objects. These features will make three-dimensional bioCAD models of natural heterogeneous objects be of great demand in practical applications from bio-engineering and medical research to multi-region modeling for fabrication. .

In Section 6.1 the principal contributions are presented. Section 6.2 outlines several avenues of future research based on the present work.

6.1 Major contributions

In this dissertation a systematic bottom-up modeling approach was proposed. In this approach, natural heterogeneous objects are decompose into separate region models and then integrates them into a global continuous implicit formulation. Two methods were designed to guarantee accuracy of region's representation while keeping quality of final bioCAD models. Some of the major contributions of this research include:

6.1.1 A computer-aided reverse-engineering based modeling method

Over the course of this research, different methods have been explored for converting discrete medical data into a fitting continuous counterpart that can be used later as a valid model representation. As a contribution, in Chapter 3 The computer-aided reverse-engineering based modeling method *(CAREM)* was proposed for three-dimensional computer modeling of anatomical organs and structures focusing on speed, precision and smoothness of the final bioCAD model. These models are able to represent hard and soft tissues. It has been shown that an implicit surface representation defined in terms of indicator functions represents a viable tool for surface reconstruction when segmented medical images are available.

6.1.2 A region-aware modeling method

Most of the existing modeling approaches target for modeling natural objects with simple geometries, but natural heterogeneous objects with multiple regions have been hardly modeled. As a contribution, in Chapter 4 a region-aware method *(RAM)* was proposed through which the natural heterogeneous object to be modeled is converted into a set of feature-regions. The *RAM* method takes advantages of a divide-and-conquer approach to model natural heterogeneous objects through a mathematical model that combines effects of the boundary set Γ_i together with the set of partitioned feature-regions Ω_i. Thus, geometry information of individual regions as well as of the whole natural object is incorporated

in the mathematical model. As a result, such models are resolution independent, more concise and mathematically rigorous.

6.1.3 A region-based decomposition approach

In this dissertation, an assembly modeling approach is used to build three-dimensional computer models of natural heterogeneous object using a region-aware structure. To convert the implicit represented heterogeneous object model into a set of two-manifold mesh surfaces (CAD model) that can be utilized in computational engineering applications, a vector implicit function *(*VIF*)* structure is proposed. This VIF keeps information about different parts of the heterogeneous object model allowing three-dimensional computer models to be generated with higher data consistencies and lower redundancies. The consistency is preserved on the interface of adjacent feature-regions, which cannot be solved by directly applying iso-surface extraction algorithms. In addition, by choosing different cell size and different refinement accuracy, mesh elements with different level-of-details can be easily determined. This is an important step towards creating three-dimensional bioCAD models of natural heterogeneous objects, obtaining adaptive and high-quality volume meshes.

6.1.4 A modeling framework

In this research, a three-dimensional anatomical modelling framework entirely written in C++ programing language was developed to generate three-dimensional models of anatomy from patient datasets. The generated models are suitable for use in virtual reality simulations, and CADx systems. Modern algorithms were employed and improved to allow the visualization, extraction, and meshing of these structures by building upon the functionality provided by several toolkits and libraries. ITK (http://www.itk.org/) is used for input and output of the volumetric data sets and for some image filtering and segmentation algorithms. VTK (http://www.vtk.org/) provides visualization of the data through direct volume rendering and polygonal mesh rendering (Schroeder et al., 2006); it also includes a wide variety of useful geometric algorithms. Finally, CGAL (http://www.cgal.org/) provides more advanced geometric algorithms, including

surface mesh generation (Project, 2016). The building of the application, toolkits, and libraries on different platforms is managed with CMake (http://www.cmake.org/) (Martin and Hoffman, 2004). This modeling framework helped to prove the validity of the presented modeling approach, and can be used in practical downstream applications for modeling and analysis of natural heterogeneous objects.

6.2 Future work

Research and development work on bioCAD modeling of natural heterogeneous objects is far from mature. The approach presented in this dissertation demonstrated only an alternative for three-dimensional bioCAD modeling of such objects. Towards practical applications, functionalities provided by the presented modeling approach is still too limited. Such applications include surgical simulation, virtual body exploring, as well as manufacturing and design of patient specific prosthetic devices. So, there is still much room for function improvements, and practical and theoretic developments. The studies conducted in this research can be further extended in a few important directions.

6.2.1 A GPU implementation

Modern graphics processing units (GPUs) are among the most powerful processing chips that exist today. State of the art GPUs are capable of over 12 teraflops of single precision floating-point arithmetic and have in excess of 250 GB/s of memory bandwidth – orders of magnitude more than current multi-core CPUs. We believe that the *CAREM* method and the *RAM* method have sufficient inherent data parallelism to perform well on GPUs. For future work, , a GPU implementation is to be proposed with performance speed-ups enough to reconstruct models of natural heterogeneous objects in real-time.

6.2.2 An integrated CAD-CAE system

Although bioCAD models, presented in this dissertation, can be measured, modified, and used in the design of biomedical devices, they cannot record the local

material compositions. This can be done by computer-aided engineering (CAE) models, but CAE models lack the capability of geometry manipulation. Therefore, CAD and CAE models in combination can realize a most complete heterogeneous object modeling process. For future work, a prototyping system is to be developed in order to conveniently implement an integrating CAD-CAE modeling of natural heterogeneous objects. To make the system really working, further research should be carried out to make full use of the capabilities of both bioCAD models and CAE models, and to build the tight connection in-between.

6.2.3 A mesh-free finite element analysis system

With recent developments of finite element analysis in various active research fields, such as computational medicine and computational biology, geometric modeling and mesh generation become more and more important for the simulation of the behavior of anatomical organs and biological structures. Implicit models of natural heterogeneous objects, obtained with the *CAREM* method and the *RAM* method, can be used to generate meshes adapted to requirements of finite element methods. However, the meshing process is time consuming as Subsection 5.3.2 has shown. For future work, a mesh-free modeling method is to be proposed. Such a method will rely on the implicit representation, presented in this dissertation, to satisfy all boundary conditions as accurately as desired.

6.2.4 A virtual-reality based simulation system

This dissertation has been focused on building three-dimensional computer models of natural heterogeneous objects while letting users browse them by simply rendering. However, techniques for interacting with three-dimensional computer models are also equally important for providing users with more rich and intuitive experiences. For future work a virtual-reality based simulation system is to be create in which users can realistically interact with three-dimensional bioCAD models. The system will take preprocessed CT image data for building bioCAD models using volume and surface modeling methods. These models need to be continuously updated during the virtual simulation.

Bibliography

Adzhiev, V., Kartasheva, E., Kunii, T., Pasko, A., and Schmitt, B. (2002). Cellular-functional modeling of heterogeneous objects. In *Proceedings of the seventh ACM symposium on Solid modeling and applications*, pages 192–203. ACM. 54

Angenent, S., Pichon, E., and Tannenbaum, A. (2006). Mathematical methods in medical image processing. *Bulletin of the American Mathematical Society*, 43(3):365–396. 30

Ayachit, U. (2015). *The ParaView Guide: A Parallel Visualization Application*. Kitware, Inc., USA. 130

Bajaj, C. (1997). *Introduction to implicit surfaces*. Morgan Kaufmann. 36

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517. 32, 34, 35

Bern, M., Edelsbrunner, H., Eppstein, D., Mitchell, S., and Tan, T. S. (1993). Edge insertion for optimal triangulations. *Discrete & Computational Geometry*, 10(1):47–65. 78

Biswas, A., Shapiro, V., and Tsukanov, I. (2004). Heterogeneous material modeling with distance fields. *Computer Aided Geometric Design*, 21(3):215–242. 54

Blinn, J. F. (1982). A generalization of algebraic surface drawing. *ACM Transactions on Graphics (TOG)*, 1(3):235–256. 52

Braid, I. C. (1974). *Designing with volumes*. Cantab Press, 2nd revised edition. 41

Caon, M. (2004). Voxel-based computational models of real human anatomy: a review. *Radiation and environmental biophysics*, 42(4):229–235. 56, 62

Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., and Evans, T. R. (2001). Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76. ACM. 44, 45, 46, 53

Cavalcanti, P. R., Carvalho, P. C. P., and Martha, L. F. (1997). Non-manifold modelling: an approach based on spatial subdivision. *Computer-Aided Design*, 29(3):209–220. 52

Chen, K.-Z. and Feng, X.-A. (2004). Cad modeling for the components made of multi heterogeneous materials and smart materials. *Computer-Aided Design*, 36(1):51–63. 49

Cheng, J. and Lin, F. (2005). Approach of heterogeneous bio-modeling based on material features. *Computer-Aided Design*, 37(11):1115–1126. 49

Cignoni, P., Rocchini, C., and Scopigno, R. (1998). Metro: measuring error on simplified surfaces. In *Computer Graphics Forum*, volume 17, pages 167–174. Wiley Online Library. 117, 130

De Santis, R., Ambrosio, L., Mollica, F., Netti, P., and Nicolais, L. (2007). Mechanical properties of human mineralized connective tissues. In *Modeling of Biological Materials*, pages 211–261. Springer. 67

Desaulniers, H. and Stewart, N. F. (1992). An extension of manifold boundary representations to the r-sets. *ACM Transactions on Graphics (TOG)*, 11(1):40–60. 38

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302. 115

Diehl, R. (1988). Conversion of boundary representations to bintrees. In *Eurographics' 88: proceedings of the European Computer Graphics Conference and Exhibition, Nice, France, 12-16 September, 1988*, page 117. North Holland. 32

d'Otreppe, V., Boman, R., and Ponthot, J.-P. (2012). Generating smooth surface meshes from multi-region medical images. *International Journal for Numerical Methods in Biomedical Engineering*, 28(6-7):642–660. 18

Evans, P. M. (2008). Anatomical imaging for radiotherapy. *Physics in medicine and biology*, 53(12):R151. 62, 63

Fabri, A., Giezeman, G.-J., Kettner, L., Schirra, S., and Schönherr, S. (1998). *On the design of CGAL, the computational geometry algorithms library*. PhD thesis, INRIA. France. 89

Feng, P., Ju, T., and Warren, J. (2010). Piecewise tri-linear contouring for multi-material volumes. In *Advances in Geometric Modeling and Processing*, pages 43–56. Springer. 90

Folland, G. B. (2013). *Real analysis: modern techniques and their applications*. John Wiley & Sons, 2nd edition. 46

Fram, J. R. and Deutsch, E. S. (1975). On the quantitative evaluation of edge detection schemes and their comparison with human performance. *Computers, IEEE Transactions on*, 100(6):616–628. 19

Frey, P. J. (2000). About surface remeshing. *In Proceedings of the 9th Int. Meshing Roundtable*, pages 123–136. 78

Frey, P. J. and Borouchaki, H. (1998). Geometric surface mesh optimization. *Computing and visualization in science*, 1(3):113–121. 125

Frisken, S. F., Perry, R. N., Rockwood, A. P., and Jones, T. R. (2000). Adaptively sampled distance fields: a general representation of shape for computer graphics. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 249–254. ACM Press/Addison-Wesley Publishing Co. 46

Gonzalez, R. C. and Woods, R. E. (2002). *Digital image processing, vol. 2*. Prentice-Hall Inc, New Jersey. 23

Greß, A. and Klein, R. (2004). Efficient representation and extraction of 2-manifold isosurfaces using kd-trees. *Graphical Models*, 66(6):370–397. 32

Hartmann, E. (1999). On the curvature of curves and surfaces defined by normal forms. *Computer Aided Geometric Design*, 16(5):355–376. 73

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. (1992). *Surface reconstruction from unorganized points*, volume 26. ACM. 42, 71

IRCAD (2015). http://www.ircad.fr/research/3d-ircadb-01/. 68, 113, 114

Jackins, C. L. and Tanimoto, S. L. (1980). Octrees and their use in representing three-dimensional objects. *Computer Graphics and Image Processing*, 14(3):249–270. 32, 33

Jackson, T., Liu, H., Patrikalakis, N., Sachs, E., and Cima, M. (1999). Modeling and designing functionally graded material components for fabrication with local composition control. *Materials & Design*, 20(2):63–75. 32

Jamin, C., Alliez, P., Yvinec, M., and Boissonnat, J.-D. (2015). Cgalmesh: a generic framework for delaunay mesh generation. *ACM Transactions on Mathematical Software*, 41(4):24. 80

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331. 67

Kaufman, A., Cohen, D., and Yagel, R. (1993). Volume graphics. *Computer*, 26(7):51–64. 29

Kazhdan, M. (2005). Reconstruction of solid models from oriented point sets. In *Proceedings of the third Eurographics symposium on Geometry processing*, page 73. Eurographics Association. 48

Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7. 42, 47, 48, 53, 75, 76, 77

Knopf, G. K. and Al-Naji, R. (2001). Adaptive reconstruction of bone geometry from serial cross-sections. *Artificial Intelligence in Engineering*, 15(3):227–239. 58

Kou, X. and Tan, S. (2007). Heterogeneous object modeling: A review. *Computer-Aided Design*, 39(4):284–301. 29, 49, 54, 75

Kou, X., Tan, S., and Sze, W. (2006). Modeling complex heterogeneous objects with non-manifold heterogeneous cells. *Computer-Aided Design*, 38(5):457–474. 49

Kumar, V., Burns, D., Dutta, D., and Hoffmann, C. (1999). A framework for object modeling. *Computer-Aided Design*, 31(9):541–556. 38, 49

Kumar, V. and Dutta, D. (1997). An approach to modeling multi-material objects. In *Proceedings of the fourth ACM symposium on Solid modeling and applications*, pages 336–345. ACM. 50, 52

Kumar, V. and Dutta, D. (1998). An approach to modeling & representation of heterogeneous objects. *Journal of Mechanical Design*, 120(4):659–667. 17, 49

Lien, S.-l. and Kajiya, J. T. (1984). A symbolic method for calculating the integral properties of arbitrary nonconvex polyhedra. *Computer Graphics and Applications, IEEE*, 4(10):35–42. 81, 83

Lin, C.-F., Yang, D.-L., and Chung, Y.-C. (2001). A marching voxels method for surface rendering of volume data. In *Computer Graphics International 2001. Proceedings*, pages 306–313. IEEE. 59

Löhner, R. (1997). Automatic unstructured grid generators. *Finite Elements in Analysis and Design*, 25(1):111–134. 100

Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. In *ACM siggraph computer graphics*, volume 21, pages 163–169. ACM. 58, 59, 69, 78

Mantyla, M. (1988). An introduction to solid modeling. *Principles of Computer Science Series*, pages 161–174. 28

Martin, K. and Hoffman, B. (2004). Mastering cmake: A cross-platform build system. *Kitware, Inc.*, 2nd edition. 142

Massey, W. S. (1991). *A basic course in algebraic topology*, volume 127. Springer Science & Business Media. 39

Meagher, D. (1982). Geometric modeling using octree encoding. *Computer graphics and image processing*, 19(2):129–147. 32

Menon, J., Wyvill, B., Bajaj, C., Bloomenthal, J., Guo, B., Hart, J., Wyvill, G., and Bajaj, C. (1996). Implicit surfaces for geometric modeling and computer graphics. *ACM SIGGRAPH-96 Course Notes*, pages 4–9. 37

Mortenson, M. E. (1997). Geometric modeling. *Wiley Computer Publishing, New York*, 2nd edition. 39

Moustakides, G., Briassoulis, D., Psarakis, E., and Dimas, E. (2000). 3d image acquisition and nurbs based geometry modelling of natural objects. *Advances in Engineering Software*, 31(12):955–969. 21, 58, 60

Naylor, B. (1990). Binary space partitioning trees as an alternative representation of polytopes. *Computer-Aided Design*, 22(4):250–252. 32

Nielson, G. M. (2000). Volume modelling. In *Volume Graphics*, pages 29–48. Springer. 30

Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., and Seidel, H.-P. (2005). Multi-level partition of unity implicits. In *ACM SIGGRAPH 2005 Courses*, page 173. ACM. 46, 53, 69

Ohtake, Y. and Suzuki, H. (2013). Edge detection based multi-material interface extraction on industrial ct volumes. *Science China Information Sciences*, 56(9):1–9. 18

Park, S.-M., Crawford, R. H., and Beaman, J. J. (2001). Volumetric multi-texturing for functionally gradient material representation. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*, pages 216–224. ACM. 52

Pasko, A., Adzhiev, V., Schmitt, B., and Schlick, C. (2001). Constructive hypervolume modeling. *Graphical models*, 63(6):413–442. 50, 54

Pasko, A., Adzhiev, V., Sourin, A., and Savchenko, V. (1995). Function representation in geometric modeling: concepts, implementation and applications. *The Visual Computer*, 11(8):429–446. 38, 53

Paterson, M. S. and Frances Yao, F. (1992). Optimal binary space partitions for orthogonal objects. *Journal of Algorithms*, 13(1):99–113. 32

Prince, J. L. and Links, J. M. (2006). *Medical imaging signals and systems*. Pearson Prentice Hall Upper Saddle River, NJ. 23, 62, 66

Project, T. C. (2016). *CGAL:User and Reference Manual*. CGAL Editorial Board, 4.8.1 edition. 142

Qian, X. and Dutta, D. (2003). Heterogeneous object modeling through direct face neighborhood alteration. *Computers & Graphics*, 27(6):943–961. 39

Reddy, D. R. and Rubin, S. (1978). Representation of three-dimensional objects. *Computer Science Department CMU-CS. Carnegie–Mellon University, Pittsburgh*, pages 79–113. 32

Requicha, A. (1977). Mathematical models of rigid solid objects. *Technical Report Memo 28*, University of Rochester, Production Automation Project. 38, 51

Requicha, A. and Tilove, R. (1978). Mathematical foundations of constructive solid geometry: General topology of closed regular sets. 38

Requicha, A. G. (1980). Representations for rigid solids: Theory, methods, and systems. *ACM Computing Surveys (CSUR)*, 12(4):437–464. 38

Ricci, A. (1973). A constructive geometry for computer graphics. *The Computer Journal*, 16(2):157–160. 37, 38, 53

Rvachev, V. L., Sheiko, T. I., Shapiro, V., and Tsukanov, I. (2001). Transfinite interpolation over implicitly defined sets. *Computer aided geometric design*, 18(3):195–220. 97

Samet, H. and Tamminen, M. (1985). Bintrees, csg trees, and time. In *ACM SIGGRAPH Computer Graphics*, volume 19, pages 121–130. ACM. 32

Samet, H. and Webber, R. E. (1988). Hierarchical data structures and algorithms for computer graphics. i. fundamentals. *Computer Graphics and Applications, IEEE*, 8(3):48–68. 32

Schroeder, C., Regli, W. C., Shokoufandeh, A., and Sun, W. (2005). Computer-aided design of porous artifacts. *Computer-Aided Design*, 37(3):339–353. 21

Schroeder, W., Martin, K., and Lorensen, B. (2006). *Visualization Tollkit: An Object-Oriented Approach To 3D Graphics*. 4th edition. 61, 141

Shin, K.-H. and Dutta, D. (2001). Constructive representation of heterogeneous objects. *Journal of Computing and Information Science in Engineering*, 1(3):205–217. 49

Siu, Y. and Tan, S. (2002). Source-based heterogeneous solid modeling. *Computer-Aided Design*, 34(1):41–55. 51, 52

Starly, B., Darling, A., Gomez, C., Nam, J., Sun, W., Shokoufandeh, A., and Regli, W. (2004). Image based bio-cad modeling and its applications to biomedical and tissue engineering. In *Proceedings of the ninth ACM symposium on Solid modeling and applications*, pages 273–278. Eurographics Association. 56

Stytz, M. R., Frieder, G., and Frieder, O. (1991). Three-dimensional medical imaging: algorithms and computer systems. *ACM Computing Surveys (CSUR)*, 23(4):421–499. 19, 60

Sun, W., Darling, A., Starly, B., and Nam, J. (2004). Computer-aided tissue engineering: overview, scope and challenges. *Biotechnology and Applied Biochemistry*, 39(1):29–47. 20, 56

Sun, W. and Hu, X. (2002). Reasoning boolean operation based modeling for heterogeneous objects. *Computer-Aided Design*, 34(6):481–488. 49, 51, 52

Szeliski, R. and Lavallée, S. (1996). Matching 3-d anatomical surfaces with non-rigid deformations using octree-splines. *International Journal of Computer Vision*, 18(2):171–186. 76

Taubin, G. (1995). Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 902–907. IEEE. 124

Thibault, W. C. and Naylor, B. F. (1987). Set operations on polyhedra using binary space partitioning trees. In *ACM SIGGRAPH computer graphics*, volume 21, pages 153–162. ACM. 32

Turk, G. and O'brien, J. F. (2002). Modelling with implicit surfaces that interpolate. *ACM Transactions on Graphics (TOG)*, 21(4):855–873. 44, 53

Van Dam, A., Feiner, S. K., McGuire, M., and Sklar, D. F. (2013). *Computer graphics: principles and practice*. Pearson Education. 41

Vese, L. A. and Chan, T. F. (2002). A multiphase level set framework for image segmentation using the mumford and shah model. *International Journal of Computer Vision*, 50(3):271–293. 95

Vivodtzev, F., Linsen, L., Bonneau, G.-P., Hamann, B., Joy, K. I., and Olshausen, B. A. (2003). Hierarchical isosurface segmentation based on discrete curvature. In *5th Joint Eurographics-IEEE TCVG Symposium on Visualizationf (VisSym'03)*, pages 249–258. ACM Siggraph. 19

Voelcker, H. and Requicha, A. (1977). Constructive solid geometry. *Technical Memorandum no. 25*, Production Automation Project, University of Rochester, Rochester, NY. 38

Wang, C. C. (2007). Direct extraction of surface meshes from implicitly represented heterogeneous volumes. *Computer-Aided Design*, 39(1):35–50. 92

Wang, L., Yu, Y., Zhou, K., and Guo, B. (2011). Multiscale vector volumes. *ACM Transactions on Graphics (TOG)*, 30(6):167. 18, 50

Wang, M. Y. and Wang, X. (2005). A level-set based variational method for design and optimization of heterogeneous objects. *Computer-Aided Design*, 37(3):321–337. 48, 54, 92

Wang, W. (2010). *Reverse engineering: technology of reinvention*. CRC Press. 59

Webster, J. G. (1988). *Medical Devices and Instrumentation*. New York: Wiley. 64

Weiler, K. (1988). The radial edge structure: a topological representation for non-manifold geometric boundary modeling. *Geometric modeling for CAD applications*, 1988:3–36. 39, 50

Wyvill, G., McPheeters, C., and Wyvill, B. (1986). Soft objects. In *Advanced Computer Graphics*, pages 113–128. Springer. 52

Yamaguchi, K., Kunii, T., Fujimura, K., and Toriya, H. (1984). Octree-related data structures and algorithms. *IEEE Computer Graphics and Applications*, 4(1):53–59. 32

Yuan, Z., Yu, Y., and Wang, W. (2012). Object-space multiphase implicit functions. *ACM Transactions on Graphics (TOG)*, 31(4):114. 18, 54, 90

Zaidi, H. and Tsui, B. M. (2009). Review of computational anthropomorphic anatomical and physiological models. *Proceedings of the IEEE*, 97(12):1938–1953. 21

Zhang, Y., Bajaj, C., and Xu, G. (2009). Surface smoothing and quality improvement of quadrilateral/hexahedral meshes with geometric flow. *Communications in Numerical Methods in Engineering*, 25(1):1–18. 125

Zhou, K., Huang, J., Snyder, J., Liu, X., Bao, H., Guo, B., and Shum, H.-Y. (2005). Large mesh deformation using the volumetric graph laplacian. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 496–503. ACM. 125