Accurate and high-quality surface mesh extraction from medical image data

About the cover

The front and back cover are meshes generated from a CT dataset of a human ankle. The back cover is generated using the Marching Cubes algorithm, the front cover is generated using the SurfaceNets algorithm (Chapter 3).

Accurate and high-quality surface mesh extraction from medical data

Proefschrift

ter verkrijging van de graad doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 8 november om 15:30 uur

door

Paul Wessel DE BRUIN

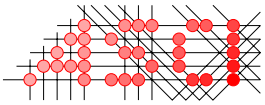
werktuigkundig ingenieur
geboren te Gouda.

Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. ir. F.W. Jansen

Samenstelling promotiecommissie:

Rector Magnificus, voorzitter
Prof. dr. ir. F.W. Jansen, Technische Universiteit Delft, promotor
Ir. F.H. Post, Technische Universiteit Delft, toegevoegd promotor
Prof. dr. ir. H.G. Stassen, Technische Universiteit Delft
Prof. dr. A.M. Vossepoel, Erasmus Universiteit Rotterdam
Prof. dr. J.D. Foley, Georgia Institute of Technology, Verenigde Staten van Amerika
Dr. S. de Blok, Onze Lieve Vrouwe Gasthuis, Amsterdam
Dr. F.M. Vos, Technische Universiteit Delft
Prof. dr. J. Dankelman, Technische Universiteit Delft, reservelid

Frans Vos heeft als begeleider in belangrijke mate aan de
totstandkoming van het proefschrift bijgedragen.



This work was carried out in the ASCI graduate school.
ASCI dissertation series number 106.

ISBN 90-77595-83-X

`http://visualisation.tudelft.nl/~paul/thesis`

They may lock up in their own bosoms the mysteries they have penetrated... whilst they reap in pecuniary profit the legitimate reward of their exertions. It is open to them, on the other hand, to disclose the secret they have torn from nature, and by allowing mankind to participate with them, to claim at once that splendid reputation which is rarely refused to the inventors of valuable discoveries in the arts of life. The two courses are rarely compatible.

Charles Babbage [117]

Preface

The research described in this thesis was conducted at the Computer Graphics and CAD/CAM Group of the Faculty of Electrical Engineering, Mathematics, and Computer Science of Delft University of Technology. It is part of two research programmes of the Delft Inter-Faculty Research Center on Medical Engineering (DIOC 9): Minimally Invasive Surgery and Intervention Techniques (MISIT) and Development of Improved endoProstheses for the upper EXtremities (DIPEX).

The work before you is not entirely my fault. As these things go in large projects, many people have contributed their valuable time, ideas, opinions and support. I thank them all, and some in particular.

Frits Post, my supervisor, for his guidance, patience, and support. My project on medical visualisation was a break with a long tradition of flow visualisation and feature detection in his group. Nevertheless, Frits' extensive grasp of visualisation methods and knowledge of the literature were welcome partners in discussions when things seemed to be at a dead end.

Erik Jansen, my promotor, for giving me the opportunity to work in his group, and for the helpful and quick feedback on my thesis-in-progress.

Frans Vos, my other supervisor, for the discussions, good times at conferences, games of squash, and dinners with a good price-quality ratio. Frans is for a large part responsible for (ever so gently) forcing me to structure my work and writing into an academic mold; he has donated a substantial part of his time to me, for which I thank him.

Albert Vossepoel, my other other supervisor, *pater familias* of the MISIT 3, and later DIPEX 2 projects, for his voice of reason, and interest in my work. Linda and I really enjoyed visiting Cambridge and Tokyo with Wies and you.

Sjoerd de Blok, for his enthusiastic support and giving me my first proper understanding of how difficult my project really was. The people of the gynaecology and radiology department at the OLVG, for being very helpful and friendly.

Sarah Frisken and Jim Foley, for giving me the opportunity to work at MERL for four months. It was a thoroughly enjoyable experience. Jeroen van Baar for helping me finding my way around MERL. Howard for giving Linda and me an unforgettable Millennium New Year's Eve on a Boston brownstone rooftop.

The three M.Sc. students who contributed to this Ph.D. project. Magchiel van Meeteren worked on mesh improvement and edge detection (as well as my vocabulary and language). Vincent Dercksen was the driving force behind the manual interactive

segmentation technique. Finally, Jasper Aubel worked on active shape modelling of the carpal bones. It was fun, and many ideas were born during our discussions. I am proud that each of you is co-author on a paper about your work.

Geert Streekstra, for co-guiding Vincent and Jasper at the AMC. Jaap Stoker and Mario Maas from the AMC Radiology Department for testing an early version of the manual segmentation software.

The people from Kretz, for giving on loan a 3D ultrasound machine for part of this research.

The *x*io's: Benjamin, Alex, Rafa, Michal, Eelco, and the student lab: Geoffrey, Berg, Rick, André, Jorik, and Joris, for creating an enjoyable working environment with a steady supply of coffee in the Computer Graphics and CAD/CAM Group.

The system administrators Bart *Mac* Vastenhouw, Gert *Disturbing-colours* Bultman, and Ruud *Pragmatist* de Jong, for providing system administration with a moustache.

Toos Brussee, the *mater familias* of the group, for providing the non-technical essentials.

My colleagues Freek Reinders and Ari Sadarjoen, for welcoming me into the wonderful world of visualisation and helping me getting started with my work.

Charl Botha, my roommate and good friend, for being the perfect devil's advocate, living next to the best Belgian beer pub in Delft, and being an all-round smartypants: a deadly combination.

My good friends Felix Chang, for the fontological magic, and Jochem Welvaadt, for hiring my services temporarily with all consequences.

My mother Truus, sister Marije, grandmother Dini, and aunt Annie for doing what mothers, sisters, grandmothers and aunts do.

Finally, Linda, *mijn maatje*, without whom all this would not have been and possible.

Thanks,

Paul de Bruin

Contents

Introduction 1

This thesis researches methods for medical visualisation. The goal of visualisation is to help someone obtain a deeper understanding of data. In hospitals, medical imaging modalities, such as Magnetic Resonance (MR) and Computed Tomography (CT), are prevalent and a large amount of patient-specific data is scanned routinely. The goal of medical visualisation is to make the contents of this data available to a clinician.

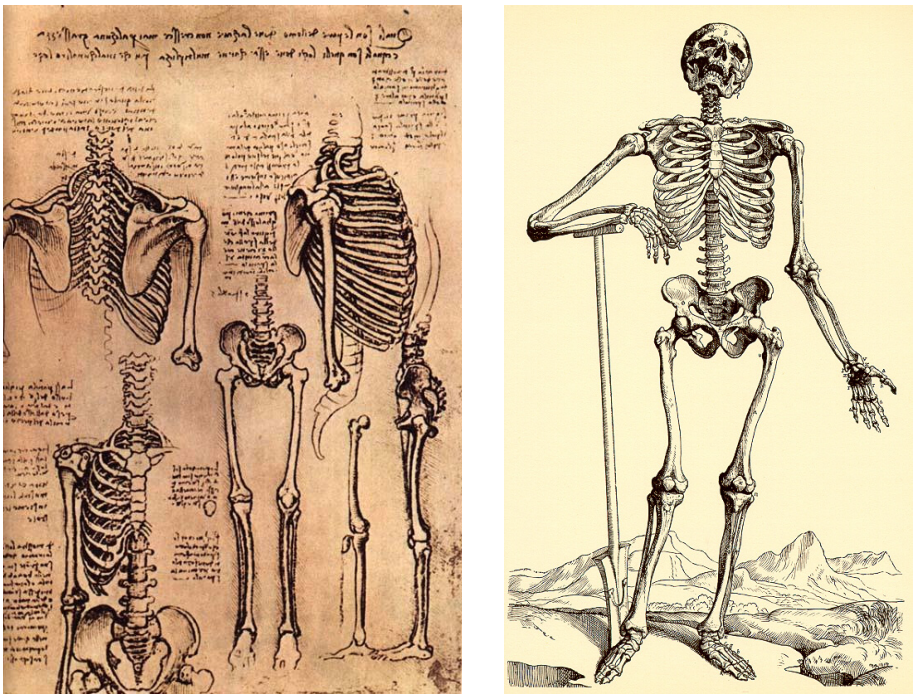
1.1 A short history of medical visualisation

The early Renaissance marks the beginning of modern medical visualisation. Between 1489 and 1515 Leonardo da Vinci produced beautiful drawings of his studies of the human body (Figure 1.1 on the following page). These drawings were intended for a book by the Paduan anatomist Marc Antonio della Torre. Unfortunately, the book was never published and the drawings were not very well known at the time [77].

In 1543[1], Vesalius published his ground-breaking textbook of human anatomy "De Humanis Corporis Fabrica". This set of seven books contains many highly detailed drawings of human anatomy, in various stages of dissection. Although some of the drawings were made by Vesalius himself, most of the larger pieces are attributed to Titian (Tizian Vecelli, 1485–1576) and his pupil Jan Steven van Calcar (1499?–1546?). The hand of an artist is clearly visible in the posture and setting of the skeleton in Figure 1.1 on the next page. Some of the backgrounds are attributed to Domenico Campagnola (1500–1546) who worked as a landscape painter in Titian's studio [77].

Slowly, the dissection of human bodies for research became more and more accepted. For example, the Greek-Roman Galen of Pergamon (130-200) obtained his knowledge from dissecting pigs, dogs, Barbary apes, and cattle; his *De usu partium* was until Vesalius' time *the* textbook on human anatomy. However, discoveries made by dissecting animals cannot always be applied to humans. Galen proposed that the energy of the body (the vital spirit) was carried through the arteries at the level of the *rete mirabilis*, where it was transformed into nerve impulse (the animal spirit). However, the *rete mirabilis* is a network of veins at the base of the brain found in cattle, not in humans. In later times, Vesalius obtained his first skeleton from a gallows

[1]Also in the year 1543, Copernicus published "De Revolutionibus Orbium Celestium" which discarded the notion of Ptolemy that the universe is centered around the earth



(a) Leonardo da Vinci (1452–1519) (b) Andreas Vesalius (1514–1574)

Figure 1.1: Early examples of medical visualisation. On the left Leonardo da Vinci's "Drawing of the Torso and Arms" – Bibliotheca Ambrosiana, Milan. On the right an image from Andreas Vesalius' "De Humani Corporis Fabrica Libri Septem" (from book 5).

and smuggled the bones across the city walls. One of the first textbooks on human anatomy that was at least partly derived from human dissection was published in Bologna by Mundinus (1275–1326) [106].

The next breakthrough is the invention of the microscope in the early 17th century. In 1665 Robert Hooke (1635–1703) describes in his "Micrographia" his observations of plant tissues (and coins the term *cell*). The book contains a collection of illustrations made using a compound microscope. Although Antoni van Leeuwenhoek (1632–1723) did not invent the microscope (in fact he first held a microscope probably in 1670), he is regarded as the father of microscopy because he was the first to achieve a high-quality magnification of 200. His single-lens microscopes are famous for their clarity, resolution and power of the lenses. One of the best of the

surviving lenses is 1.2mm thick, has a radius of curvature on both sides of roughly 0.7mm, has a focal length of slightly less than 1mm and a magnification of approximately 270 [108]. Such high-quality instruments enabled Van Leeuwenhoek to study and describe objects in great detail, for example, he reported the existence of bacteria in plaque.

Arguably the most revolutionary event in the history of medical imaging is the discovery[2] of X-rays in 1895 by Wilhelm Conrad Röntgen (1845–1923). His preliminary report "Über eine neue Art von Strahlen" was given on December 28th 1895 to the president of the Würzburg Physical-Medical Society together with experimental radiographs and an image of his wife's hand. This discovery went around the world in a very short time and was used clinically almost instantly. Röntgen declined to seek patents or proprietary claims on his discovery and was awarded the first Nobel prise in physics in 1901 (but he bequeathed the Nobel prise money to scientific research at Würzburg). From then on it was possible to examine internal structures of the body *in vivo* without damage to the patient (at least, not too much). However, the X-ray images are 2D – and more importantly – the images are projections. Interpretation of these images is therefore required.

Next, the now common medical imaging modalities CT, MRI, and lately 3D ultrasound were introduced [132]. The first clinical CT machine was introduced in 1972 [4, 3, 65], the first commercial MRI scanner in 1978 (the same year when the first abdominal MR scan was reported [85]), and 2D ultrasound in the 1950's. The first 3D ultrasound experiments were reported in 1984. Each of these modalities produces images based on a different physical principle. Correct interpretation of these images is therefore very important. Fundamentally, this touches on the question of what exactly constitutes an object. For example, tissue in CT images is uniquely identifiable by its Hounsfield number. MR can be tuned to a high degree to produce desired contrasts, and therefore, MR images of the same body part may look very different. Finally, ultrasound images the transition between different tissues and consequently produces data that can be considered to be a gradient image of the object.

The output of a modern medical imaging device is 3D data volume. In a sense, we have come full circle. In the beginning the actual body(part) was examined in detail *ex vivo*, now the same part can be examined virtually *in vivo*, with minimal risk for the patient.

1.2 Motivation and goal

Medical imaging devices, such as MR or CT, produce large volumes of patient data. Traditionally, clinicians are comfortable viewing 2D images of X-ray projections, and

[2]Nikola Tesla (1856–1943) independently discovered X-rays around the same time [5].

therefore, a 3D volume is often presented as a stack of 2D images. However, reconstructing a mental 3D image from 2D slices is a difficult and error-prone task. Performing this task on time-varying data is almost impossible. Clearly, help is required in processing, viewing, and interpreting this data.

Analysing these large 3D datasets can be performed by creating models that represent anatomical structures in the data. Geometric and mechanical models are important for surgical training, diagnosis, planning, simulation and visualisation. A common modelling approach is to create 3D surface representations that approximate the boundaries of anatomical objects. Such surface meshes consist of sets of connected polygons – usually triangles or quadrilaterals. This representation is well suited for intuitive viewing and manipulation, and plays an important role for visualisation during minimally invasive procedures where doctors need to *fly–by–wire*. Properties of surface meshes, such as surface area and volume, can be quantified and used for further analysis. For simulations, volume meshes – usually consisting of sets of connected tetrahedrons – can be created from surface meshes [52].

The application of surface meshes to describe and analyse medical data can be performed only under certain restrictions. Surface meshes, and the methods to generate these, should conform to a set of criteria regarding integrity, accuracy, triangle quality, compactness, smoothness, and reproducibility:

- *Integrity:* the mesh should be free of geometric and topological errors. For example, the polygons should be oriented consistently, and the surface should not self-intersect;

- *Accuracy:* the representation of the organ surface geometry should be sufficiently accurate;

- *Triangle quality:* the shape of the triangles in the mesh should be as near as possible to equilateral to aid subsequent analysis, for example, simulation by finite element methods and visualisation;

- *Compactness:* to achieve fast response times, the number of elements (triangles) in the model should be minimal; consequently, the resolution of the triangle mesh should be considerably lower than the medical image, with minimal loss of accuracy;

- *Smoothness:* the model should conform to the smooth organ boundaries. Sharp corners should be avoided as these can cause disturbing artifacts such as stress concentrations in a finite element simulation. Of course, if a shape is not smooth then the mesh should conform. However, within the space allowed by the inaccuracy, the mesh can be made as smooth as possible;

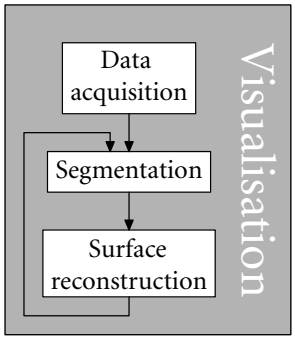


Figure 1.2: The medical visualisation pipeline.

- *Reproducibility:* the method used to create the surfaces should yield the same results for different users upon repeated extraction.

This set of criteria taken as a whole are at the basis of the goal of this work.

The extraction of surfaces is part of a chain of methods for medical visualisation. Figure 1.2 sketches a model for medical visualisation, containing from top to bottom: data acquisition, segmentation, and surface reconstruction as required components.

After data acquisition, segmentation is performed to separate objects of interest (tissue, bones, organs, tumours) from the background. Commonly, segmentation results in a classification of each point in a dataset. In this work three basic segmentation methods are used: thresholding (the basis for iso-surface extraction, Chapter 3), edge-detection (based on gradient information, Chapter 4), and manual segmentation (Chapter 5). Each of these methods has certain properties that are discussed extensively. In the last step, after segmentation, a surface is constructed that connects classified data points, where an optimum is sought between the accuracy of the fit and the quality of the surface as specified by the criteria above. The goal of this work is to find this optimum by connecting the segmentation and surface extraction steps, as is illustrated in Figure 1.2.

1.3 Outline

The main part of this work consists of a six published papers that can be read separately. These papers are incorporated verbatim with minor adaption of the layout and typography. The subject of research is a moving target and therefore some Chapters contain an addendum with new material.

Chapter 2 gives a survey of the problem field of surface extraction and mesh improvement.

Chapter 3 consists of *Improving Triangle Mesh Quality with SurfaceNets* [27]. In this paper a method is presented to create an iso-surface from volume data. The method consists of an iterative process with consecutive steps of surface extraction and mesh improvement.
Chapter 4 is *Accurate and High Quality Triangle Models from 3D Grey Scale Images* [26]. In this paper a method is presented that creates an accurate surface using a multi-scale edge-detection technique. This is also an iterative method where edge-detection steps are followed by mesh improvement steps.
Chapter 5 is *Interactive 3D segmentation using connected orthogonal contours* [25]. In this paper a fast interactive segmentation method is presented that integrates manual and semi-automatic segmentation. The method is based on interacting with a set of planar, orthogonal and connected contours in 3D.

Next, two Chapters with applications are presented.
Chapter 6 contains *Interactive matching of ultrasound and MRI for visualization during resection of myomata* [28]. This paper outlines a method for manual registration of MRI and 3D ultrasound volumes based on a small number of anatomically positioned landmarks.
Chapter 7 contains *A Statistical Shape Model of Carpal Bones* [131] where a statistical shape model of two carpal bones is researched. A new method is used for creating automatic landmark correspondences. The fast interactive segmentation method is used to create initial shapes.

Finally, in **Chapter 8** conclusions and directions for further and future research are given.

Surface mesh extraction and improvement techniques 2

Creating a conforming surface mesh, following the criteria set out in Section 1.2, requires (a combination of) surface extraction and mesh improvement techniques. Accordingly, this chapter is organised into two parts. In the first part, Section 2.1, an overview is given of surface extraction and reconstruction techniques. The second part, Section 2.2, surveys a selection of mesh improvement techniques.

2.1 Surface mesh extraction

2.1.1 Introduction

A common format for 3D medical data is a regular, Cartesian grid (see Section 2.2.1) where on each grid point a data value is specified (Figure 2.1 on the following page), also called the *cuberille* model. The data value usually is scalar and is also referred to as grey value. A voxel is the block around each grid point (indicated by the dashed line in Figure 2.1 on the next page) scaled such that neighbouring voxels do not overlap and all voxels fill the complete data space. Similarly, a cell is defined by the space between eight neighbouring data values that form a cuboid. Commonly, the data value is assumed to be constant over the voxel.

The process of extracting a surface mesh from 3D data consists of two parts: segmentation and surface reconstruction. Segmentation techniques are used to find the boundary and contours of anatomical objects. In this thesis two common segmentation methods are employed: thresholding (as the basis for iso-surfacing) (Section 2.1.2) and edge-detection (Section 2.1.3). After the segmentation step, a mathematical description of the surface is reconstructed. In this thesis the focus is on surface reconstruction using medical data. Overviews of generic surface extraction methods can be found in [21] and [127].

2.1.2 Iso-surface techniques

An often-used method to find a boundary in 3D data is thresholding in combination with an iso-surface technique. Iso-surfacing is probably the most often used surface extraction technique. Its popularity can be attributed to its simplicity and the fact

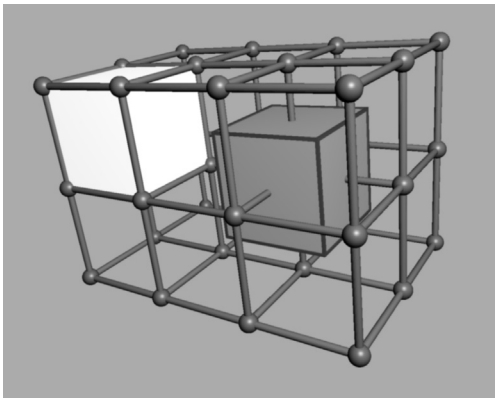


Figure 2.1: A regular grid, where each grid intersection is a data point (indicated by a sphere). A *cell* is formed by the cube connecting 8 data points (indicated by the white cube). A *voxel* is the cube formed by connecting the centers of the eight cells that surround a grid intersection (indicated by the darker cube).

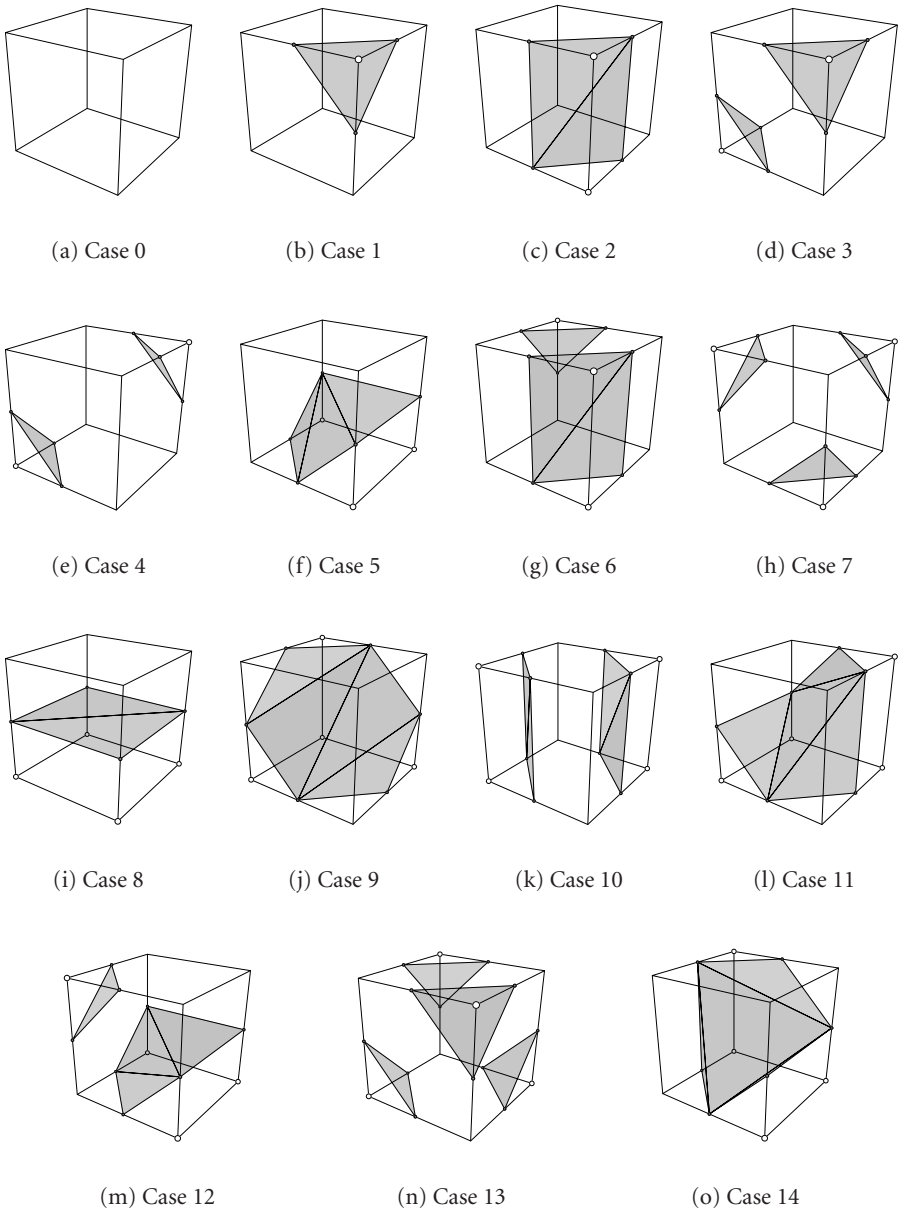
that it forms a good starting point for exploration of unknown data.

In general, an iso-contour is defined by the boundary on which a scalar function is constant. In 2D, an iso-contour is defined by $f(x,y) = I$, where I is a constant – the iso-value. Similarly, in 3D, the function $f(x,y,z) = I$ defines an iso-surface. Well-known applications of iso-contours are topographical height maps where iso-lines connect points of constant height and weather maps with isobars and isotherms.

Generally, iso-surface extraction algorithms create a polygonal approximation of an iso-surface. Methods for extracting such approximations based on a defining implicit function f can be found in [17, 140]. More common is the case where a dataset is available that is a sampling of an unknown function f. In that case, the extraction of an iso-surface is performed using the 3D array of discrete data points, $d(i, j, k) = I$. Usually, iso-surface extraction algorithms perform a search to identify the cells in the data that contain part of the iso-surface, for example, by detecting if a certain threshold occurs. Next, in each of these cells a polygonal approximation of the iso-surface is constructed.

Iso-surfacing has proven to be particularly useful for the extraction of surfaces from CT data. The intensity, or Hounsfield value, in CT corresponds directly to a specific tissue. For certain tissues the Hounsfield values are known and can be used as an iso-value, e.g., to extract bone.

The most common polygonal iso-surface extraction technique is the Marching Cubes algorithm (MC) [81]. MC operates on a volume that is divided into discrete cells (see Figure 2.1). The data value on each corner of each cell is tested whether it is under or above the threshold and the result (0 or 1) is stored in an 8 bit array for each



(a) Case 0 (b) Case 1 (c) Case 2 (d) Case 3

(e) Case 4 (f) Case 5 (g) Case 6 (h) Case 7

(i) Case 8 (j) Case 9 (k) Case 10 (l) Case 11

(m) Case 12 (n) Case 13 (o) Case 14

Figure 2.2: Marching Cubes case table.

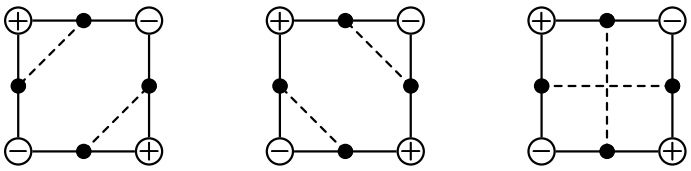


Figure 2.3: Topological ambiguity problem. Different topological configurations are possible based on a single state of the corners. MC chooses one of the first two arbitrarily. The third configuration is always used in SurfaceNets (see [60] and Chapter 3 of this thesis).

(a) Case I (b) Case II

Figure 2.4: Ambiguous Marching Cubes cases. Different surface topologies can be fitted to one corner configuration. The corner configuration corresponds to Figure 2.2(d) on the page before.

cell. The iso-surface passes through each cell where this classification is not equal to 0 (all corners below threshold) or 255 (all corners above threshold). The classification code is used as an index into a set of cell intersection cases (see Figure 2.2 on the page before). These intersection cases specify through which edges of the cell the surface passes. The 256 possible cases can be reduced to 15 unique cases by rotation, mirroring and inversion of the state of all corners. Finally, the exact intersections along each edge are linearly interpolated and triangles are constructed in each cell following the template case. The memory overhead of Marching Cubes is low; no intermediate datastructures are needed because the triangles of the surface are generated directly.

Marching Cubes variants

A large body of literature exists with variants of and improvements on the original Marching Cubes algorithm. One direction of research is to resolve the topological ambiguities of MC. An example of such an ambiguity is pointed out in [59] and is shown in Figure 2.3. Given the configuration of the corners of the cell, different surface topologies are possible. A similar situation for the 3D case is shown in Figure 2.4.

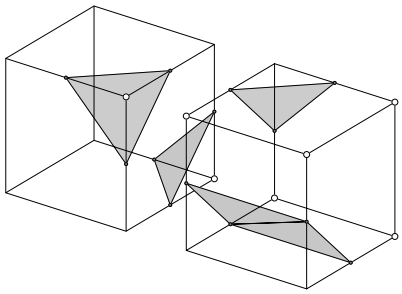


Figure 2.5: Marching Cubes cases where holes can be created. The facing sides of both cells have the same corner configuration but the surface topologies do not match. The left configuration corresponds to Figure 2.2(d) on page 9, the right configuration corresponds to Figure 2.2(g) on page 9.

Also, it is possible to create surfaces with holes (see Figure 2.5). The Asymptotic Decider [97] is a method to solve some of these problems. Topological ambiguity is resolved by using bilinear interpolation of the faces of the cells. A more rigourous treatment by the same author [96] presents an extended set of corner configurations (see Figure 2.2 on page 9) and uses trilinear interpolation.

Other variants aim at reducing the number of triangles that are generated. The Marching Cubes algorithm can generate a maximum of 4 triangles per cell (see Figure 2.2 on page 9). Discrete Marching Cubes [91, 92] is a two-stepped approach to triangle reduction. In the first step, the edge interpolation is replaced by midpoint selection. After doing so, coplanar triangles are merged easily. In the second step the edge intersections are moved to the exact locations found by linear interpolation along the edge. The final surface is an approximation, but the error is bounded by the longest edge length. The results show that Discrete MC compares favourably to MC followed by mesh simplification both in required computation time as well as in the amount of generated triangles. Other reduction variants can be found in [113], where octrees are used for decimation and in [136] where octrees are used to speed up iso-surface generation. An approach called Precise Marching cubes is presented in [2] that is based on mesh refinement. Each of the triangles found by MC is refined adaptively by using trilinear interpolation to approximate the isosurface. By doing so, the accuracy of the approximation is controlled adaptively. Extended Marching Cubes (EMC) is a variant that aims to improve the appearance of sharp edges [73]. This algorithm is treated in Section 3.5.1.

Finally, there are MC variants that use optimised cell division, traversal and search algorithms, e.g., span-space [36, 78, 114] and octrees. A comparison of several storage and searching methods can be found in [116]. Iso-surfacing can be performed in higher dimensions [14], on complex-valued datasets [133], and using a multi-

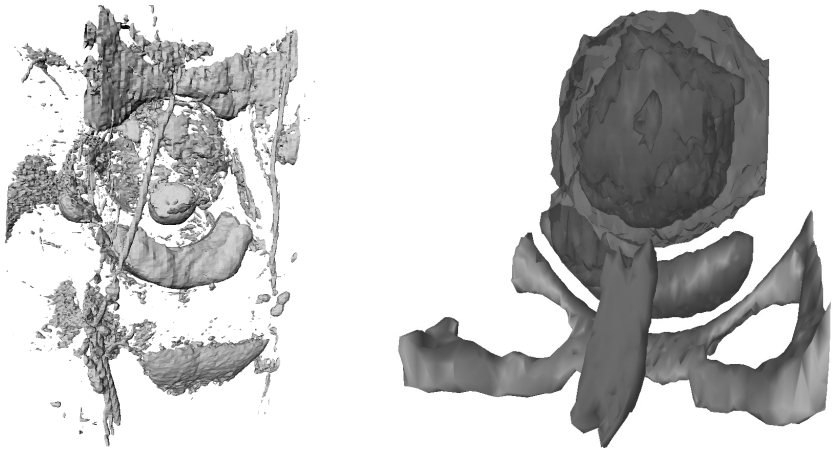


Figure 2.6: Iso-surfaces generated by Marching Cubes from a MRI dataset of a human abdomen. The figure on the left is generated by using a single-isovalue on the raw image. The figure on the right was generated after the image was classified by hand, iso-surfaces were generated by choosing appropriate iso-values corresponding to each classification.

resolution approach [72].

Unfortunately, the assumption made by iso-surfacing, that the boundary of an object can be defined by a constant value, does not always hold. Noise and (in MRI) bias fields [75] are factors to take into account when constructing an iso-surface (also see Figure 2.6). Iso-surfacing of noisy data can result in surfaces consisting of a large number of triangles and spurious surfaces. Figure 2.6 shows an example of such a surface that is extracted from a MRI scan. Two approaches can be followed to remedy this problem. The first is to prefilter the dataset using a lowpass smoothing filter (e.g., a Gaussian kernel). The second is to post-process the generated surface, for example by removing small clusters and to apply mesh smoothing techniques.

Segmentation commonly results in a binary image (i.e., classification at pixel or voxel level). Extracting a surface from these binary data results in a triangular surface model that does not meet all of the requirements above. The smoothness of the mesh can be poor due to quantisation effects, showing ridges or terraces. Gaussian pre-filtering of the binary image (before surface extraction) reduces accuracy, and significant anatomical detail (such as narrow ridges and clefts) may be lost, and still sufficient smoothness is may not be achieved [61].

2.1.3 Edge detection techniques

An iso-surface approach defines the boundary of an object by a specific data value. As was shown in the previous Section this assumption does not hold for certain types of

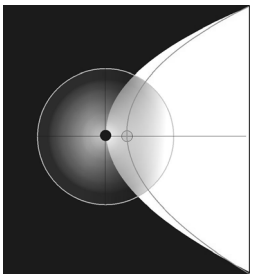


Figure 2.7: Dislocation of edge by a low-pass filter with tangential components.

data, for example when noise is present, or when the boundary of an object cannot be represented by a constant data value. In contrast, edge-detection methods find the boundary of an object by looking for transitions in data values. By doing so, absolute data values are not used for boundary identification. Unfortunately, there is no clear definition of what exactly constitutes an edge in an image. Commonly, an edge is defined in terms of image intensity and accordingly, edge detection is defined here as the identification and localisation of strong intensity changes between a region of interest and its surroundings. Intensity changes in the data are quantified by applying the gradient operator ∇ to the image I. Large transitions in image intensity correspond to large gradients, and consequently, edges are identified by local maxima of the absolute value of the gradient vector. An alternative approach is to localise edges by calculating zero-crossings of the Laplacian operator applied to the data $\nabla^2 I$. Examples of such filters can be found in any image processing handbook (e.g., [33]). An overview of basic edge-detection techniques can be found in [144].

Unfortunately, the correct identification of edges is also hampered by image noise due to the sensitivity of the gradient operator to higher frequencies. A common approach is to apply a Gaussian derivative operator to suppress noise. However, suppressing noise also dislocates edges. The location of a curved edge is affected by a low-pass filter with tangential components [32, 115, 129], which causes a bias in the detection (see Figure 2.7). Consequently, there is a trade-off between noise suppression and localisation in simultaneous filtering and detection [76]. Systematic errors influence the accuracy of the edge detector, while noise influences the reproducibility. Although accuracy and reproducibility are both important, we cannot satisfy both to the same extent simultaneously. In this work the focus is on reproducibility.

A balance between between noise suppressing, detecting and locating edges is achieved by using a scale-space approach [6, 49, 76]. Here, a stack of images is created where each consecutive image is a lower resolution, filtered version of the previous image. This is illustrated in Figure 2.8 on the following page where each image is created by filtering using a Gaussian kernel with an increasing value of σ (see Figure 2.8).

Figure 2.8: Scale-space. From left to right the images are filtered with a Gaussian kernel using $\sigma = 50, 20$, and 1 voxels, respectively. The image is taken from a μ-CT dataset of a human carpal bone.

Starting the edge detection at the top of the stack, with the image that is created using the largest σ, ensures good edge *detection*. Tracing the edge down through the stack of images towards smaller values of σ assures good edge *localisation*. By doing so, dislocation and misdetection can be prevented.

2.1.4 Contour connecting methods

Contour connecting is one of the earliest approaches to reconstrucing a surface from 3D data. The method is able to capture objects of complex topology and geometry. Contour connecting views a 3D dataset as a stack of parallel 2D images (see Figure 2.9 on the next page). After outlining a desired contour in each image of the stack, a 3D surface is created by connecting all adjacent contours by polygons, see e.g., [9, 20, 44, 55, 89]. Contour outlining can be performed either manually, or by any of the many available segmentation techniques [98].

Generally, generating a surface from a set of slices containing closed contours that do not self-intersect can be split up into the following set of problems [89]:

- The *correspondence problem*. If a dataset contains multiple objects then multiple contours can be present in a slice. Each contour must be attributed to a single object before contour connecting can be performed.

- The *tiling problem*. This problem relates to the topological relationships between slices. Points on pairs of contours need to be mapped by creating a set of triangles. The set of triangles can be constructed according to several metrics, depending on the problem. A common metric is minimisation of the resulting triangle surface area [55]. A comparison of several metrics can be found in [9].

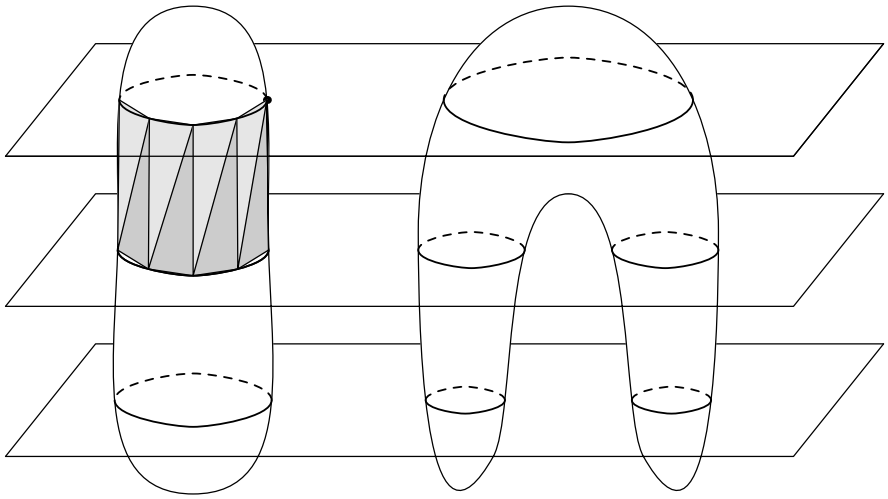


Figure 2.9: The contour connecting problems. A stack of three slices of two shapes is shown. Attributing each contour to a single shape is the correspondence problem. On the right the branching problem is illustrated by the forking of the shape. Finally, on the left the tiling problem is shown by the grey triangles that connect adjacent contours.

- The *branching problem*. Usually, a simple object will have one closed contour in each slice. However, if an object branches (e.g., a molar, or a human vertebra) then the number of contours in adjacent slices can differ. Mapping the correspondence between the contours is the branching problem. In [44] a method is outlined to solve the branching problem which links contours based on the area of the overlap.

- The *surface fitting problem*. This is a more general problem of generating a smooth surface through a set of connected contours. The spacing between the slices can be large, and the approximation of the surface by a single row of triangles can be too coarse. Creating a smooth surface is closely related to the topics of mesh generation, smoothing and parametric surface interpolation [46].

Figure 2.9 illustrates the correspondence, tiling and branching problems. Additionally, there is the problem of the disappearing contours. In contour connecting a main axis through the data is defined along which the slices are oriented. Slices that capture the extrema of an object wil contain very small contours and it may be difficult to capture an object precisely. In Chapter 5 a method is outlined that circumvents this problem, as well as all the other contour connecting problems mentioned in this list.

2.1.5 Deformable models

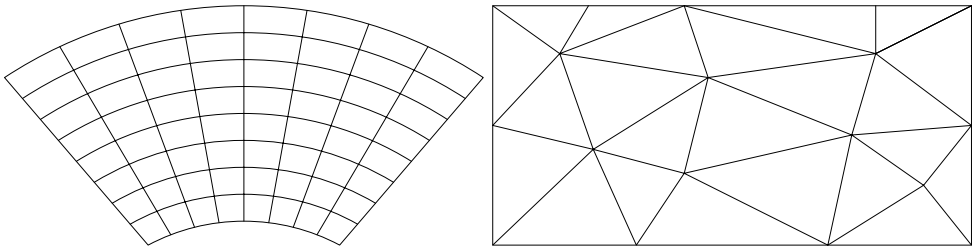
A deformable model is a generic approach to surface extraction that integrates segmentation and surface reconstruction. Instead of the 2D→3D approach of contour connecting (Section 2.1.4) the surface is reconstructed directly in 3D. Deformable models are contours or surfaces that can be adapted to approximate a certain desired shape. Fitting a desired object boundary is modeled by a process in which an initial surface is shrunk, grown, or deformed by minimising an energy function E with internal and external energy components. The internal energy term specifies properties of the surface during deformation, for example elasticity and stiffness. The external energy component couples the deformable surface to the data, and drives the surface to the desired boundary, e.g., by exploiting gradient information (see the previous Section) [41, 71, 86, 93, 101]. Minimising the total energy E drives the deformable model to the object boundary.

The precise form of the internal and external energy terms varies from study to study [66, 67, 80, 87, 88, 93, 101, 115, 141]. The extent to which the energy terms control the positions on the surface is controlled by (a possibly large number of) weight factors. For example, trade-offs can be made between boundary detection, noise suppression, and surface properties (e.g., curvature). Generally, the values of the weight factors are determined by a process of trial and error [66] – the exact tuning often lacks a physical basis. Therefore, reproducibility is limited due to the manual tuning of possibly many weight factors in the energy function. Furthermore, the vertices of the produced mesh may be spaced irregularly by the relocation process, which can produce distorted elements.

In this thesis the methods described in Chapters 3 and 4 are related to deformable models, where the number and tuning of weight factors are kept to a minimum.

2.2 Surface mesh improvement

A large amount of research is and has been performed on creating, improving, refining, and coarsening of surface meshes and grids, see e.g., [18, 53, 57, 125]. The motivation for this research is the trade-off between speed – coarse meshes – and accuracy – detailed meshes. Two driving applications behind this research are the finite element method (FEM) [70] and visualisation. FEM is a generic solving tool that subdivides a complicated problem into a set of connected simpler elements. The complete solution is found by solving each subproblem and compounding the results. Examples of large finite element simulations are flow simulations [1, 68], virtual vehicle crash tests [103, 135] and surgical simulation with soft tissue deformation [39, 74]. Mesh coarsening and refinement fit naturally in this application. For example, in a mechanical simulation the mesh can be made very fine in sections with high stress



(a) Structured grid (b) Unstructured grid

Figure 2.10: Regular and irregular grids

concentrations and coarse elsewhere. By doing so accuracy is maintained and calculation time is spent only when and where necessary. Visualisation requires similar compromises between model complexity and interactivity. Many methods exist for mesh coarsening, refinement and fast rendering [7, 58, 63, 64, 109, 111, 126].

2.2.1 Surface mesh definition

A surface mesh consists of a set of vertices, edges and faces. A vertex v_i is a location in 3D space. An edge connects two vertices; e_{ij} denotes an edge between between vertices v_i and v_j. Faces f_i are polygons (triangles, quadrilaterals, etc.) that can be indicated either by a set of edges, or a set of vertices. The latter combination is often chosen to save storage space. Depending on the specific application more elaborate data structures are available that are optimised for retrieval or traversal of the mesh [10, 30].

The neigbourhood or 1-ring neighbourhood of a vertex is formed by all vertices that are connected by an edge to that vertex. The surface can be open (sheet) or closed (sphere, torus). In a manifold surface, around every point there is a neighbourhood which is locally "flat".

Meshes are either structured (Figure 2.10(a)) or unstructured (Figure 2.10(b)). In a structured mesh the vertices are ordered in a regular fashion where each internal point (i, j, k) has $(i \pm 1, j \pm 1, k \pm 1)$ as its neighbours. In an unstructured mesh the vertices are connected by polygons and the neighbour property does not hold. Generally, unstructured meshes require a smaller number of elements to describe a complex geometry than structured meshes because the placement and size of elements is more flexible. However, unstructured meshes usually require more storage space.

Mesh quality and quality criteria are important both for FEM and visualisation and are covered extensively in Section 2.2.2. Usually, mesh extraction methods do not

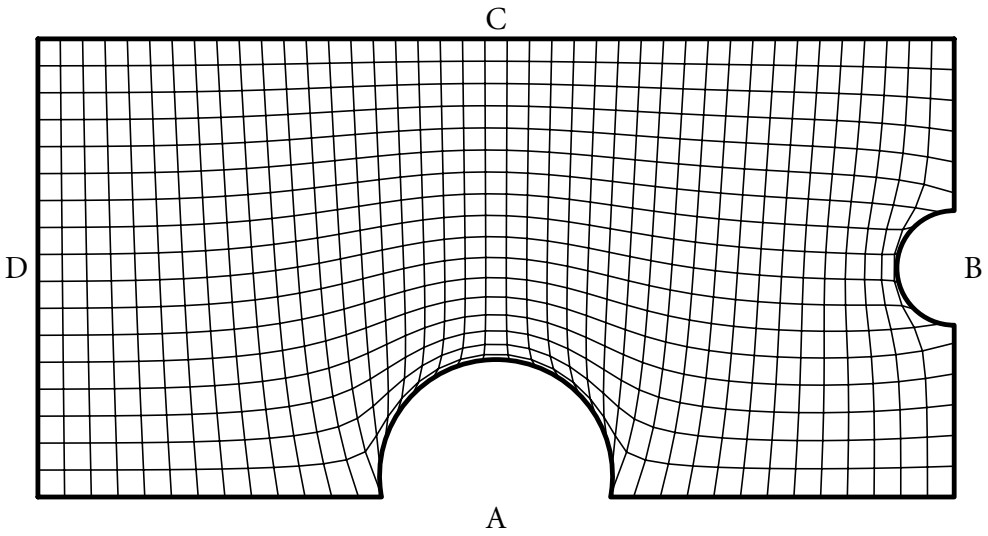


Figure 2.11: Example of a curvilinear Laplacian grid. The locations of the interior points are calculated from the four boundary curves A, B, C, and D.

take triangle quality into account. For example, the meshes created by the Marching Cubes algorithm [81] can be of very poor triangle quality. If further processing steps have requirements on the quality of a mesh then it is often necessary to improve the quality of such extracted meshes. Algorithms for mesh improvement fall into three categories [52]:

- Relocation of vertices;
- Local reconnection of mesh elements;
- Mesh coarsening and refining [57, 64].

The second and third category change the topology of the mesh, the first category leaves topology intact.

The simplest approach to vertex relocation is Laplacian smoothing and its variants and derivatives. These techniques are popular chiefly because of their computational simplicity, local operation, and global effect. From Section 2.2.3 onwards several variants are discussed. An example of local reconnection of mesh elements is edge swapping. Some standard edge swapping techniques are shown in Section 2.2.6. In this work a very simple mesh refinement technique is applied (see Chapter 4). Good overviews of mesh refinement techniques can be found in [57, 111].

2.2.2 Surface mesh quality

The quality of a surface is related to the number and shape of the triangles that form the surface. Several metrics can be found in the literature for quantification of the (improvement of) quality of a surface mesh [8, 12, 102]. Unfortunately, there is no generally agreed upon definition of mesh quality; the choice of quality metric depends on the subsequent application, e.g., visualisation and finite element modelling methods. In these applications good quality triangles and a smooth surface are preferred (visualisation) and even required (FEM) most of the time. Numerical solving methods put requirements on the shape of triangles to avoid inaccuracies and stress concentrations. For specific cases quality metrics can be derived based on the approximation function and a known target [12]. Error analyses indicate that finite element models and most triangle-based surface representation schemes are most accurate when the triangles are nearly equilateral. Angles close to 180° are to be avoided. Making triangles more equilateral also works automatically toward elimination of obtuse triangles from the mesh. A triangulation with only acute triangles is important in unstructured finite difference approximations to partial differential equations [54].

A common and straightforward approach to determine mesh quality is visual inspection [11]. Figure 2.12 on the next page shows an example of two meshes that are extracted from a CT dataset of a human ankle. The left mesh is apparently of lower quality than the right mesh.

One approach to quantifying triangle quality is to measure the deviation from an equilateral triangle. Examples of such metrics are the ratio of shortest edge and longest edge, and the ratio of the smallest and largest angle. Both metrics yield values between 0 (bad) and 1 (good, equilateral). The quality of a mesh can now be represented by a histogram of the quality of all triangles [51]. Figure 2.13 on the following page shows such a histogram created for the meshes shown in Figure 2.12 on the next page. The histogram confirms the results of visual inspection: the right mesh has less low quality triangles and contains more high quality triangles.

Other quality metrics concern the distribution of the minimum and maximum angles [51], or take the surface area of a triangle into account [8]. An overview of these triangle quality metrics can be found in Chapter 4.

2.2.3 Laplacian smoothing

Smoothing is a name often given to two distinct processes. First, smoothing refers to rearranging the vertices of a mesh to improve the quality without altering the shape. Second, the term smoothing describes the process of removing unwanted detail, e.g., removing noise from a surface. Both effects can be achieved by applying Laplacian smoothing or one of its derivatives.

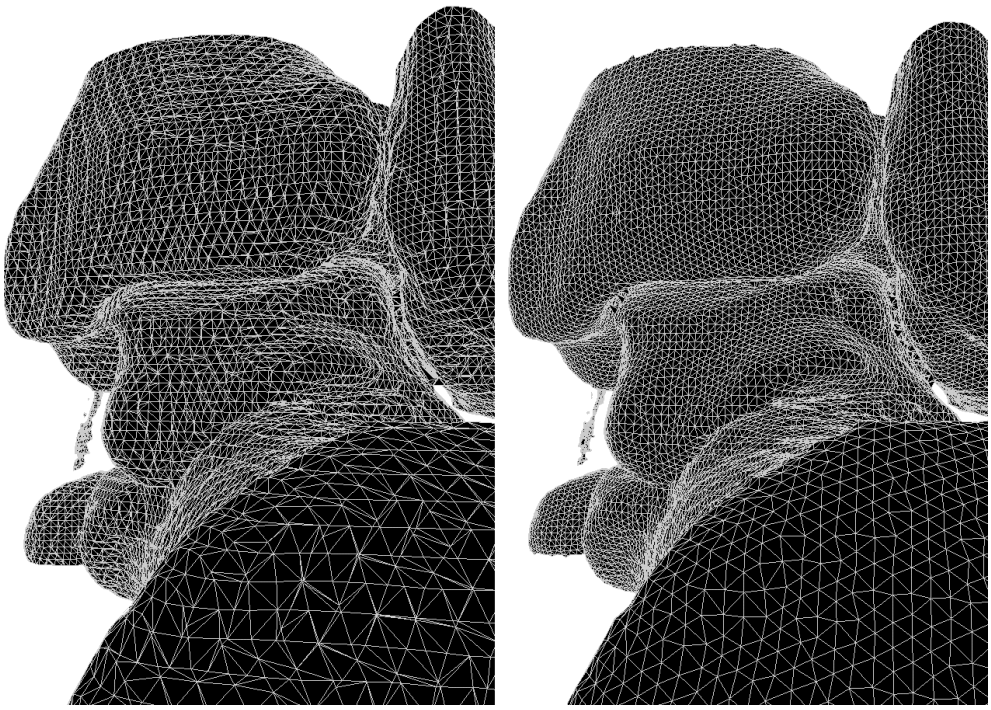


Figure 2.12: Visually comparing mesh quality. A mesh extracted from a CT volume of a human ankle dataset is shown. The initial mesh is on the left (created by the Marching Cubes algorithm), an improved version is on the right.

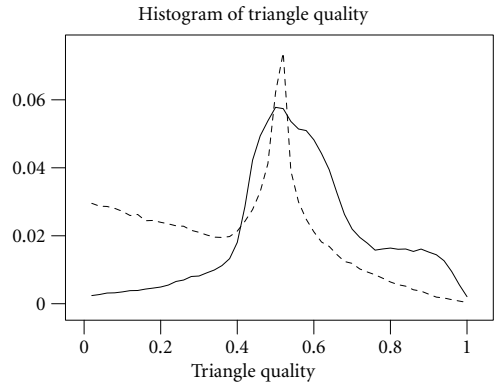


Figure 2.13: Histogram comparing the triangle quality of the meshes shown in Figure 2.12 (dashed – Marching Cubes mesh, solid – improved mesh). The quality criterion is the smallest angle divided by the largest angle of a triangle (see Section 4.2.3). The criterion yields values between 0 (bad) and 1 (good, equilateral triangle).

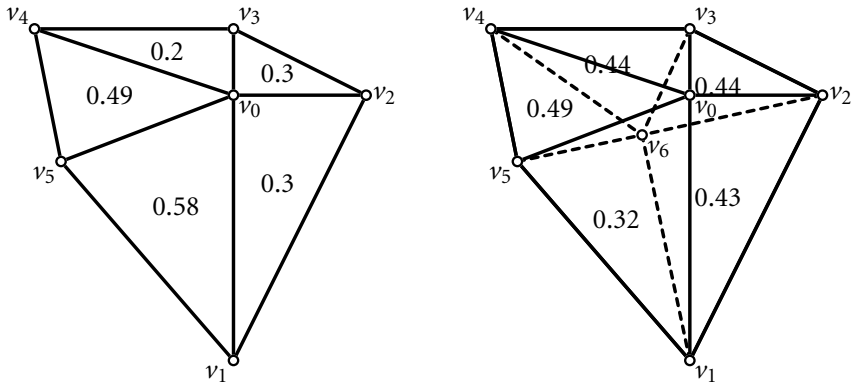


Figure 2.14: Laplacian smoothing of an initial polygon (left), in the right image the center vertex v_0 is moved to the average location of the surrounding vertices v_6 (right). The quality of each triangle is indicated. The average quality increases from 0.37 to 0.42.

Laplacian smoothing is part of the class of mesh improvement methods that operate by relocating the vertices of a surface. In its simplest form Laplacian smoothing moves each vertex of a mesh to the average position of its linked neighbours.

The origin of Laplacian mesh smoothing can be traced back to the grid generation literature of the late 1960s [1] [29, 35, 47, 79]. The idea was prompted by the visual analogy between the solutions to the partial differential equations that describe electrostatic potential and potential flow problems and the generation of smooth structured grid lines [125, page xiii] (see Figure 2.11 on page 18 for an example of such a grid). Particularly of interest is the partial differential equation $\nabla^2\psi = 0$ — Laplace's equation [137, 138]. Any function ψ that satisfies this equation is called harmonic. The superposition of two solutions is also a solution because the equation is linear. It can be shown from the calculus of variations that grids generated from the Laplace equation have the smoothest possible spacing of grid lines [125, page xvii].

The equipotential method [29] regards the lines of a structured grid as two intersecting sets of equipotentials, with each set satisfying Laplace's equation. Laplace's equation can be solved if the boundary conditions are given; all internal nodes can then be calculated by satisfying the differential equation over the regular grid. The solving procedure is outlined here for the 2D case (following [29]), generalisation to higher dimensions is performed similarly. The second derivative of ψ is approximated

[1] The first generator for smooth triangle meshes was implemented by R. Maclean in 1958 for the IBM 704 [138]. In 1959 Leith developed a triangle zoner for the IBM 704 which placed each interior mesh point at the average position of its nearest neighbours by an iterative procedure [137].

by the central differences

$$\frac{\partial^2 \psi}{\partial \xi^2} = \frac{1}{h^2}\left(\psi(\xi+h,\eta) - 2\psi(\xi,\eta) + \psi(\xi-h,\eta)\right) \tag{2.1}$$

$$\frac{\partial^2 \psi}{\partial \eta^2} = \frac{1}{h^2}\left(\psi(\xi,\eta+h) - 2\psi(\xi,\eta) + \psi(\xi,\eta-h)\right) \tag{2.2}$$

where (ξ,η) is the coordinate system and h is the size of a step in either the ξ- or η-direction. Discretisation of these two equations is performed by setting $h = 1$, and substituting the grid coordinates (i, j) for (ξ,η), yielding the following two difference equations:

$$\begin{aligned} x_{i+1,j} + x_{i-1,j} + x_{i,j+1} + x_{i,j-1} - 4x_{i,j} &= 0 \\ y_{i+1,j} + y_{i-1,j} + y_{i,j+1} + y_{i,j-1} - 4y_{i,j} &= 0 \end{aligned} \tag{2.3}$$

where (x,y) is a node of the grid. Rewriting Equation 2.3 yields

$$\begin{aligned} x_{i,j} &= \frac{1}{4}\left(x_{i+1,j} + x_{i-1,j} + x_{i,j+1} + x_{i,j-1}\right) \\ y_{i,j} &= \frac{1}{4}\left(y_{i+1,j} + y_{i-1,j} + y_{i,j+1} + y_{i,j-1}\right) \end{aligned} \tag{2.4}$$

in which the averaging quality of Laplacian smoothing can be recognised readily. Given the locations of points on the boundary of the grid, Equation 2.4 can be used to calculate the locations of interior points. Figure 2.11 on page 18 is an example of a mesh that is generated by this method from four boundary curves. The end result is that each point lies at the average location of its linked neighbours (see Figure 2.14 on the preceding page).

Generalising Equation 2.4 for arbitrary connectivity – and thus unstructured grids or meshes – yields

$$\sum_i w_i (x_i - x) = 0, \qquad \sum_i w_i (y_i - y) = 0 \tag{2.5}$$

where (x_i, y_i) are the linked neighbours and w_i is a weighting function. The weights are chosen such that $\sum_i w_i = 1$. Several of the variants of Laplacian smoothing use this weighting function to achieve specific results (see Section 2.2.4).

Finally, the complete iterative solving scheme is given by

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \sum_i w_i^n (\boldsymbol{x}_i^n - \boldsymbol{x}^n) \tag{2.6}$$

where $\boldsymbol{x}$ denotes a vertex and n is the iteration number.

The following observations on the relation between Laplacian smoothing and Laplacian grid generation can be made:

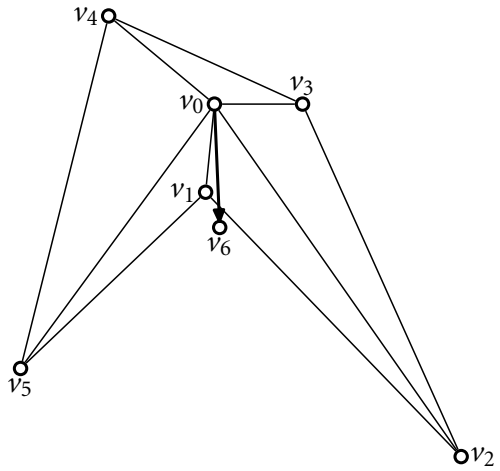


Figure 2.15: Concave 1-ring neighbourhood. Vertex 0 is moved to the average location of its linked neighbours, vertex 6. If the surrounding polygon is concave the center can be located outside and the two triangles $1-2-0$ and $1-0-5$ invert.

- Grid generation usually generates structured grids that have a fixed connectivity number, surface meshes have variable connectivity numbers.
- Boundary conditions are usually not specified for surface meshes. Repeated application of Laplacian smoothing will in these cases shrink the mesh.

Usually, the newly calculated coordinate in Equation 2.6 on the facing page is immediately used for all subsequent Laplacian smoothing of other vertices [47]. However, it is also possible to take a batch approach where each new vertex location is calculated and then the entire mesh is updated after all new locations are determined. In Section 4.5.1 the update mechanisms and their implications are examined in closer detail.

It is possible to generate geometrically invalid meshes using Laplacian smoothing. If the polygon defined by the 1-ring neighbourhood of a vertex is concave, then the application of Equation 2.6 on the preceding page *can* move the vertex outside the polygon. The result is that triangles are inverted and the mesh is folded back onto itself. Figure 2.15 illustrates this problem. If the surrounding polygon is convex then it is by definition impossible for the central vertex to move outside by the averaging operation. This problem is explored further in Section 4.5.

To conclude this Section a list is given of some of the properties of Laplacian smoothing:

- Improves the mesh quality by making triangles closer to equilateral. Also, the averaging action smoothes the mesh surface.

- Computationally efficient and easy to implement. Only the directly connected vertices are needed to move a vertex and averaging is a simple operation. However, convergence may require many iterations.

- Vertices are not always moved to the optimal position to get the best element quality for each individual triangle. This can be seen e.g., in Figure 2.14 on page 21 where, although the average quality increases after a smoothing step, one of the triangles drops in quality from 0.58 to 0.32.

- Inverted elements can be generated if the center vertex is moved outside the polygon defined by the 1-ring neighbourhood.

- Uniform triangle size tends to get lost. For example, Figure 2.11 on page 18 shows that the mesh has smaller elements in "complex" regions and larger element in "simpler" regions.

- Mesh shrinkage due to missing boundary conditions. Consider a triangulated sphere; moving a vertex to the average location of its neighbours will always result in a position closer to the center of the sphere. After repeated application of smoothing, the sphere noticeably shrinks and will, after a large number of smoothing steps, collapse into a single point.

2.2.4 Laplacian smoothing variants

The properties of standard Laplacian smoothing have inspired several variants. The boldest approaches ignore mesh connectivity and perform the smoothing based on the n-closest neighbours [29, 143]. However, such methods are computationally intensive because of the exhaustive lookups in unstructured meshes. Another generic adaptation is to move a vertex only if the quality of the triangles improves [31, 51, 52].

Alternate weighting schemes

Most variants are based on some form of weighting the contributions of a vertex and its linked neighbours [142] (cf. Equation 2.6 on page 22). Setting $w_i = \frac{1}{N}$, where N is the number of neighbours yields standard Laplacian smoothing function. Other weightings that can be found in the literature are:

- Edge-length weighted [15, 56, 120]: $w_i = \dfrac{1}{|v_i - v_j|}$

  The contribution of each vertex is weighted inversely proportional to its distance to the center vertex. By doing so, only the direction of the edge vectors

is important. It is noted in [120] that this method reduces the amount of geometric errors. It can be deduced that by reducing the importance of the length of edges the center vertex travels a shorter distance and is therefore less likely to move outside a concave surrounding polygon.

- Area weighted [22]: $w_i = \text{area}_i$

  The idea behind area weighting is to promote equal area triangles. Different from normal Laplacian smoothing this method does not use the edge vectors but the vectors from the central vertex to the centroid of each triangle. Smoothing using the centroids can also be found in [94].

- Desbrun weights [42, 121]: $w_i = \cot\alpha_{ij} + \cot\beta_{ij}$, where α_{ij} and β_{ij} are the two angles opposite to edge e_{ij} in the two triangles having e_{ij} in common. The total movement of a vertex consists of a normal and a tangential component to the surface. In some cases the tangential component is unwanted, for example when the surface is texture-mapped. These weights eliminate the tangential component. The resulting smoothing affects the shape, but not directly the quality of the triangles.

- Angle weighted [142]:

  Another smoothing approach is based on the angles of the 1-ring neigbourhood [142]. A line is constructed through each linked vertex such that the line bisects the angle of the 1-ring neighbourhood at that vertex (see Figure 2.16 on the following page). Next, the center vertex is projected perpendicularly onto each of the bisectors. A new location for the center vertex is then calculated by averaging the calculated projected points. Although more computational effort is required than common Laplacian smoothing, better results are achieved in terms of final mesh quality [142].

Approximate volume preservation

A drawback of Laplacian smoothing is that it will shrink the mesh. One way to counteract this effect is described in [22]. The method works by approximating the volume difference before and after moving a vertex. Then, the location of the vertex is modified using this difference to preserve the volume. The volume V of a cone spanned by the surrounding polygon with center vertex v can be calculated by

$$V = \sum_f \boldsymbol{v} \cdot \boldsymbol{N}_f \tag{2.7}$$

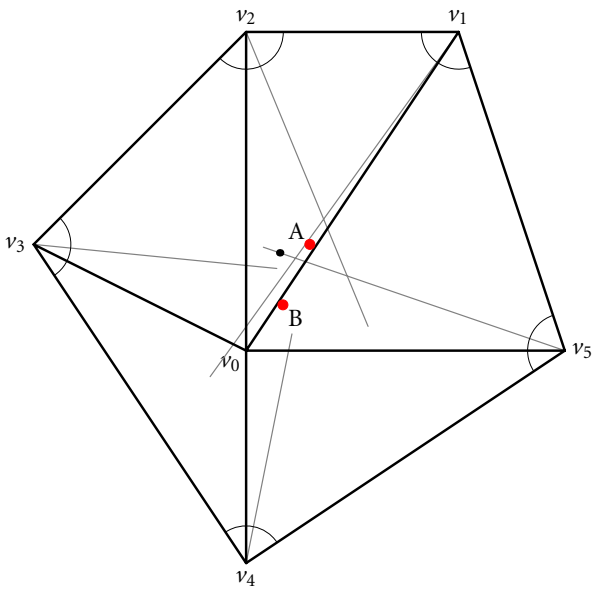


Figure 2.16: The grey lines bisect the angles of the surrounding polygon at each of the linked vertices v_1, v_2, v_3, v_4, v_5. The center vertex v_0 is projected perpendicularly onto each of the bisectors (indicated by the black dots). Point B is the average location of these projections. For comparison the average of the linked vertices is indicated by point A.

where N_f is the (not normalised) normal of the incident triangle f. The vertex normal N is approximated by the average of the triangle normals. Next, the new position v_{avg} for the center vertex is calculated. Volume preservation is obtained by adjusting the average position along N by a factor λ such that the volume after smoothing equals the initial volume

$$\left(v_{\text{avg}} - \lambda N\right) \cdot N = v \cdot N \quad . \tag{2.8}$$

The multiplication factor λ can then be calculated by

$$\lambda = \frac{v_{\text{avg}} \cdot N - v \cdot N}{N \cdot N} \quad , \tag{2.9}$$

and the corrected vertex location v_{new} is calculated as

$$v_{\text{new}} = v_{\text{avg}} - \lambda N \quad . \tag{2.10}$$

Signal processing approach to smoothing (Taubin smoothing)

Another view on mesh smoothing can be found in [42, 120, 121] where the repeated application of Laplacian smoothing is considered as a signal processing problem. The relocation of a vertex x_i under Laplacian smoothing can be written as

$$\Delta x_i = \sum_j w_{ij} \left(x_j - x_i\right) \quad j \neq i \tag{2.11}$$

where j is an index in the 1-ring neighbourhood of x_i. Define the matrix $W = w_{ij}$, where the elements are zero when vertices are not neighbours. Define the matrix $K = I - W$, where I is the identity matrix. Equation 2.11 can now be rewritten as

$$\Delta x_i = -Kx \quad . \tag{2.12}$$

The connectivity of the mesh is encoded in the matrix K. The new position after one smoothing step can be calculated by

$$x_{\text{new}} = x + \lambda \Delta x = (I - \lambda K)\, x = f(K)x \tag{2.13}$$

where λ is a scaling factor between 0 and 1 that controls the speed of the process. Equivalent to linear filter theory [99] the function $f(k)$ is the transfer function of the mesh smoothing filter. First, consider n consecutive steps of common Laplacian smoothing; the equivalent transfer function is the polynomial given by $f(k) = (1 - \lambda k)^n$. Stable behaviour, i.e., $(1 - \lambda k)^n \to 0$ for $n \to \infty$, is obtained only for $k \in (0, 2]$. All frequency components are attenuated for large n except for the zero

frequency, which corresponds to the center of gravity of the mesh. This provides another explanation for the shrinking of the mesh using Laplacian smoothing. Now, the entire toolbox of linear filter design is at our disposal to create filters to smooth a mesh. In [121] a set of filters is created for low-pass filtering of a mesh.

One example of filter design for smoothing is the $\lambda|\mu$ algorithm [120, 121]. Based on the filter theory above a low-pass filter is designed that prevents the shrinking of the mesh. The following second-degree transfer function is used

$$f(k) = (1 - \lambda k)(1 - \mu k) \tag{2.14}$$

where $\mu < -\lambda < 0$. In terms of Laplacian smoothing this transfer function corresponds to performing two steps: first a step with the positive scaling factor λ, followed by a step with the negative scaling factor μ. The shrinking in the first step is compensated by an opposite movement of a vertex in the second step. A thorough overview of this type of filter design can be found in [122].

Note that stable convergence of the filter does not automatically imply that the surface remains free of artifacts. As mentioned before, if the surrounding polygon of a vertex is concave then there is always the possibility of moving the vertex outside and inverted triangles are created.

Diffusion and curvature flow

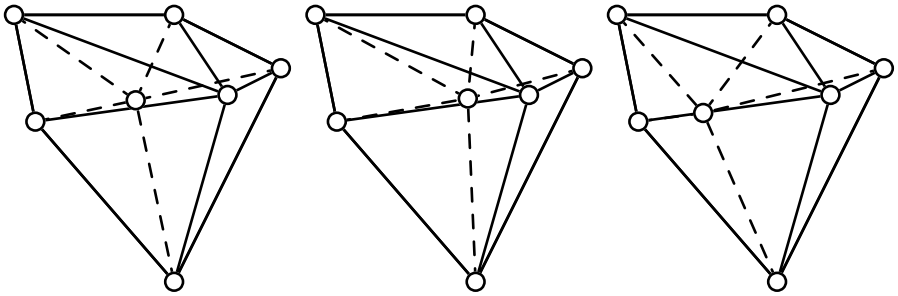
Up until now iteration is used to move the vertices of the mesh. The alternative is to solve a set of partial differential equations. This approach is taken by [42], where a method is described to smooth a mesh by using the sparse nature of the involved matrices and by using an implicit integration scheme.

2.2.5 Visual overview

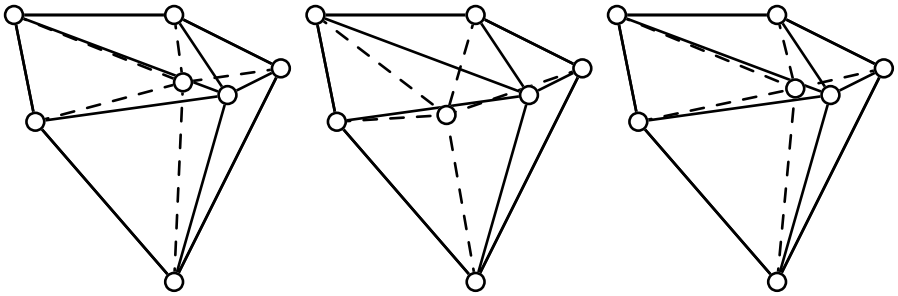
Finally, at the end of this Section on smoothing techniques a visual overview of the effect of all weighting schemes is shown. Figure 2.17 on the facing page illustrates the effect of each method. Note that the Desbrun weighting does not move the center vertex because the triangles are co-planar.

2.2.6 Edge swapping techniques

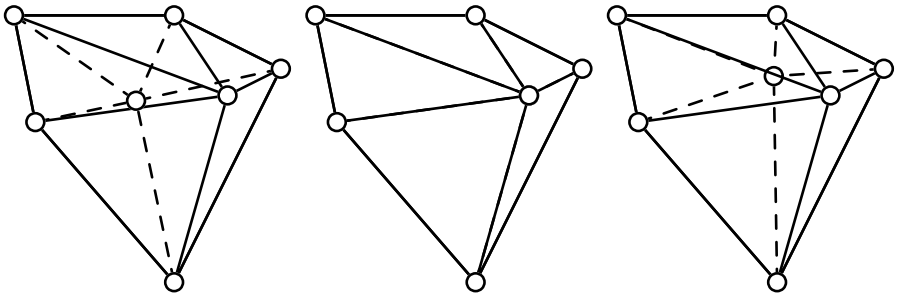
Edge swapping is a topology altering mesh improvement technique. The concept is shown in Figure 2.18 on page 30. Two triangles that share an edge can be considered as a quadrilateral. The edge is swapped if the mesh improves. Several criteria can be used to determine if an edge swap improves the mesh. Straightforward techniques are the shortest edge and the Delaunay criteria. The shortest edge criterion swaps an edge



(a) Laplace (b) Laplace (centroids) (c) Length weighted

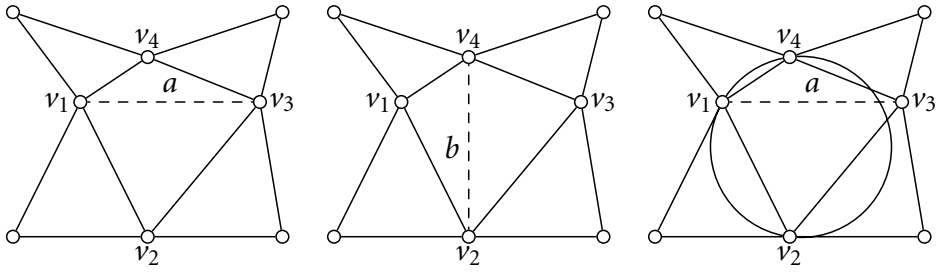


(d) Inverse length weighted (e) Area weighted (centroids) (f) Inverse area weighted (centroids)



(g) Taubin (h) Desbrun (i) Angle criterion

Figure 2.17: A visual overview of the smoothing approaches.



(a) Before swap (b) After swap (c) Delaunay criterion

Figure 2.18: Edge swapping. The Delaunay criterion swaps edge a if vertex v_3 is located outside the circle through the three other vertices.

if the resulting edge after swapping is shorter. The Delaunay criterion is illustrated in Figure 2.18(c). Here, a circle is fitted through three of the four vertices that form the quadrilateral. If the fourth vertex is inside the circle then the edge is swapped.

One of the targets of mesh improvement is to create triangles that are as equilateral as possible. The degree d or connectivity number of vertex v in a mesh is the number of incident edges to that vertex. The only way to fill a flat surface with equilateral triangles is to have $d = 6$ at all vertices. In that case, each of the incident angles around a vertex can be equal to 60 degrees. An edge swapping approach that is based on the connectivity of a vertex is described in [54]. The method defines a relaxation index E by

$$E = (d_1 - 6) + (d_2 - 6) + (d_3 - 6) + (d_4 - 6) \tag{2.15}$$

where the subscripts correspond to the four corners of the quadrilateral formed by the vertices in Figure 2.18. An edge is swapped if the relaxation index after swapping is lower than before swapping. By doing so, edge swapping promotes the creation of vertices with $d = 6$ and consequently, potentially equilateral triangles.

A problem with the previous definition of the relaxation index is that edges can start to oscillate between two states. To prevent oscillation the relation index is redefined [18] as

$$E_b = (d_1 - 6)^2 + (d_2 - 6)^2 + (d_3 - 6)^2 + (d_4 - 6)^2 \quad . \tag{2.16}$$

If the quadrilateral formed by two triangles is concave then extra precautions have to be taken to ensure that an edge swap does not result in an invalid mesh. Section 4.5 describes some of the techniques we have used to ensure mesh validity.

2.3 Combined techniques

Methods from the fields of segmentation, and mesh generation and optimisation are rarely combined. Commonly, data is segmented, a surface is reconstructed and if the surface mesh is not acceptable, for example if it consists of too many polygons or is not smooth (enough), then mesh optimisation techniques are applied. Although mesh improvement methods try to keep surface properties, such as area or volume, constant, the connection between the surface and the data is lost. A combined approach that integrates both worlds has clear advantages, as is shown in Chapters 3 and 4.

One example of a combined approach is SurfaceNets [27, 60], a local mesh extraction method based on intensity values of the data (iso-surface extraction) that also incorporates mesh optimisation techniques. A mesh is generated at the voxel level. Subsequently, smoothing and surface detection are performed alternatingly. In this way, an accurate and smooth mesh is generated consisting of triangles of improved quality. The SurfaceNets technique requires few parameters. The dependence on an iso-value limits the field of application. Also, the mesh is created at the voxel level and contains a large amount of triangles, which impedes post-processing and further analysis.

A locally deformable surface model is proposed by Snel [115]. Here, mesh optimisation techniques are executed during the minimisation of the energy function E. Triangulations are regularized by vertex averaging and the surface is resampled to obtain triangles with an approximate prescribed area. In this way, a smooth mesh is generated consisting of high quality triangles. Accuracy is maintained by scale-space relaxation perpendicular to the mesh surface. However, reproducibility is limited due to the manual tuning of weight factors in the energy function. Furthermore, the two different smoothing operators are applied simultaneously, with the risk that the methods are cancelling each other out.

2.4 Roadmap

In this Chapter an overview is given of methods that form the basis or are related to the methods that are developed in the following Chapters. In the literature a large amount of work can be found on surface extraction and mesh improvement. However, techniques in which the two are integrated are not often found. Surface extraction methods create accurate surfaces, but do not take the triangle quality into account. On the other hand, mesh improvement techniques improve the quality of a surface but it is entirely possible that accuracy is lost. In the next two Chapters methods are developed that combine surface extraction and mesh improvement such that

both goals are achieved. In Chapter 3 a method based on iso-surface extraction is presented, called SurfaceNets. The ideas that are developed in this Chapter are taken a step further in Chapter 4 where an edge-detection based method is developed. The creation of an initial mesh for an iso-surface based method can be done by a simple thresholding. However, this is not possible for edge detection. A method to create such an initial mesh is presented in Chapter 5.

Improving triangle mesh quality with SurfaceNets

3

de Bruin, P.W., Vos, F.M., Post, F.H., Frisken-Gibson, S.F., and Vossepoel, A.M., Improving triangle mesh quality with SurfaceNets, *in* S. Delp, A. DiGioia, and B. Jaramaz, eds., *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, Oct. 2000 pp. 804–813, Third International Conference, Pittsburgh, PA, USA

Abstract

Simulation of soft tissue deformation is a critical part of surgical simulation. An important method for this is finite element (FE) analysis. Models for FE analysis are typically derived by extraction of triangular surface meshes from CT or MRI image data. These meshes must fulfill requirements of accuracy, smoothness, compactness, and triangle quality. In this paper we propose new techniques for improving mesh triangle quality, based on the SurfaceNets method. Our results show that the created meshes are smooth and accurate, have good triangle quality, and fine detail is retained.

3.1 Introduction

In recent years, endoscopic surgery has become well established practice in performing minimally-invasive surgical procedures. In training, planning, and performing procedures, pre-operative imaging such as MRI or CT can be used to provide an enhanced view of the restricted surgical field. Simulation of intra-operative tissue deformation can also be used to increase the information provided by imaging. However, accurate simulation requires patient-specific modeling of the mechanical behavior of soft tissue under the actual surgical conditions.

To derive an accurate and valid model for intra-operative simulation, we propose a five-stage process:

1. Image data acquisition (MRI, CT)
2. Image segmentation
3. Deformable tissue model generation
4. Intra-operative simulation of tissue deformation, guided by actual surgical conditions and/or intra-operative measurements conditions
5. Enhanced intra-operative visualization

In order to simulate tissue deformation, many authors have proposed finite element (FE) analysis of the relevant structures (see for example [24, 39, 74]). The FE models are commonly initialized by supervised segmentation of preoperative image data, resulting in a classification accurate to the pixel level. Using a surface extraction technique such as the Marching Cubes algorithm [81], the result is converted into a set of triangular meshes representing the surfaces of relevant organs. Such a representation can then be imported into an environment for FE analysis.

For optimal mechanical modelling and visualization, the triangular surface models should meet the following requirements :

- *Accuracy:* the representation of the organ surface geometry should be sufficiently accurate;
- *Smoothness:* the model should conform to the smooth organ boundaries. Sharp corners should be avoided as these can cause disturbing artifacts such as stress concentrations;
- *Compactness:* to achieve fast response times, the number of elements (triangles) in the model should be minimal; the resolution of the triangle mesh should be considerably lower than the medical image, with minimal loss of accuracy;

- *Triangle quality:* the shape of the triangles in the mesh should be as near as possible to equilateral to avoid FE errors and visualization artifacts.

Segmentation commonly results in a binary image (i.e., classification at pixel level). Extracting a surface from these binary data results in a triangulated surface model that does not meet all of the requirements above. The smoothness of the mesh can be poor due to quantization effects, showing ridges or terraces. Some solutions to this problem are inadequate. For example, Gaussian prefiltering of the binary image (before surface extraction) reduces accuracy, and significant anatomical detail (such as narrow ridges and clefts) may be lost, while insufficient smoothness is achieved [61].

In addition, the number of triangles generated by surface extraction may be very large. Compactness may be improved using mesh decimation techniques [58, 92], but these techniques are usually most effective with smooth meshes. Thus, smoothing of a surface mesh with minimal loss of accuracy is useful to avoid errors in FE analysis and for reducing mesh size. Exploiting the original greyscale data rather than binary segmented data can help to achieve this.

Recently, a technique called *SurfaceNets* was proposed to optimize a triangle mesh derived from binary data [60]. In this paper, the SurfaceNets method is extended to incorporate greyscale data. Several new techniques are examined and compared with Marching Cubes.

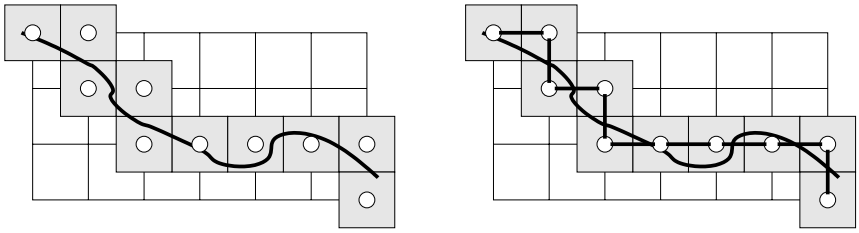
The paper is organized as follows. Section 3.2 briefly describes the basic SurfaceNets method, the extension to incorporate grey-scale data and new techniques for achieving smoothness, accuracy, and good triangle quality. In Section 3.3 these techniques are evaluated with respect to the requirements for mechanical modeling and visualization listed above. Finally, Section 3.4 summarizes our findings and draws conclusions.

3.2 Techniques

This section presents a brief explanation of the original SurfaceNet method (largely following [60]) which assumes that a binary segmentation of the original data exists. Then, two techniques will be introduced that utilize the greyscale image data during relaxation of the SurfaceNet.

3.2.1 Generating a SurfaceNet from binary data

The goal of the SurfaceNet approach is to create a globally smooth surface that retains the fine detail present in the original data. Generation of the surface net for binary data consists of the following four steps [60]:



(a) Before linking the nodes.

(b) After linking the nodes.

Figure 3.1: Building a SurfaceNet. The white squares represent voxels, the thick black line represents the edge of an object and the grey squares are cells with nodes represented by white circles in the center.

1. Identify nodes of the SurfaceNet;
2. Create links between the nodes;
3. Relax node positions while maintaining constraints on node movement;
4. Triangulate the SurfaceNet for visualization and FE analysis.

The first step in creating a SurfaceNet is to locate the cells that contain the surface. A cell is formed by 8 neighbouring voxel centers in the binary segmented data (Figure 3.1 presents the 2D case as illustration). If all eight voxels have the same binary value, then the cell is either entirely inside or entirely outside of the object. If, however, at least one of the voxels has a binary value that is different from its neighbours, then the cell is a surface cell. The net is initialized by placing a node at the center of each surface cell (step 1). Subsequently, links are created with nodes that lie in adjacent surface cells (step 2). Assuming only face connected neighbours, each node can have up to 6 links (corresponding to right, left, top, bottom, front and back neighbours). Once the SurfaceNet has been defined, each node is moved to achieve better smoothness and accuracy ("relaxation", step 3) subject to the constraint that each node must remain within its original cell. The relaxation process is described in more detail in the next section.

3.2.2 Improving smoothness

Once a SurfaceNet has been defined, the node positions are adjusted to improve the smoothness of the surface. This is often desirable to remove furrows and terraces due to the binary segmentation. Let us first only consider the smoothness of the net.

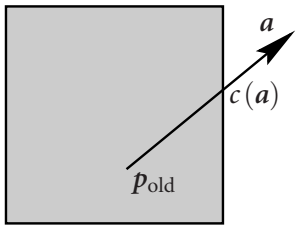


Figure 3.2: Position constraint of a node. If $\boldsymbol{p}_{\text{old}} + \boldsymbol{a}$ is outside the cell boundary, the function c is used such that $\boldsymbol{p}_{\text{old}} + c(\boldsymbol{a})$ is on the cell boundary.

One way to smooth the surface is to move every node to the average position of its linked neighbours [47]. The vector $\boldsymbol{a}$ pointing from the current position of the node $\boldsymbol{p}_{\text{old}}$ to the average position is calculated as:

$$\boldsymbol{a} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{p}_i - \boldsymbol{p}_{\text{old}} \tag{3.1}$$

where $\boldsymbol{p}_i$ corresponds to the position of a linked neighbour and N is the total number of neighbours of this node.

It may well be that the average position is outside the original cube and therefore diverges from the initial segmentation. To impose conformance, the relocation vector $\boldsymbol{a}$ is constrained to stay within the boundaries of the original cell by the function c (see Figure 3.2):

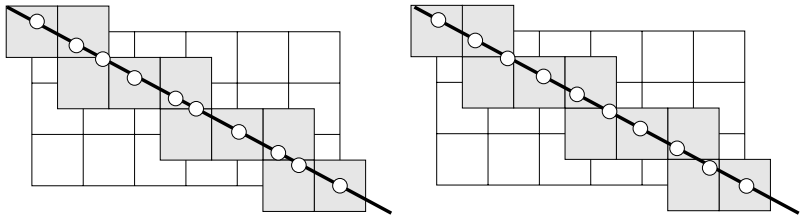
$$\boldsymbol{p}_{\text{new}} = \boldsymbol{p}_{\text{old}} + c(\boldsymbol{a}) \quad . \tag{3.2}$$

Here, c is defined to satisfy the proper constraint of the node position such that $\boldsymbol{p}_{\text{new}}$ is always within the boundaries of the cell. Note that this approach is different from the original SurfaceNet method which simply clips the new position's x, y, and z coordinates to cell boundaries when the new position falls outside the cell.

The relaxation is implemented in an iterative manner by considering each node in sequence and calculating a relocation vector for that node. The SurfaceNet is updated only after each node in the net has been visited. This procedure is repeated until the number of iterations has reached a preset threshold, or when the largest relocation distance is less than a given minimum value.

3.2.3 Increasing accuracy using greyscale data

The technique described above ignores all greyscale information in the dataset after building the SurfaceNet. The nodes shrink-wrap around the object without trying to conform to an iso-surface of the data. This is reasonable when the binary segmentation is the best estimate of the object. However, if the object surface can be estimated



(a) Using the gradient method (b) Combining the methods

Figure 3.3: Using the gradient method for relaxation the nodes (white circles) are projected onto the iso-surface (thick line, left). The combined relaxation technique also spaces out the nodes along the iso-surface (right).

to lie at an iso-surface of the image data, this iso-surface can be used to increase the accuracy of the SurfaceNet.

Let us assume that the true object surface can be obtained by drawing an iso-surface (at I_{iso}) in the original greyscale data. For instance, in many CT based applications the Marching Cubes algorithm is used to approximate the object shape in this way. By definition, at a given point the greyscale gradient vector is perpendicular to the iso-surface through that point. Thus, to enhance accuracy; a node can be displaced along the gradient vector to the iso-surface (see Figure 3.3(a)). This is expressed as:

$$\boldsymbol{g} = \text{SIGN}\left(I_{\text{iso}} - I\left(\boldsymbol{p}_{\text{old}}\right)\right) \nabla \boldsymbol{p}_{\text{old}} \quad . \tag{3.3}$$

Here, SIGN is a function that returns the sign of its argument, $I(\boldsymbol{p}_{\text{old}})$ is the interpolated intensity and $\nabla \boldsymbol{p}_{\text{old}}$ is the normalized gradient vector at $\boldsymbol{p}_{\text{old}}$. The latter vector is obtained either by a central difference gradient method or by convolution with Gaussian derivatives.

The node position is updated by:

$$\boldsymbol{p}_{\text{new}} = \boldsymbol{p}_{\text{old}} + c\left(d\boldsymbol{g}\right) \quad . \tag{3.4}$$

In this equation, d is a scaling parameter representing the distance to the iso-surface. The value of d can be estimated by assuming a linear image field near the iso-surface and interpolating the greyscale values at the node and at a point sampled along the vector $\boldsymbol{g}$. As in Equation 3.2 on the page before, c imposes a position constraint on the node to stay within the boundaries its cell.

3.2.4 A combined approach

Combining the methods presented in Section 3.2.2 and Section 3.2.3, we obtain a surface that fits the iso-surface of the data and is also globally smooth. To combine these features, a node should be displaced to obtain better smoothness within the iso-surface. The combination is made by first calculating the projection $\boldsymbol{a}_p$ of the averaging vector $\boldsymbol{a}$ on the plane perpendicular to the gradient $\boldsymbol{g}$ (cf. Equation 3.1 on page 37, Equation 3.3 on the preceding page):

$$\boldsymbol{a}_p = \boldsymbol{a} - \boldsymbol{g}\,(\boldsymbol{a}\cdot\boldsymbol{g}) \quad . \tag{3.5}$$

Subsequently, the combined displacement function is defined as:

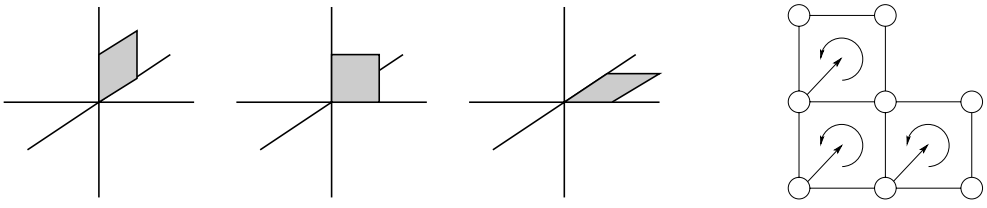
$$\boldsymbol{p}_{\text{new}} = \boldsymbol{p}_{\text{old}} + c\left(\boldsymbol{a}_p + d\boldsymbol{g}\right) \quad . \tag{3.6}$$

This formula combines relocation towards the iso-surface with smoothing in the orthogonal plane (i.e., on the surface). This can be seen in Figure 3.3 on the facing page where the nodes are first projected onto the line and then evenly spaced out along the line by the averaging. Again, c ensures that the new position of each node is always within the boundaries of its original surface cell. Note that there may be some tension between the goals of a smooth SurfaceNet and one that fits the iso-surface. One of these goals can be favored over the other either by weighting the independent contributions differently or by applying them sequentially rather than simultaneously; ending with the favored goal.

3.2.5 Triangulation

After relaxation the SurfaceNet is triangulated to form a 3D polygonal surface. We have simplified the original triangulation process described in [60]. Instead of directly building the triangles, first quadrilaterals are identified. There are three sets of four links of a node that lie in a plane (e.g., the left, right, top, and bottom links of a node lie in a plane) (see Figure 3.4 on the next page). In each plane the connected nodes form quadrilaterals and each node is a vertex of at most four quadrilaterals. In order to find all quadrilaterals it is sufficient to check in each plane one "corner" of a node. For example, in Figure 3.4(d) on the following page, all quadrilaterals are found by checking the upper right region of a node.

After relaxation each quadrilateral is triangulated using either a shortest diagonal or a Delaunay criterion [47]. Either of these criteria creates triangles that result in a smoother shape than choosing a fixed configuration. The resulting triangle mesh can be rendered using standard 3D graphics techniques.



(a) (b) (c) (d)

Figure 3.4: Three possible configurations for a quadrilateral. For each main direction only the top right corner has to be checked for a quadrilateral (right).

3.3 Results

To evaluate the relative effectiveness of the presented techniques, the SurfaceNet is compared to Marching Cubes, which is the standard iso-surface extraction tool [81]. The effectiveness of each technique will be tested against the requirements listed in Section 3.1. Each of these requirements is measured as follows.

- A measure expressing the local smoothness of a polygon mesh is given in [130]. As a first step, the angles α_i of all triangles around a vertex are summed. If all triangles connected to a vertex are coplanar this sum is equal to 2π. A measure of the local smoothness at a vertex is defined by $2\pi - \sum \alpha_i$, the absolute value of which is then averaged over all vertices.

- A simple and direct measure for triangle quality is found upon division of the smallest angle of each triangle by its largest angle. If the triangle is equilateral this expression is equal to 1.

- The accuracy is expressed by the unidirected modified Hausdorff distance that represents the mean distance of the generated mesh to a reference shape [43]:

$$H_{\text{ave}}(S_1, S_2) = 1/N \sum_{p \in S_1} e(p, S_2) \tag{3.7}$$

  where e is the minimum distance between a point and a surface, and S_1 and S_2 are two surfaces.

Using these measures, the following experiments are conducted. Two volumes, containing greyscale images of distance maps of respectively a plane and a sphere were created, where the greyscale values were stored as floats. An iso-surface is extracted using Marching Cubes (MC), a SurfaceNet with averaging (SNA) and a SurfaceNet

with the combined technique (SurfaceNet with Extended Relaxation and Triangulation SNERT) as presented in Section 3.2. These surfaces are compared to the exact reference shape. The results of this comparison are shown in Table 3.1 on the next page.

Comparing the quality of the triangles for each method shows that both SNA and SNERT produce triangles of a higher quality than MC for the plane as well as the sphere. Also, in the case of the plane the MC and SNERT method produce a smoother (=flatter) surface than SNA. The sphere has a constant curvature that corresponds to the smoothness outcome of MC and SNERT. SNA shrinks the mesh and pulls the nodes away from the iso-surface accounting for the lower smoothness and accuracy. The accuracy of the SNERT surface is lower than MC because the nodes are placed according to the trilinearly interpolated values. However, the error at the vertices for SNERT is smaller than the error at the face centers for the Marching Cubes generated sphere.

To illustrate the effectiveness of our technique a graphical example is shown in Figure 3.7 on page 44. Clearly, the SNERT surface is as flat as the MC surface and the triangles have higher quality. Figure 3.6 on page 44 shows the mesh generated by MC and SNERT on a dataset containing two overlapping spheres. The average triangle quality for the Marching Cubes mesh is 0.64, for the SNERT mesh this number is equal to 0.93.

In addition to the results presented, several experiments were done on true greyscale MRI and CT data. Figure 3.5 on page 43 shows a histogram of triangle quality for meshes generated by MC and SNERT from a greyscale CT dataset containing part of a human ankle. It can be seen that the SNERT mesh contains less low quality triangles and contains more high quality triangles. Figure 3.8 on page 45 shows the meshes generated from a CT-scan of a human ankle. Lastly, Figure 3.9 on page 45 shows a close-up of the bladder extracted from a 256x256x61 MRI dataset of the abdomen of a female patient.

3.4 Conclusions

Finite element analysis is a standard way to simulate soft tissue deformation. For proper modelling, triangular mesh models must satisfy requirements of accuracy, smoothness and triangle quality. Several approaches proposed in the literature do not meet these requirements (e.g., Marching Cubes in combination with low pass filtering).

In this paper we extended the SurfaceNet method, and evaluated two variants. Optimization of a triangle mesh was performed by averaging vertices, stepping in the direction of the gradient to the iso-surface, and a combined approach.

Table 3.1: Measured results on the Plane and the Sphere. Methods are Marching Cubes (MC), SurfaceNet Averaging only (SNA) and SurfaceNet with Extended Relaxation and Triangulation (SNERT). The accuracy is measured respectively at the vertices and at the centers of the faces.

	Quality		Smoothness		Accuracy (vertices)		Accuracy (face centers)	
	Plane	Sphere	Plane	Sphere	Plane	Sphere	Plane	Sphere
MC	0.64	0.54	$0.25\ 10^{-6}$	0.0028	$5.59\ 10^{-6}$	$2.92\ 10^{-3}$	$4.81\ 10^{-6}$	0.028
SNA	0.92	0.74	$16.2\ 10^{-3}$	0.0108	0.092	0.204	0.788	0.426
SNERT	0.93	0.75	$0.15\ 10^{-6}$	0.0028	$20.9\ 10^{-6}$	$12.3\ 10^{-3}$	$12.5\ 10^{-6}$	0.043

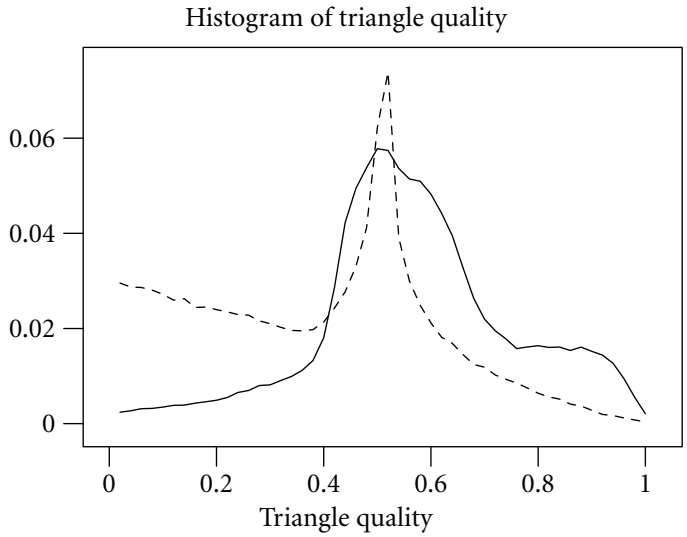


Figure 3.5: Histogram of triangle quality for meshes generated using Marching Cubes and a SurfaceNet. The mesh was generated from a CT scan of an ankle. The peak at 0.5 indicates that many right triangles are generated (quality measure is smallest angle divided by largest angle of a triangle).

From visual inspection of test objects, the meshes generated by a SurfaceNet appear to be of similar quality as those created by Marching Cubes. This is backed up by measurements. The SurfaceNet meshes are more suitable for finite element modelling as they are significantly smoother and have a low number of poor quality triangles.

We conclude that SurfaceNet creates a globally smooth surface description that retains fine detail.

Future research will focus on improving the performance of the SurfaceNets technique and developing suitable mesh reduction techniques for finite element analysis.

Acknowledgements

This research is part of the MISIT (Minimally Invasive Surgery and Intervention Techniques) programme of the Delft Interfaculty Research Center on Medical Engineering (DIOC-9). The work described here was largely carried out at MERL – a Mitsubishi Electric Research Laboratory in Cambridge, MA (USA).

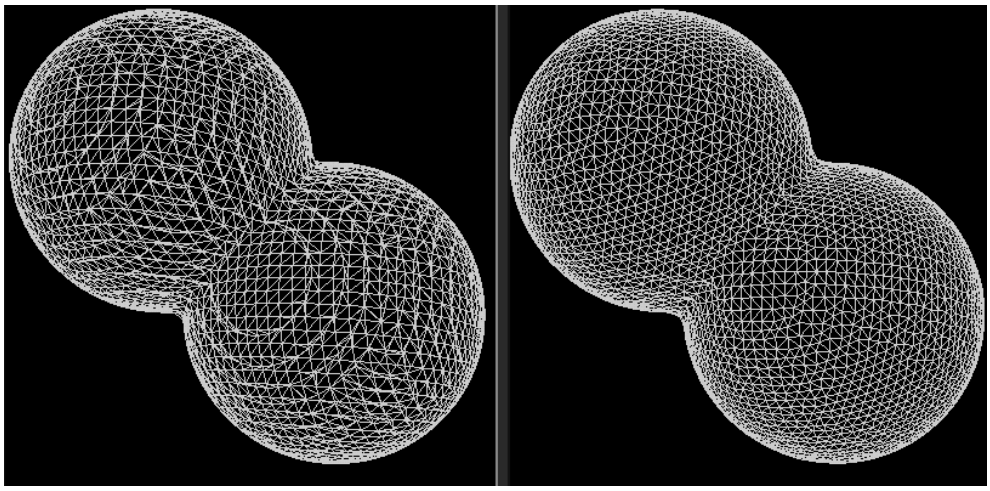


Figure 3.6: Two spheres partly overlapping. Meshes generated by Marching Cubes (left) and SNERT (right). Both meshes have the same number of triangles.

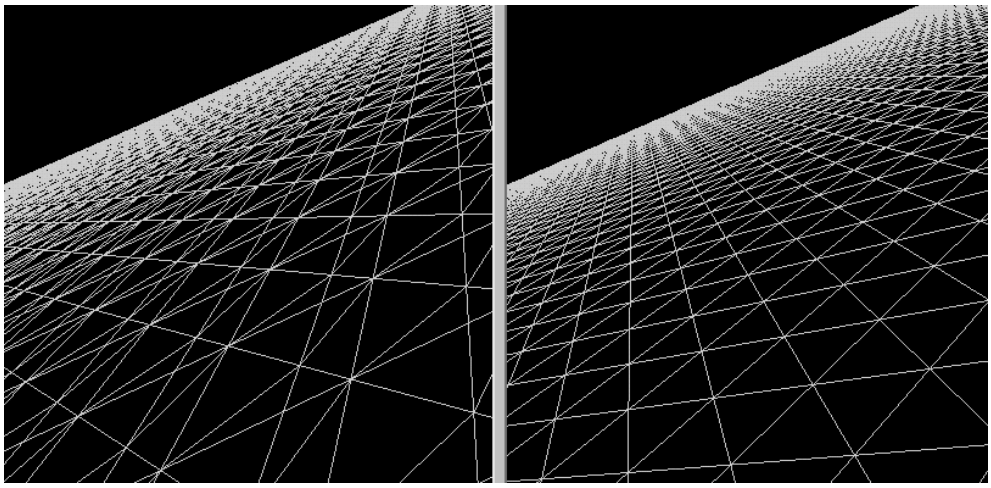


Figure 3.7: Generated mesh using Marching Cubes (left) and SurfaceNets (smoothing+gradient) (right). A view of a plane is shown.

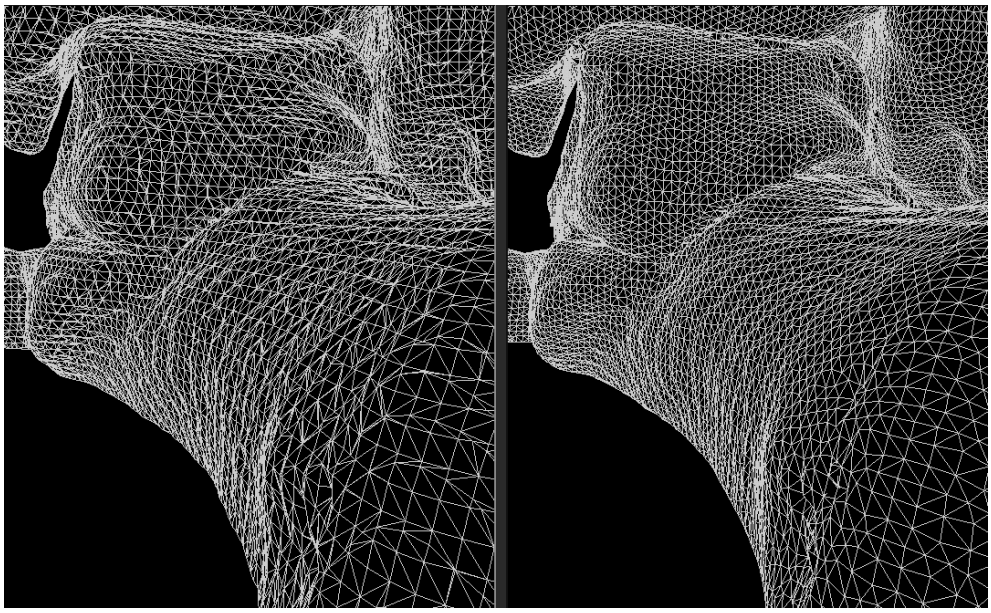


Figure 3.8: Mesh generated by Marching Cubes (left) and SurfaceNet (right) on a greyscale image of an ankle. The dataset is a CT-scan with dimensions $132 \times 141 \times 69$.

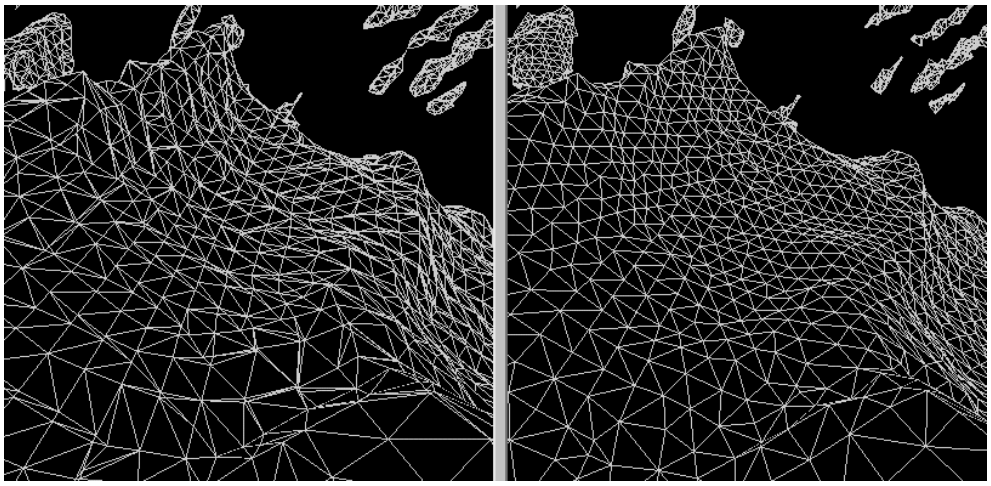


Figure 3.9: View of a bladder extracted using Marching Cubes (left) and using SurfaceNets (right).

3.5 Addendum

3.5.1 Recent developments

Extended Marching Cubes [73] is an extension of Marching Cubes (see Section 2.1.2) that improves the appearance of sharp edges in the extracted mesh. Each cell is classified either as normal or as having a sharp feature (sharp edge or corner). Normal cells are processed as usual by MC. In feature cells an extra point is inserted *in* the cell on the location of the feature and a triangle fan is constructed with the inserted point as center. Finally, an edge swap (see Section 2.2.6) is applied if two features are connected after swapping. This adaptive technique retains sharp features in the image. The insertion of a point in a cell puts EMC somewhere inbetween MC and SurfaceNets. Consequently, SurfaceNets can easily be extended to include the placement of a vertex on a feature idea from EMC.

In [69] the SurfaceNets datastructure is seen as the dual of MC (vertices of the SurfaceNets mesh correspond to faces of the MC mesh and vice versa). In the paper an alternative to the feature classifier of EMC is presented that uses the SurfaceNets datastructure to build a mesh.

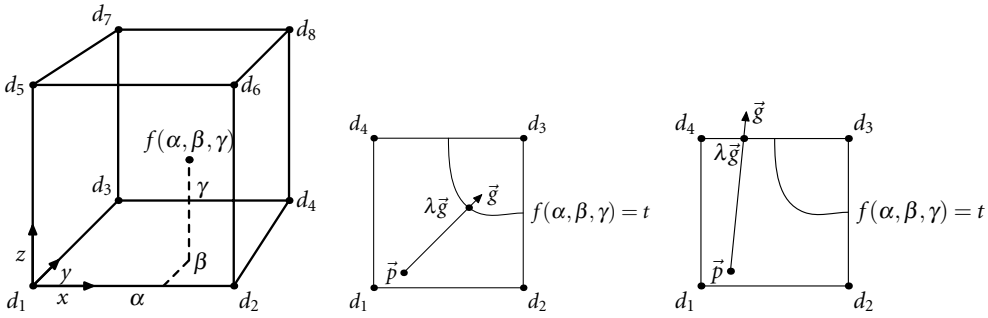
In [134] a level-set approach [100] is taken to the problem of extraction of surfaces from binary volumes that is inspired by SurfaceNets. The method is interesting because it does not use an explicit surface model. Instead, it operates directly on the volume and it is equally easy to extract a surface or a volume. The latter is of interest for volume rendering techniques.

3.5.2 Further work

Exact iso-surface intersection

An alternative to the simple iterative interpolation scheme that is used by SurfaceNets (see Section 3.2.3) is to calculate the intersections with the iso-surface by using the trilinear interpolation function directly. By doing so, repeated interpolation steps would be prevented. The trilinear interpolation function $f(\alpha,\beta,\gamma)$ (see Figure 3.10(a) on the next page) is defined by

$$\begin{aligned} f(\alpha,\beta,\gamma) = {} & d_1(1-\alpha)(1-\beta)(1-\gamma) + d_2\alpha(1-\beta)(1-\gamma) \\ & d_3(1-\alpha)\beta(1-\gamma) \quad\quad + d_4\alpha\beta(1-\gamma) \\ & d_5(1-\alpha)(1-\beta)\gamma \quad\quad + d_6\alpha(1-\beta)\gamma \\ & d_7(1-\alpha)\beta\gamma \quad\quad\quad + d_8\alpha\beta\gamma \end{aligned} \tag{3.8}$$



(a) Trilinear interpolation in a cube.

(b) Intersection with the isosurface.

(c) Missing the isosurface

Figure 3.10: Given a point $\boldsymbol{p}$, a vector $\boldsymbol{g}$ and a threshold t, calculate the intersection $\boldsymbol{p}+\lambda\boldsymbol{g}$ with the isosurface.

where (α,β,γ) is the local coordinate in the cube and d_i are the data values at the eight corners. Using this function a position is sought such that

$$[\alpha\beta\gamma] = \boldsymbol{p}+\lambda\boldsymbol{g} \quad . \tag{3.9}$$

Substituting Equation 3.9 reduces Equation 3.8 on the preceding page to a polynomial in λ of degree 3. Solving for λ yields three roots, of which the real-valued ones correspond to the iso-surface intersections. Unfortunately, it is possible to entirely miss the iso-surface as is shown in Figure 3.10. Therefore, it is necessary to repeat the intersection calculation in those cases and the advantage of stepping at once to the correct intersection using the trilinear interpolation function is lost. The same argument can be made for other intersection calculation methods. Of course, the EMC approach described in Section 3.5.1 to place a node according to a feature is interesting in this respect, but has not been implemented by us. In [69] this point placement method is used, but without the sequential mesh updating method.

Adaptive cell size

An approach that was considered to reduce the number of triangles was to use an adaptive cell size, based on a measure of surface complexity (curvature, planarity). For example, a block of eight cells could be merged into a single cell. The data structure would need to be reconsidered because the fact that the maximal number of neighbours is not higher than six is used extensively. Using an adaptive cell size can create nodes that are linked to many other nodes.

Gradient search

The extraction of objects from newer MRI datasets could not be performed using iso-surfacing. After the publication of this paper we investigated the possibility of moving a node according to an edge detection algorithm. Quickly, it became apparent that the confining structure of the cells was too restrictive for edge detection. For example, nodes would have to move outside a cell in order to find a local maximum of the image gradient magnitude. A new approach was needed and the next Chapter is the result of that research.

Accurate and high quality triangle models from 3D grey scale images

4

P.W. de Bruin, P.M. van Meeteren, F.M. Vos, F.H. Post, and A.M. Vossepoel, Accurate and high quality triangle models from 3D grey scale images, *in* T. Dohi and R. Kikinis, eds., *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, Springer, Sep. 2002 pp. 348–355, Fifth International Conference, Tokyo, Japan

Abstract

Visualization of medical data requires the extraction of surfaces that represent the boundaries of objects of interest. This paper describes a method that combines finding these boundaries accurately and ensuring that this surface consists of high quality triangles. The latter is important for subsequent visualization and simulation. We show that the surfaces created using this method are both accurate and have good quality triangles.

4.1 Introduction

Analysis of 3D medical images is aided by creating 3D surface representations that describe the boundaries of anatomical structures. A triangle mesh facilitates viewing and manipulating the data easily. Apart from that, these meshes can be used to calculate metrics on its size and shape. Applications include computer aided diagnosis, surgical planning, and simulation.

Extracting a mesh, or segmenting data, requires detecting and grouping of parts in the data that share certain characteristics. A straightforward method to segment data is isosurfacing (e.g., Marching Cubes [81]), where points that share the same greyscale value (the isovalue) are assumed to describe the surface. However, this assertion does not always hold due to noise and bias (e.g., in Magnetic Resonance Imaging).

Another approach to segmenting is to identify objects by an edge: an abrupt *change* in greyscale intensity. By doing so, the exact greyscale value of the points on the boundary is not relevant. Edge detection is also hampered by noise due to the sensitivity of the derivative operator to high frequency noise. Therefore, a Gaussian derivative is often used to suppress noise. Unfortunately, the reduction of noise coincides with dislocation of edges. A good trade-off is achieved between detecting and locating edges and suppressing noise by using a scale-space approach [6, 49].

The quality of an extracted mesh is important for subsequent processing steps, such as visualisation and finite element modeling methods. Here good quality triangles and a smooth mesh are preferred and sometimes even required (see Section 4.2.3). Surface extraction methods such as deformable models [86, 115] and snakes [80, 101] do not explicitly attempt to produce meshes of good triangle quality. Apart from that, deformable models and snakes require several parameters that have to be tuned for each case. Therefore, it is difficult to maintain reproducibility. The SurfaceNets method does take triangle quality into account, but is limited to isosurfaces [27, 60]. The conventional approach is to proceed after extraction by applying mesh improvement techniques. However, this is an undesirable situation because such techniques are purely geometrical and topological methods. The connection is lost between the mesh and the original data.

Summarizing, a good mesh extraction method should meet requirements concerning accuracy (edge detection), reproducibility (number of parameters), and triangle quality and smoothness of the generated mesh.

In this paper we present a new method that produces meshes that fulfill these criteria and does not lose the connection with the greyscale data. Our main contribution is the combination of two types of methods: multi-scale edge detection and mesh improvement. We will discuss several novel adaptations of the techniques used, and we show how the number of user-specified parameters is kept to a minimum.

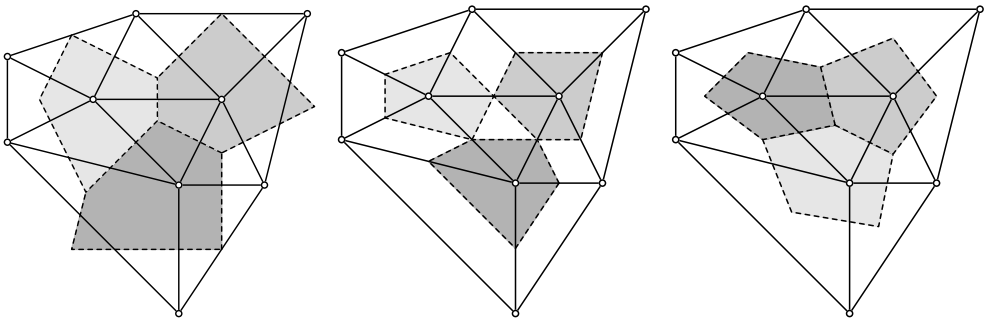


Figure 4.1: Constraining regions based on (from left to right) Voronoi, edge midpoint and centroid criterion. The triangulation does not meet the Delaunay criterion, hence the odd shape. The edge midpoint does not cover the entire surface. The centroid method is insensitive to the Delaunay criterion *and* fills the entire mesh area.

In the next Section first each of the techniques indicated above is explained, followed by our method of applying these techniques. Section 4.3 shows the results of our method when applied to synthetic and to medical data. Conclusions and future research can be found in Section 4.4.

4.2 Methods

To achieve the goals and criteria outlined in Section 4.1 a two-fold approach is used in which high quality scale-space edge detection and mesh improvement methods are combined. Our approach consists of alternating between edge-detection and mesh improvement until a desired result is obtained. Each of the separate methods is explained below, followed by aspects of the combination.

4.2.1 Scale-space edge detection

Edge detection based on image gradients is the identification of intensity changes between a region of interest and its surroundings. The modulus of the first derivative of an image has a local maximum at locations corresponding to edges. However, high-frequency noise will yield local maxima as well. Therefore, noise suppression is necessary. A common solution to this problem is to apply a low-pass filter to the image. Although this suppresses noise, it dislocates and suppresses edges also. The Gaussian kernel optimises the criteria of good signal-to-noise ratio and good localisation simultaneously [32]. The parameter σ of the Gaussian kernel determines the level of smoothing that is applied to the image.

Scale-space theory embeds an image in a one-parameter family of derived images: the scale space [49, 139]. A stack of images is created by convolving the original image with increasingly smoothing Gaussian derivative kernels.

The image corresponding to the largest value for σ is severely blurred, but noise is suppressed. In this image the detection of edges is a simpler task. Once an edge is detected, the found location is taken as the initial position in the image corresponding to the next lower value for σ. This process continues until the lowest σ is reached. At this point the edge is localised completely.

In order to limit the search space a simple representation of the object to extract is required. This coarse segmentation does not need to be accurate but an estimate of the maximum distance from the mesh to the real object is required. The search for edges is limited to the region indicated by the mesh and the maximum distance.

The following three important points have to be addressed: 1) selecting the upper and lower bound of σ, 2) sampling the scale space, and 3) locating the edge (selection of maximum).

The upper and lower bound σ selection is important because not *every* scale is relevant. The upper bound effectively determines the region of capture of the edge detector and is set equal to the maximum deviation of the initial mesh to the real object. The lower bound of σ is set equal to voxel length.

Next, a discrete set of σ's between the upper and lower bound of σ is calculated. A straightforward linear sampling is not appropriate because this might lead to aliasing at fine scales and over-sampling at coarse levels of scale. We have chosen for a logarithmic sampling method described by $\sigma_i = \sqrt[n]{2^i}$ [128]. The number n determines how many times σ is sampled per doubling of the scale parameter. By doing so, a proper and intuitive sampling is obtained because a large σ corresponds to a large stepsize and a small σ corresponds to a small stepsize.

At every vertex in the initial mesh the grey scale gradient vector is calculated. Along this vector the search for the edge is performed. The search window is made scale-relative to avoid distraction by spurious edges. The total width of the search window is set equal to 2σ . At five points along the line the modulus of the gradient is calculated with step size equal to 0.5σ. The point corresponding to the maximum value of the five points is the starting point in the next lower scale. Sampling the search window at 5 discrete positions introduces an uncertainty of half the step size. Because the lowest scale we encounter is $\sigma = 1$, the localisation will have an uncertainty equal to 0.25 voxel lengths. When going from one scale to the next lower the new search window is placed centered at the found maximum. If the new window exceeds the boundaries, it is translated along the gradient vector until it is completely located inside the region of capture. Without this correction the search could move outside the region of capture of the detector.

4.2.2 Mesh improvement techniques

Mesh improvement techniques operate either on the geometry or the topology to improve the quality of a mesh [51] (for measures of mesh quality see Section 4.2.3). Geometric methods reposition vertices, whereas topological methods operate on the connections between the vertices. For our work we have chosen one of each type: Laplacian smoothing (geometric) and edge swapping (topological).

The Laplacian smoothing operator $\mathbf{S}_\mathrm{L}$ (see e.g., [51]) moves each vertex $\boldsymbol{v}_i$ of a mesh to the average position of its N linked neighbour vertices $\boldsymbol{v}_j$ by

$$\bar{\boldsymbol{v}}_i^{\,L} = \mathbf{S}_\mathrm{L}(\boldsymbol{v}_j) = \frac{1}{N}\sum_{j=1}^{N}\boldsymbol{v}_j\,, \quad j \neq i \quad .$$

The position of each vertex in the mesh is updated after all new positions are calculated.

Advantages of Laplacian smoothing are that it is a computationally inexpensive operation, it produces a mesh with a smooth surface, and it does not require any parameters to tune. Disadvantages of this method are that it does not *guarantee* an increase of the triangle quality (e.g., vertices placed symmetrically around a vertex), it shrinks the mesh, converges slowly, and it can introduce geometric errors.

A geometric error is an inversion of the triangle orientation, which is detected by a change in direction of the triangle normal (e.g., pointing outward from a closed object surface). To prevent geometric errors a region of constrained movement for each vertex is created. These regions should be mutually exclusive and collectively exhaustive. By doing so, the region of movement for each vertex is as large as possible without the possibility of geometric errors. Figure 4.1 shows three constraining regions. The first is a Voronoi region. Here, the regions are constructed by connecting the regions created by a Delaunay triangulation (i.e., the circumcircle of each triangle is an empty circle). The figure shows that if the mesh does not meet the Delaunay criterion the regions overlap. The second idea is to connect the midpoints of the edges as is shown in Figure 4.1. The regions do not overlap, but do not cover the entire surface area either. We propose a new method where we create an allowable region of movement for each vertex by connecting the centroids of the triangles surrounding the vertex. This method covers the entire mesh surface and the regions are mutually exclusive.

Our centroid smoothing method $\mathbf{S}_\mathrm{C}$ prevents geometric errors and is expressed as a weighted sum of standard Laplacian smoothing $\mathbf{S}_\mathrm{L}$ and the current vertex $\boldsymbol{v}_i$

$$\bar{\boldsymbol{v}}_i^{\,C} = \mathbf{S}_\mathrm{C}(\boldsymbol{v}_i, \boldsymbol{v}_j) = \frac{1}{3}\boldsymbol{v}_i + \frac{2}{3}\mathbf{S}_\mathrm{L}(\boldsymbol{v}_j) \quad .$$

Edge swapping is a technique to improve the quality of the triangles in a mesh.

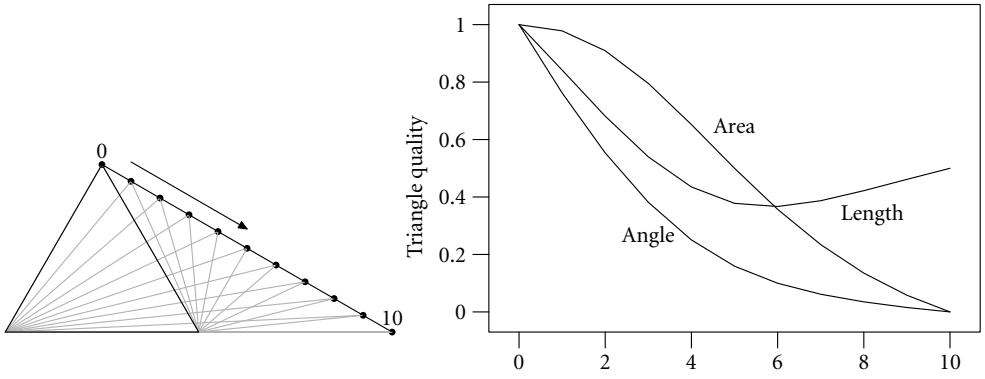


Figure 4.2: A comparison of three triangle quality criteria. Starting with an equilateral triangle the top vertex is moved along the line indicated by the arrow until the triangle is degenerate. At each point along the line the quality is calculated and displayed in the graph.

Each pair of triangles sharing an edge is considered as a quad with the edge as a diagonal. According to a criterion the edge is swapped, or not.

Here, the edge swapping algorithm proposed by [54] and improved by [19] is used. The algorithm improves the *regularity* of a mesh. For each quad in the mesh a *relaxation index* R_i is calculated that depends on the connectivity number of each vertex. Let the degree of the four vertices be d_1, d_2, d_3, and d_4, then R_i is defined by $R_i = (d_1 - 6)^2 + (d_2 - 6)^2 + (d_3 - 6)^2 + (d_4 - 6)^2$. An edge is swapped if the relaxation index after swapping is closer to zero. This method actively creates vertices with connectivity number 6, which allows all the incident angles to be close to 60 degrees.

4.2.3 Quality measures

The edge detector places vertices on the edge with an a priori known inaccuracy of 0.25σ. The inaccuracy of a mesh is defined by the distance from the centroid of each triangle to the surface as defined by the edge detector, averaged over all triangles.

There is no generally agreed definition of mesh quality; the quality measure depends on the subsequent use of the generated mesh. For specific cases quality indicators can be derived based on the approximation function and a known target. More generally, the consensus appears to be that a good quality mesh consists of triangles with not too small and not too large angles. Apart from that, meshes with regular or smoothly varying elements are "visually pleasing" [12].

A simple approach is to define the equilateral triangle as the highest quality triangle. Straightforward expressions of the quality of a triangle are the ratio of the shortest edge to the longest edge and the ratio of the smallest to the largest angle. Both criteria

define the equilateral triangle as the perfect triangle with quality 1 (one). However, it is required that an ideal mesh (with average quality 1) consists of *equal-area* triangles. Therefore, a third criterion is considered that weighs the area A of the triangle in the quality measure [8] (the area of a unit equilateral triangle is $\sqrt{3}/4$. The criteria are:

$$q_{\text{length}} = \frac{\min(|\boldsymbol{e}_1|, |\boldsymbol{e}_2|, |\boldsymbol{e}_3|)}{\max(|\boldsymbol{e}_1|, |\boldsymbol{e}_2|, |\boldsymbol{e}_3|)},$$
$$q_{\text{angle}} = \frac{\min(\alpha, \beta, \gamma)}{\max(\alpha, \beta, \gamma)},$$
$$q_{\text{area}} = \frac{4\sqrt{3}A}{|\boldsymbol{e}_1|^2 + |\boldsymbol{e}_2|^2 + |\boldsymbol{e}_3|^2}$$

where edges are indicated by $\boldsymbol{e}_i$, angles are represented by α, β and γ and A denotes the area of the triangle.

Figure 4.2 shows a graphical representation of the behaviour of the quality functions in a specific situation. Clearly, the q_{length} criterion is not well behaved, because for increasingly lower quality triangles the criterion outcome increases. The q_{angle} and q_{area} criteria are well behaved because they are bijective and monotous mappings. We have chosen the q_{area} criterion for our experiments.

4.2.4 Protocol

We have described three processes that are necessary to extract a mesh with accuracy and quality. Here we describe the protocol consisting of successive applications of each of the methods. The consecutive steps in the method are: 1) Edge detection (ED), 2) Edge swap (ES), 3) (Multiple) improvement steps (S), 4) repeat steps 1,2, and 3 until desired quality is reached, and 5) a final ED and ES.

The first step should be an edge detection step to ensure that all the vertices are positioned on the boundary. Next, we proceed by regularising the mesh using the edge swap. We have found in our experiments that smoothing steps after an edge swap converge faster than starting without an edge swap. Subsequently, one or more modified Laplacian smoothing steps are applied, followed by an edge detection step and edge swap. The region of capture of the edge detector is known and, therefore, several smoothing steps can be applied to achieve faster mesh improvement. The distance that each vertex travels from its initial position is calculated and if this distance exceeds the region of capture of the edge detector the next cycle is started. When the quality of the triangles has reached a desired value the sequence is ended with an edge detection step and an edge swap to ensure that each vertex is on the boundary.

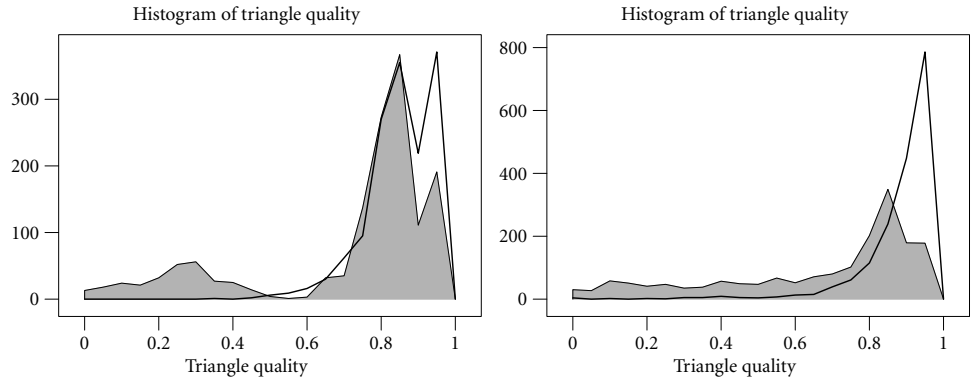


Figure 4.3: Histograms of triangle quality (q_{area} criterion) before (solid grey) and after processing of a sphere with additive noise (left) and the Os Lunatum (right).

4.3 Results

The method was applied to 3D images of a sphere and CT images of a human wrist.

A grey scale volume ($100 \times 100 \times 100$) containing a sphere of diameter 25 placed at the centre is used to test our method.

Table 4.1 shows the results from a run of the complete protocol. The table shows that the difference between each radius of the mesh after an edge-detection step (ED) and the real radius never exceeds 0.25. Therefore, the vertices are placed within the precision of the edge detector. The convergence of modified smoothing steps (S) is slow, but several steps can be applied before ED is necessary. The variance after a modified smoothing step is lower than after ED. This can be attributed to the low-pass nature of smoothing and the high-pass nature of edge detection. Note that the variance after ED is a tenth of the initial variance. Figure 4.3 shows a histogram of the triangle quality of the mesh before and after processing by our method. Clearly, the method removes low-quality triangles and increases the amount of high-quality triangles. Similar results are obtained where the sphere was corrupted with additive Gaussian noise ($\sigma = 2$).

Next, we present some of the results obtained from a CT dataset of a human wrist. Four different metacarpal bones were processed: the os lunatum, the os scaphoid, the os triquetrum, and the os trapezium (see Figure 4.4). Table 4.2 shows the results of the protocol. From the table it follows that the quality obviously increases and the inaccuracy decreases for all bones. The os lunatum and the os scaphoid allow more modified smoothing steps until an edge detection is required. Figure 4.3 shows histograms of triangle quality of the os scaphoid and the os lunatum before and after

Table 4.1: The results of the protocol on the sphere dataset. The original mesh consists of 720 vertices and 1436 triangles. An edge-detection step is indicated by ED, modified smoothing by S, and edge swapping by ES. Of the multiple smoothing steps only the first and last are shown in each case.

Operation	Average Min. Angle	Average Quality	Average Inaccuracy	Mean Radius	Variance
Org	35.41	0.76		24.78	0.471
ED	35.01	0.75	0.08	25.01	0.038
ES	36.15	0.77			
S	42.53	0.89		24.82	0.012
S	2 smoothing steps performed				
S	44.35	0.91		24.27	0.016
ED	44.49	0.91	0.121	24.98	0.049
ES	44.73	0.91			
S	46.16	0.92		24.80	0.016
S	5 smoothing steps performed				
S	47.37	0.93		23.76	0.017
ED	47.41	0.93	0.130	24.92	0.044
ES	47.41	0.93			

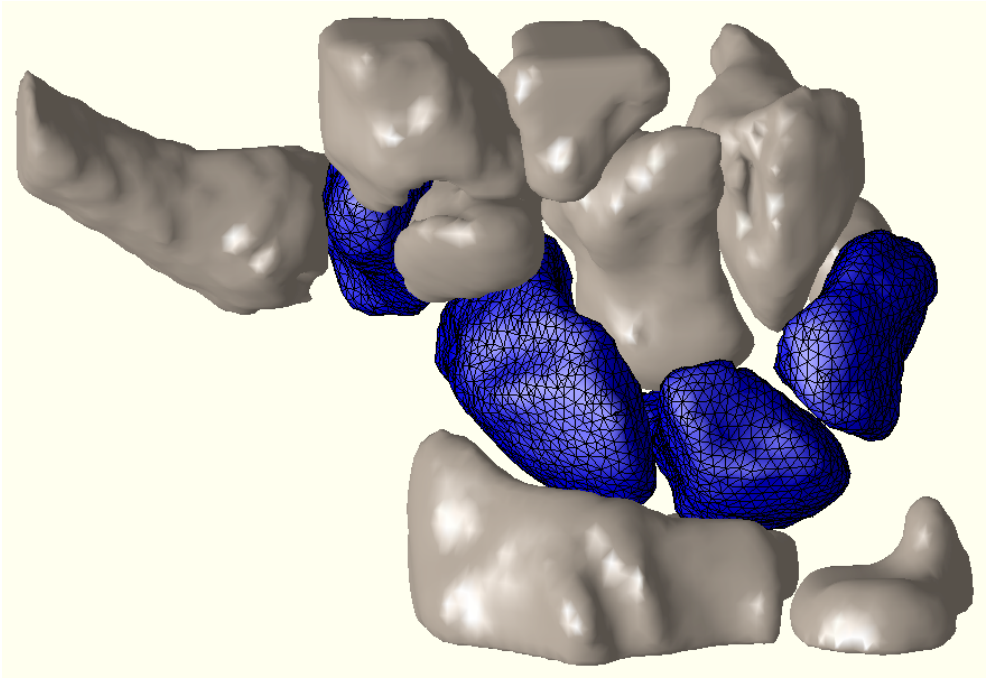


Figure 4.4: A rendering of the wrist. The highlighted bones were processed by our method. From left to right the bones are: the os trapezium, the os scapoid, the os lunatum and the os triquetrum.

processing. The histograms show a distinct improvement of the mesh quality.

4.4 Conclusions

We have created a high-quality scale-space edge detector that requires an initial approximation of the shape to extract, and an upper bound estimate of the deviation from the actual object. A method to improve the mesh without generating geometric errors was shown. Both methods were succesfully coupled by the sequential application protocol. The meshes generated by our method are both accurate and consist of triangles of high quality. Apart from the initial mesh and the maximum deviation of this mesh, the only input required from a user is the desired quality of the mesh.

Further research will focus on adjusting the resolution of the mesh. Validation using phantom models in CT and MRI modalities is planned.

Table 4.2: Results using the protocol on four metacarpal bones. Edge detection is indicated by ED, modified smoothing by S, and edge swapping by ES. For each ED step the inaccuracy is calculated. Note that the number of S steps is different for some bones (indicated by ↓). For each bone the number of vertices and triangles is indicated.

Operation	**Os Lunatum** 882v and 1760t			**Os Scaphoid** 1206v and 2408t			**Os Triquetrum** 668v and 1368t			**Os Trapezium** 790v and 1576t		
Average→	Min. Angle	Quality	Inacc.	Min. Angle	Quality	Inacc.	Min. Angle	Quality	Inacc.	Min. Angle	Quality	Inacc.
Org	31.41	0.69		31.54	0.68		30.92	0.67		31.41	0.68	
ED	31.64	0.69	0.27	31.73	0.69	0.30	31.40	0.69	0.36	31.58	0.69	0.65
ES	33.02	0.71		33.36	0.71		32.77	0.71		33.61	0.72	
S	41.88	0.87		42.88	0.88		41.81	0.87		42.86	0.88	
S	43.16	0.88		44.12	0.90		↓	↓		↓	↓	
ED	43.37	0.89	0.18	44.27	0.90	0.16	41.96	0.87	0.20	42.89	0.88	0.29
ES	43.51	0.89		44.36	0.90		42.13	0.87		43.04	0.88	
S	45.05	0.90		45.89	0.91		44.71	0.90		45.67	0.91	
S	45.42	0.91		46.28	0.92		45.36	0.91		↓	↓	
ED	45.57	0.91	0.18	46.38	0.92	0.16	45.45	0.91	0.19	45.46	0.91	0.26
ES	45.57	0.91		46.38	0.92		45.48	0.91		45.45	0.91	

4.5 Addendum

4.5.1 Recent developments

Preventing geometric errors

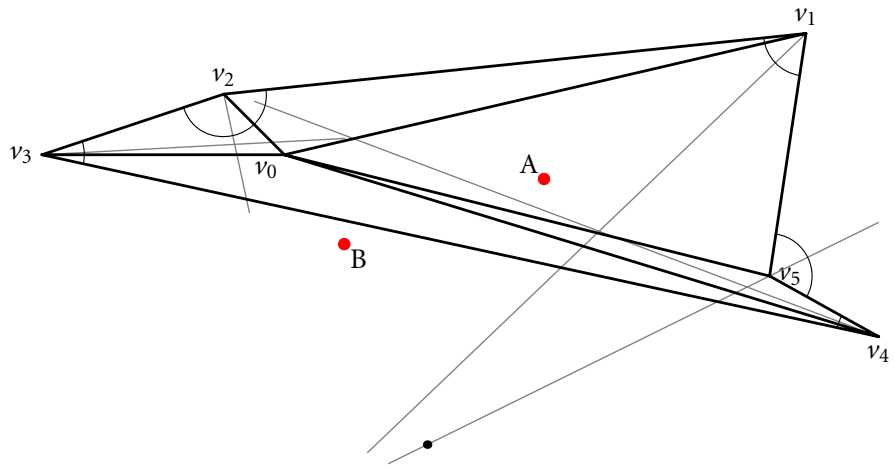
Geometric errors can occur when using Laplacing smoothing (see Section 2.2.3). All of the Laplacian smoothing variants presented in Section 2.2.4 potentially suffer from this problem. For example, angle-based smoothing [142] claims to prevent errors from occurring, however, Figure 4.5(a) on the next page shows a counterexample where the center vertex moves outside the 1-ring neighbourhood. A similar example is shown in Figure 4.5(b) on the facing page for inverse edge-length weighted smoothing.

The cause of the problem of geometric errors is a combination of three factors:

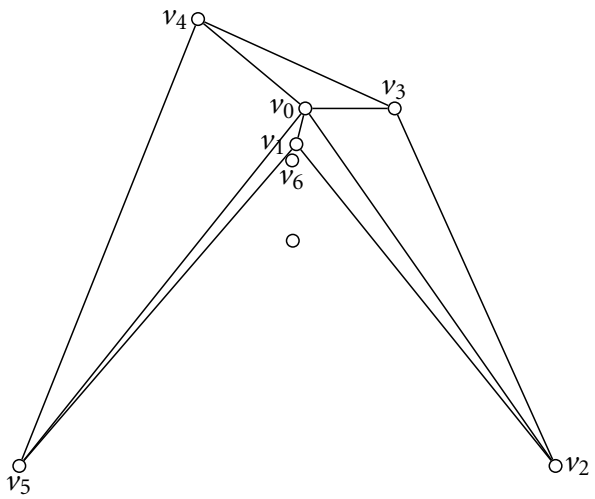
- a concave 1-ring neighbourhood;
- the vertex update mechanism;
- overlapping regions of movement.

The problem of the concave 1-ring neighbourhood is illustrated in Figures 2.15 and 4.5(b). The standard vertex update mechanism for Laplacian smoothing is to move a vertex to its new position immediately and to use the new location for subsequent smoothing steps. An alternative method is to calculate all new positions using the current mesh and to update all vertices at once. In the following these update methods are referred to as either *immediate* or *batch* updating. The advantage of batch vertex updating is that there is no order dependence. The results using immediate updating may vary slightly depending on with which vertex the smoothing process is started. Overlapping regions are related to the 1-ring neighbourhood problem. The effect of overlapping regions of movement is illustrated in Figure 4.6 on page 62 where a small section of a mesh is shown. The allowable range of movement of the two center vertices overlaps, indicated by the grey area. Geometric errors are therefore possible using batch updating. Note that both 1-ring neighbourhoods are convex.

The centroid smoothing method in this Chapter unfortunately does not prevent geometric errors from occurring, although the possibility of an error is low. As is shown in Figure 4.1 on page 51 the centroid smoothing method defines allowable regions of movement that cover the entire surface and are mutually exclusive. The exclusive property holds even when the 1-ring neighbourhood is concave. We assumed that the region of movement defined by the polygon through the centroids is always convex. However, Figure 4.7 on page 62 shows that it is possible to create a concave neighbourhood, and consequently, geometric errors are still possible using both immediate and batch vertex updating.



(a) Angle-based smoothing can generate inverted elements. The black dots are the perpendicular projections of v_0 onto each of the bisectors of the angles of the surrounding polygon (grey lines). Point B is the average of these points and is located outside the 1-ring neighbourhood. Point A is the average location of the linked vertices. Also see Figure 2.16 on page 26.



(b) Inverse edge-length weighted Laplacian smoothing can result in geometric errors. The new position v_6 of the center vertex v_0 is outside the 1-ring neighbourhood (similar to Figure 2.15 on page 23).

Figure 4.5: Variants of Laplacian smoothing are also vulnerable to the problem of geometric error.

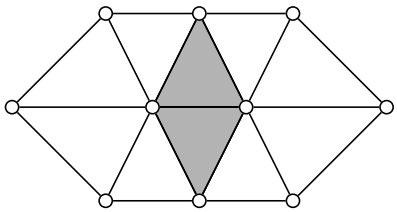
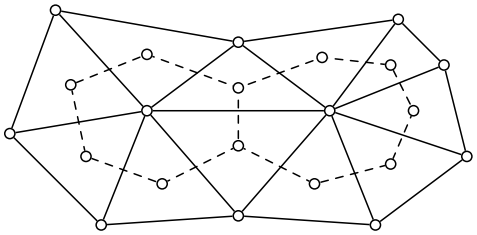
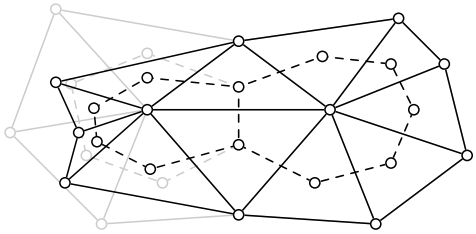


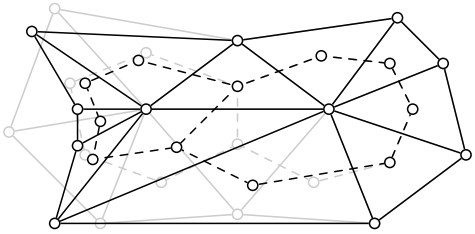
Figure 4.6: Overlapping regions of a section of a mesh. The grey area indicates the overlapping regions of movement of the two center vertices.



(a) The dashed lines connecting the centroids of the triangles indicate the allowable region of movement for each center vertex.



(b) The 1-ring neighbourhood of the left center vertex is concave. The polygons formed by connecting the centroids of the triangles are convex.



(c) Both the 1-ring neighbourhood and the allowable region of movement of the left center vertex are concave.

Figure 4.7: Several possibilities of concavity of the 1-ring neighbourhood and centroid polygons.

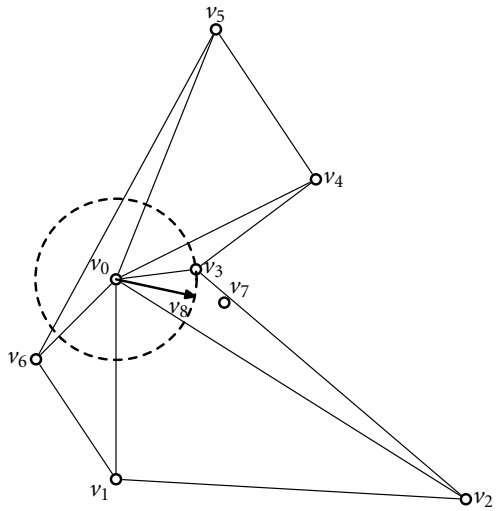


Figure 4.8: Safe smoothing. Vertex 7 is the average of the surrounding polygon of vertex 0. The range of movement is limited to the distance from vertex 0 to its closest neighbour. The center vertex is moved to vertex 8, indicated by the arrow.

A geometric error cannot occur using immediate updating as long as the relocation of a vertex is limited to the area outlined by a convex 1-ring neighbourhood. Therefore, a solution to the problem of creating inverted mesh elements is to apply an edge-length weighting scheme such that convex 1-ring neighbourhoods are guaranteed. Figure 4.8 illustrates this approach where the relocation vector of a center vertex is clipped by the distance to the closest vertex. Similarly, each edge can be weighted by the distance to the closest vertex. Effectively, this makes all edges equal to the shortest edge. By definition, a 1-ring neighbourhood of equal-length edges is convex. A drawback is that convergence of the method is slower and not the entire surface area of the mesh is available for vertex relocation.

An alternative to the prevention of errors is to repair errors. In [118] a method is outlined that operates on an iso-surface and it is of interest to research if this can be applied to our method.

Edge swapping

Although edge swapping is presented in this thesis as a clear-cut technique it is not that simple to apply. One important thing to note is that all swapping techniques operate well on convex quadrilaterals. As soon as the quadrilateral formed by two linked triangles is concave then it is possible that holes are created in the mesh (a triangle is flipped). In the software implementation of our method a range of checks

had to be built in to ensure that a swap would not result in an invalid mesh:

- Connectivity test. A swap is not allowed if the connectivity number of a vertex after the swap falls below three. A vertex with connectivity two is not part of a continuous surface.
- Concavity test. This is ill-defined, even more so because the quadrilaterals are non-planar most of the time. A simple test is to calculate the normals of each triangle before and after the swap. If the dot product of the two normals is lower than zero then a swap is not allowed.

4.5.2 Initialisation

The application of the edge-detection based method requires an initial mesh. Creating such a mesh is not a trivial task. For an iso-surface based method, such as SurfaceNets, initialisation can be performed by simple thresholding. Another method to create such a mesh, that is applicable to edge detection, is contour connecting (see Section 2.1.4). However, not only is it tedious to manually outline each contour, connecting the contours is not straightforward (see Section 2.1.4). This need for a smart manual segmentation tool was the origin of the fast interactive segmentation method that is presented in the next Chapter.

Interactive 3D segmentation using connected orthogonal contours

5

P.W. de Bruin, V.J. Dercksen, F.H. Post, A.M. Vossepoel, G.J. Streekstra, and F.M. Vos, Interactive 3D segmentation using connected orthogonal contours, *Computers in Biology and Medicine*, 2004, Accepted for publication

Abstract

This paper describes a new method for interactive segmentation that is based on cross-sectional design and 3D modelling. The method represents a 3D model by a set of connected contours that are planar and orthogonal. Planar contours overlayed on image data are easily manipulated and linked contours reduce the amount of user-interaction. This method solves the contour–to–contour correspondence problem and can capture extrema of objects in a more flexible way than manual segmentation of a stack of 2D images. The resulting 3D model is guaranteed to be free of geometric and topological errors. We show that manual segmentation using connected orthogonal contours has great advantages over conventional manual segmentation. Furthermore, the method provides effective feedback and control for creating an initial model for, and control and steering of, (semi-)automatic segmentation methods.

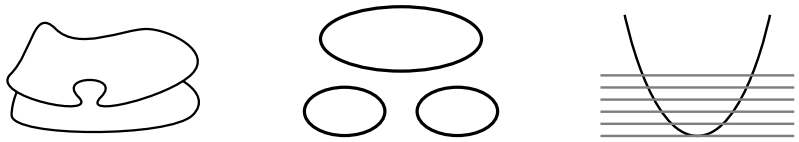
5.1 Introduction

Segmentation and visualisation of anatomical structures is important for medical applications such as diagnosis, biomechanical simulation and surgical planning. The data is obtained from medical imaging equipment, such as MRI (Magnetic Resonance Imaging), CT (Computed Tomography) or ultrasound. A large number of methods is available to segment structures of interest from the data. It is generally not possible, however, to extract a local structure of interest fully automatically. Limiting factors are noise and field inhomogeneities. Additionally, the *as low as reasonably achievable* principle minimises the radiation dose a patient incurs by X-ray imaging. By doing so, the images may not always contain the required detail for automatic segmentation.

Fully manual and semi-automatic segmentation methods usually operate in 2D (see [80, 98]). After a shape is outlined by a set of contours in a stack of 2D images, a 3D shape is constructed by connecting the contours [44, 55]. A drawback of such methods is that it is tedious work to construct each contour. In addition, it is not trivial to determine the contour–to–contour correspondence (see Figures 5.1(a) and 5.1(b)). Finally, it becomes increasingly difficult to draw contours towards the ends of the object, as the object quickly disappears from slice to slice (see Figure 5.1(c)).

Automatic and semi-automatic segmentation methods often must be initialised, for example by a geometric model. Manual segmentation methods effectively build such a model, but suffer from the problems indicated. In the fields of cross-sectional design and 3D modelling, methods have been developed to build and deform geometric models [62, 90, 105, 124]. It is difficult, however, to guarantee that the model is, and remains, free of geometric and topological errors during interaction.

In this paper we present a new method for interactively segmenting and constructing a 3D model. This method can be used either to segment completely manually, or to create a model to initialise and control more sophisticated (semi-)automatic segmentation methods. The method consists of manipulating a set of connected, pla-



(a) (b) (c)

Figure 5.1: The classic problems of connecting adjacent contours: a) contour–to-contour correspondence, b) branching object, and c) increasingly smaller contours at object extremes.

nar and orthogonal contours in 3D space. Planar contours are easily manipulated when overlayed on the image data. By using a connected contour data structure the slice–to–slice correspondence problem does not arise. Furthermore, because the contours are connected, the result of manipulating one contour immediately updates the connected counterparts. Thereby, the amount of interaction required to fit the model to an object is reduced. Additionally, it is easy to capture the extrema of objects because contours can be placed in three orthogonal views of the data.

5.2 Methods

5.2.1 Overview

The goal of our method is to extract three-dimensional object shapes by specifying cross-sections in three orthogonal directions. From these cross-sections the object surface model is constructed. The specific application for this method is the segmentation of carpal bones (part of the wrist joint). Therefore, the method assumes that the objects to be modelled have the following properties. The object shape is manifold and topologically equivalent to a sphere, but the geometry may contain protrusions and concavities. The object has no holes and does not self-intersect (genus zero).

An overview of the method is shown in Figure 5.2 on the following page. Starting with an initial template shape (an approximation of a sphere, see Figure 5.3(a) on the next page) the user can deform the model and add new contours to improve the fit of the model to an object. If contours are planar then manipulation of the entire contour overlayed on the image data is a simple task. By imposing the restrictions that contours are planar, and that at intersections at most two contours are connected, it is possible to add new contours automatically. As the user slices through a dataset the intersections of the model with each plane are shown. At any slice position the user can create a new contour by automatically linking the set of intersections in any of the three orthogonal planes. Next, the new contour can be deformed to improve the fit to the underlying object. The linked contours minimise the amount of user interaction that is required to move control points to desired locations.

5.2.2 Data structure

An object shape S is defined by a set of oriented contours (see Figure 5.3(a) on the following page). Given a 3D space spanned by an orthogonal basis $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$, each contour is planar and is oriented according to one of the base planes. Therefore, all points on a given contour share either the x, y, or z coordinate

$$S = \left\{ C_1^k, C_2^k, \ldots, C_{N_k}^k \right\}, \quad k \in \{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}\}, \quad \forall N_k \geq 1. \tag{5.1}$$

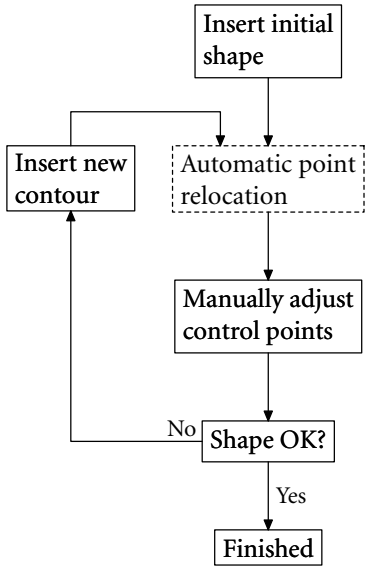
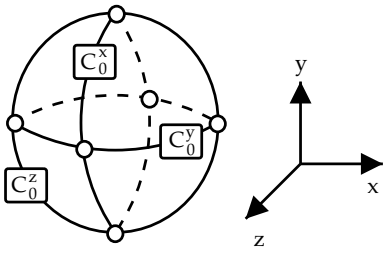
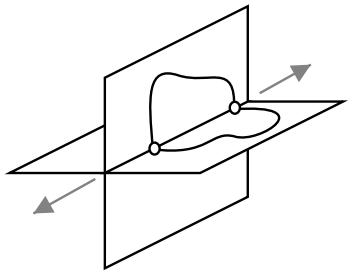


Figure 5.2: Overview of the interactive segmentation process.



(a) The initial template shape consists of three planar and orthogonal contours connected at the white dots. Depending on the interpolation function it is spherical or octahedral.



(b) A control point is allowed to move only along the intersection line of the two planes containing the contours.

Figure 5.3: The initial template model.

Each contour C_i^k contains an ordered set P_i^k of control points $\boldsymbol{p}_j$ and an interpolation function $I(P_i^k)$

$$C_i^k = I(P_i^k) \quad , \tag{5.2}$$

$$P_i^k = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M\} \quad M \geq 4 \quad . \tag{5.3}$$

The interpolation function I determines the location of points between control points. For fast evaluation, a straight line is used. For the generation of smooth curves we use Catmull-Rom splines [34]. This spline has two features that make it suitable for our application: the spline passes through the control points and it has C^1 continuity at the control points. The former makes it easy to exactly place the spline at a user-indicated position. The latter ensures a smooth transition from one spline segment to the next.

Splines are scan converted to the underlying voxel grid using a subdivision algorithm [50]. Each evaluated spline point is given a unique id that corresponds to the voxel in which it is located. An evaluated spline point is always placed at the center of the voxel and each voxel contains only one spline point per contour. By doing so, we ensure that each contour is planar and that an intersection is created in each voxel that belongs to two orthogonal contours. The latter is required because two arbitrary splines passing through the same voxel do not always intersect. Furthermore, finding the intersections of a shape with a plane reduces to traversing a list of voxels instead of finding numerical solutions to the spline equation.

The subdivision algorithm proceeds as follows. On each interval of a spline the begin ($t = 0$), end ($t = 1$), and the midpoint ($t = 0.5$) are evaluated. If the voxel containing the midpoint is empty, then the two new intervals are again subdivided. This recursion is continued until all the voxels that the spline passes through are found. The end result is a set of 4-connected voxels that form the scan converted planar contour.

By definition, at least one contour of each orientation is needed to build an initial 3D shape (Equation 5.1 on page 67). The control points are located at the intersection points of contours with different orientations. At each point where contours intersect there must be a control point and at each control point exactly two contours must intersect. Each contour must intersect at least one contour of the other two orientations. Two intersecting contours always have an even number of intersections. Therefore, a contour has at least four intersection points, and consequently, at least four control points (see Figure 5.3(a) on the facing page).

For each control point the four neighbouring control points are stored in a structure. This local connectivity information is used in Section 5.2.4 to automatically insert contours. For visualisation purposes, a polygon mesh is generated from the

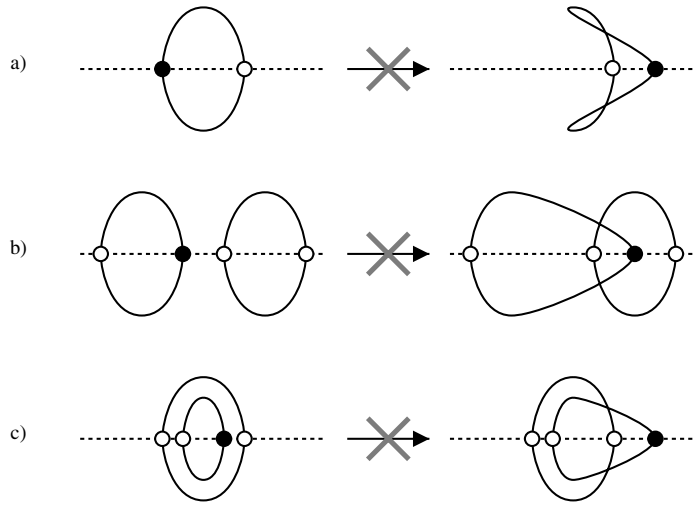


Figure 5.4: Constraint on moving a control point. The control points indicated by black dots cannot be moved past any other control points on the line of movement. Self-intersections would result in an invalid model.

linked contours by identifying all polygons enclosed by contour segments. Each polygon is then triangulated using a standard algorithm [95].

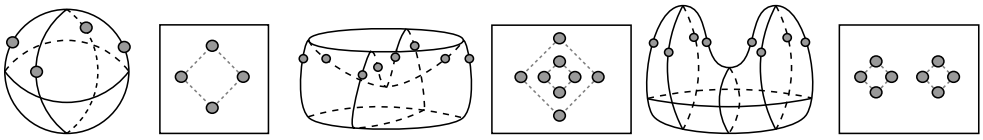
5.2.3 Model shape manipulation

By moving control points the shape model can be adapted to fit a desired object. This manipulation is made considerably easier by the planarity of the contours. Maintaining planarity of all contours requires that the movement of a control point is restricted to the intersection line of the two planes of its contours (see Figure 5.3(b) on page 68). Therefore, the interaction with a control point is always a 1D task. Note that if three planar contours were allowed to intersect, then no degree of freedom would remain for the resulting control point.

Apart from the fact that planar contours are easier to manipulate, a manifold shape is easily maintained under planar conditions. Self-intersection is efficiently prevented by requiring that a control point cannot change order with other control points on their common line of movement (see Figure 5.4).

5.2.4 Adding and removing contours

Adding contours increases the accuracy of the model because more boundary information is provided. Our method can create new contours by automatically connecting the intersections of the shape with a user-indicated plane (Figure 5.6).



(a) Sphere (b) Indentation (c) Two protrusions

Figure 5.5: Three shapes are sliced by a plane, intersections are indicated by the grey dots. a) a sphere, b) a bowl shape and c) a shape with two protrusions. The connections between the intersections are derived from the existing topology.

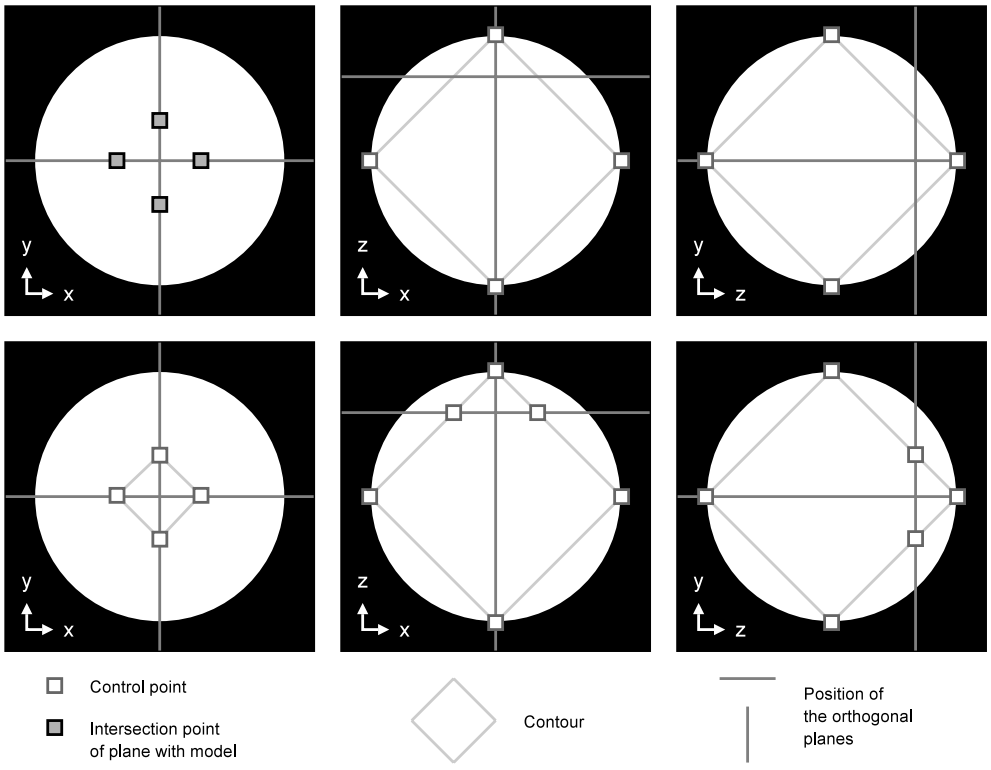


Figure 5.6: Automatically creating a contour from intersection points. The three images on the top row show three orthogonal planes before, the three on the bottom row after adding a contour. The intersection points (top left) are automatically connected to form a new contour (bottom left) after the user has indicated a plane.

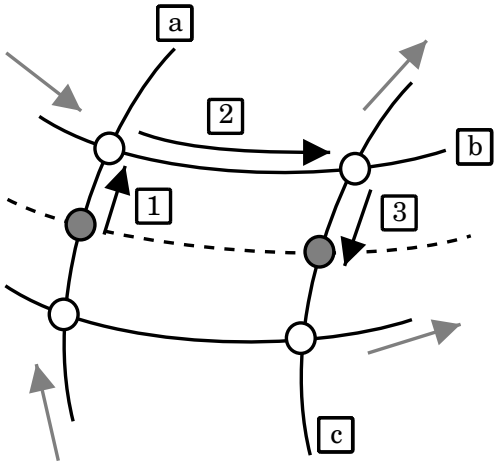


Figure 5.7: Determining the ordering of the intersections (grey dots). The dashed line is the new contour to create. The steps are explained in the text.

A new contour can be added only if two conditions are satisfied. First, the intersection plane for the new contour may not contain any control points. Otherwise, the condition would be violated that at most two contours can intersect in a control point. Second, the number of intersections of the shape with the plane must be at least four, which is the minimum number of control points required for a valid contour (see Section 5.2.2). Adding a contour to (for example) the initial sphere shape, as shown in Figure 5.5(a) on the page before, is performed by connecting the intersection points in the indicated plane.

The ordering of the control points follows from the existing topology of the contour network (see Figure 5.7). In the first step the contour on which the intersection lies (contour a) is traced until a control point is reached. In the second step, a switch is made to the linked contour (contour b) and this is traced until a control point is reached. Finally, in the third step, again a switch is made to the linked contour (contour c) and it is traced until a new intersection is reached. By doing so, the new intersection points are placed in the correct order and a new contour is formed. This algorithm applies to the case where four control points form a closed loop. Cases where a different number of control points form a closed loop are handled in a similar fashion.

Figure 5.5 on the page before shows a number of possible intersection cases. Using the ordering algorithm, a consistent shape is always maintained. Certain shapes require several contours to be added, for example, in the cases shown in Figures 5.5(b) and 5.5(c) two contours are required (see Section 5.2.5). Generally, the intersections with a plane form either a single contour (Figure 5.5(a) on the preceding page), or

(a combination of) non-intersecting sets of concentric (Figure 5.5(b) on page 71) or disjoint contours (Figure 5.5(c) on page 71).

A contour can be removed under the conditions that at least one contour of the same orientation remains in the model and that all remaining contours contain at least four control points. Removal of a contour requires the removal of all its control points, and consequently, the removal of these control points from all linked contours.

5.2.5 Concave shapes

Concave shapes are created either by moving control points deeper into or farther away from the model (see Figure 5.5 on page 71). Consistency requires that contours in an intersecting plane link *all* intersection points. By moving a control point through a plane that already contains a contour, new intersection points are generated (Figure 5.8 on the following page). The new intersections are automatically linked to form the pattern as shown in Figure 5.5(b) on page 71.

Similarly, Figure 5.8 on the following page(b) shows the creation of a protrusion. Again, if a control point is moved through a plane that already contains a contour, extra intersections are formed. Linking the new intersections creates a pattern similiar to the situation shown in Figure 5.5(c) on page 71.

Figure 5.9 on the following page illustrates a special case of creating a protrusion. If the control point intersects the plane "outside" an existing contour (Figure 5.9 on the next page(a)) then an illegal situation as shown in Figure 5.9 on the following page(b) can occur. In this case a control point would be part of two coplanar contours, and there it would violate the constraints defined in Section 5.2.2. Figure 5.9 on the next page(c) shows that in order to solve this problem, the user has to add a new contour as indicated by the arrow. By doing so, the control point can be moved through the plane and the automatically generated points form distinct coplanar contours.

5.2.6 Interaction

The method was implemented using the Visualisation ToolKit [110] and Python [83]. Our implementation will be made available online [1] in the near future. The application runs interactively on a standard PC.

Figure 5.13 on page 80 shows a screenshot of our implementation. Colour is used to guide the user in navigation and orientation in 3D. Each orthogonal plane is (optionally) given a color. Here, we use blue, red and yellow to distinguish each plane. Each line of a cross-hair is coloured according to its corresponding plane. This

[1] at `http://visualisation.tudelft.nl/~paul/fastseg`, a movie of an example segmentation session is available.

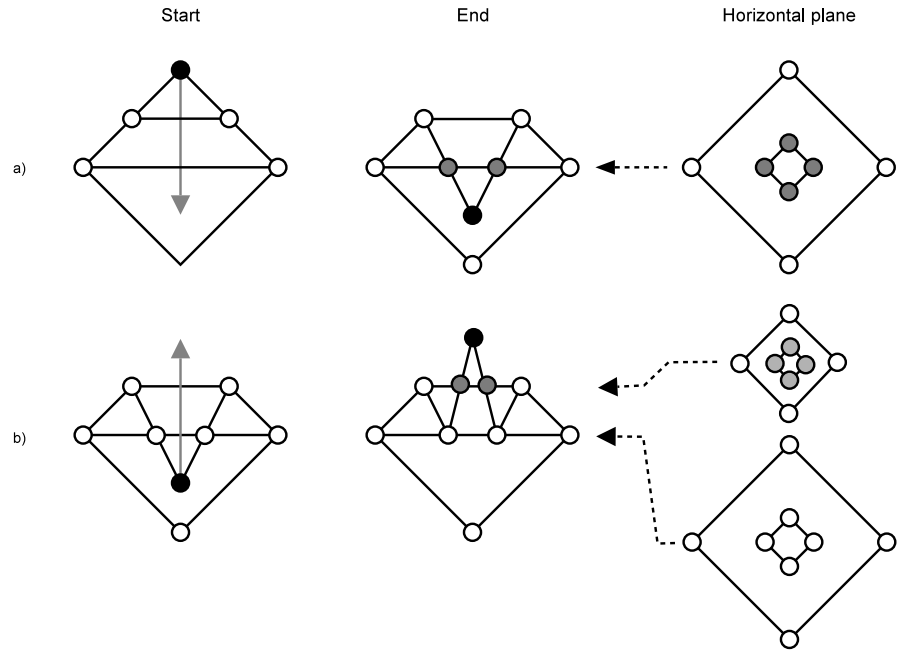


Figure 5.8: a) Creating an indentation followed by b) creating a protrusion. Appropriate extra contours (grey dots) are generated automatically as the control point (black dot) is moved through intersecting planes.

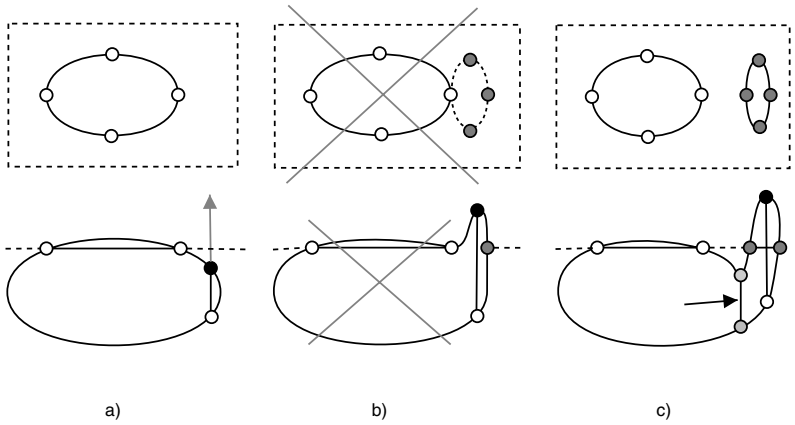


Figure 5.9: Moving a control point (black dot) through a plane already containing a control point. The plane is shown in the top row and indicated by a dashed line in the bottom row. a) When the control point is moved along the grey line passed the dashed line, a situation arises as shown in b). An illegal situation occurs because a single point is shared between coplanar contours. The solution is to add the contour indicated by the arrow.

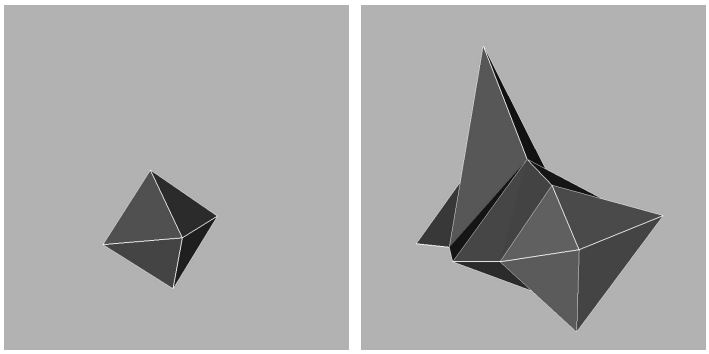
makes it easy to determine which plane is orthogonal to which and makes interaction more intuitive. For example, in the blue plane the cross-hair lines are yellow and red. Contours are indicated by green lines and control points by coloured squares. The colour of the control point corresponds to the colour of the line along which it is allowed to move. The intersections with the shape in a plane without contours are indicated by purple circles. A common mode of operation is to scroll through the data and to insert new contours in planes where the intersection points do not line up with the underlying object.

Figure 5.10 on the next page shows a number of consecutive steps of the segmentation process of a carpal bone. The initial shape in Figure 5.10(a) on the following page consists of 3 contours and the end result in Figure 5.10(e) on the next page consists of 20 contours.

This method is also well-suited to control semi-automatic segmentation methods. Any 2D or 3D method that can calculate a new position for a control point can be applied. For example, we have added to the application an adapted version of an edge-detection algorithm [26] that searches in the direction of movement of a control point for the nearest maximum of the modulus of the gradient. By doing so, the manual segmentation process can be sped up by moving control points using edge-detection enhanced manual segmentation (EDEMS). The user invokes the edge detection in the currently selected orthogonal view by pressing a key. All control points in this view are then moved to the location of the maximum, with a user-selectable search range. The calculation of new locations requires little computer time and the update is instantaneous. By doing so, the user can immediately see the result of the edge detection on *all* the affected points and can manually correct the positions. Using the method in this way speeds up the segmentation process and gives the user effective feedback and control.

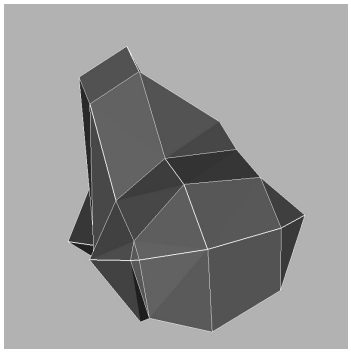
5.3 Results

To test the feasibility of the proposed method we performed an experiment where three subjects are asked to segment a number of objects (see Figure 5.11 on page 77). Synthetic datasets of a sphere, an ellipsoid, and a sphere with an indentation are created together with a reference mesh of each object (see Figure 5.12 on page 77). The accuracy of a segmentation is defined by the average and maximum distance from each point on a contour to the reference mesh. In addition to the synthetic objects, a CT dataset containing a separately scanned carpal bone is used. For this object a reference mesh is created by a multi-scale edge-detection method [26]. Of the three test subjects one had prior experience with our application, the others used it for the first time. After a short introduction to the software using the sphere dataset, the test

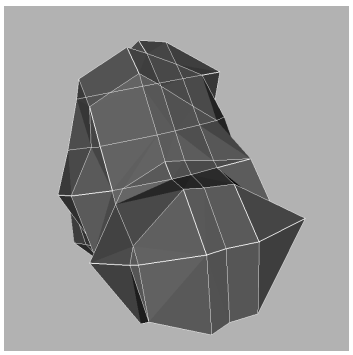


(a) Initial sphere, 3 contours

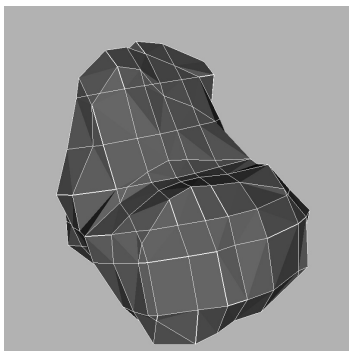
(b) 6 contours



(c) 9 contours

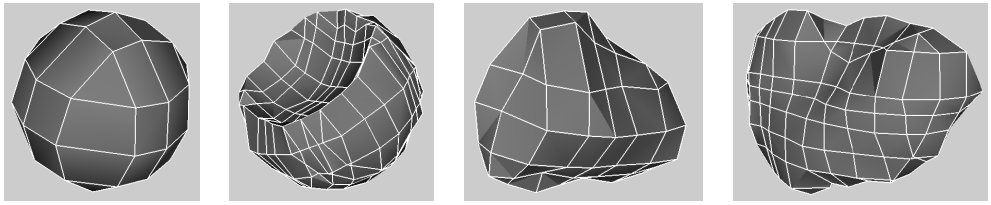


(d) 14 contours



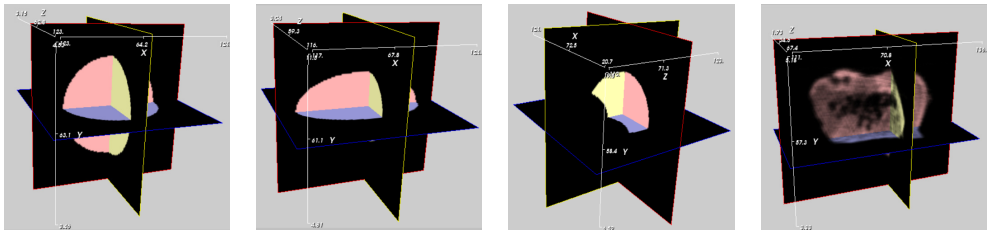
(e) Final segmentation result, 20 contours

Figure 5.10: Illustration of the segmentation process of a carpal bone.

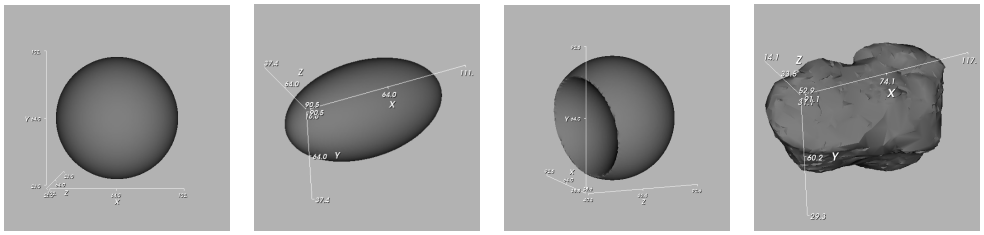


(a) (b) (c) (d)

Figure 5.11: Segmentations of a) a sphere, b) an indented sphere, c) a carpal bone in an early stage and d) the completed carpal bone.



(a) (b) (c) (d)



(e) (f) (g) (h)

Figure 5.12: Objects used for experiments. The top row shows the datasets, the bottom row shows the reference meshes. a) and e) Sphere used during training period. b) and f) Ellipsoid. c) and g) Indented sphere. d) and h) Carpal bone.

subjects were asked to segment the three remaining datasets. The results are shown in Table 5.1 on the facing page. It can be seen that after a short training period test subjects are able to segment a carpal bone in under 10 minutes time with an average accuracy of 2.5 voxel lengths.

User actions during a segmentation are logged. For example, we keep track of the distance each control point is moved, when new contours are inserted, and the accuracy of the current model. Figure 5.15 on page 82 illustrates the interaction of a user during a manual segmentation of the indented sphere dataset. It can be seen that the distance that control points are moved decreases as the model is refined. The time interval between contour insertions is initially short, but increases over time. We observed that a user spends this time searching for an appropriate location to insert a new contour. Both the maximum and the average approximation error decrease. The occasional increase of the errors occur because an insertion of a contour adds several control points to the model and consequently, more points are used to calculate the approximation error.

The manual segmentation process can be sped up by using EDEMS. During each segmentation session the distance that each control point travels manually and automatically is logged. A comparison was made by segmenting a carpal bone manually, and by using EDEMS. In both case contours were added in approximately the same locations and the total number of added contours is 15. The results of the manual and EDEMS processes are shown in Figure 5.16 on page 83 and Figure 5.18 on page 85 respectively. The manual process takes twice the time of the EDEMS session. In Figure 5.16 the distance that each control point travels decreases over time, but after each new contour insertion a set of control points has to be relocated. The EDEMS session in Figure 5.18 shows that the amount of manual interaction decreases during the session and is limited to a few corrections near the end. Three carpal bones are segmented using the assisted segmentation. The distance travelled automatically by control points was between 59% and 64% of the total distance. The number of manual control point relocations decreases from 104 in the fully manual session to 26 in the EDEMS session.

The method was also applied to three other datasets. Figure 5.17(a) on page 84 shows the end result of manually segmenting a MRI dataset of a tumor located between the eyes. Figure 5.14 on page 81 shows a screenshot of our application after manually segmenting a MRI dataset of a brain tumor. During segmentation the surface area and volume of the model can be calculated. In this case, the volume of the model is 12.4cm^2. Finally, we applied the EDEMS method to a MRI dataset of a large myoma (fibroid). In Figure 5.17(b) on page 84 the segmented myoma and bladder are shown.

Table 5.1: Test subject segmentation results. Subject A has experience using the method, subjects B and C are novices.

	Ellipsoid			Indented Sphere			Carpal Bone		
User	time [s]	avg.	max.	time [s]	avg.	max.	time [s]	avg.	max.
		error [voxels]			error [voxels]			error [voxels]	
A	271	0.92	3.1	430	0.96	3.6	533	2.50	6.1
B	284	0.96	2.5	731	0.84	3.7	576	2.65	6.6
C	320	0.88	3.6	591	0.92	4.8	364	2.60	6.8

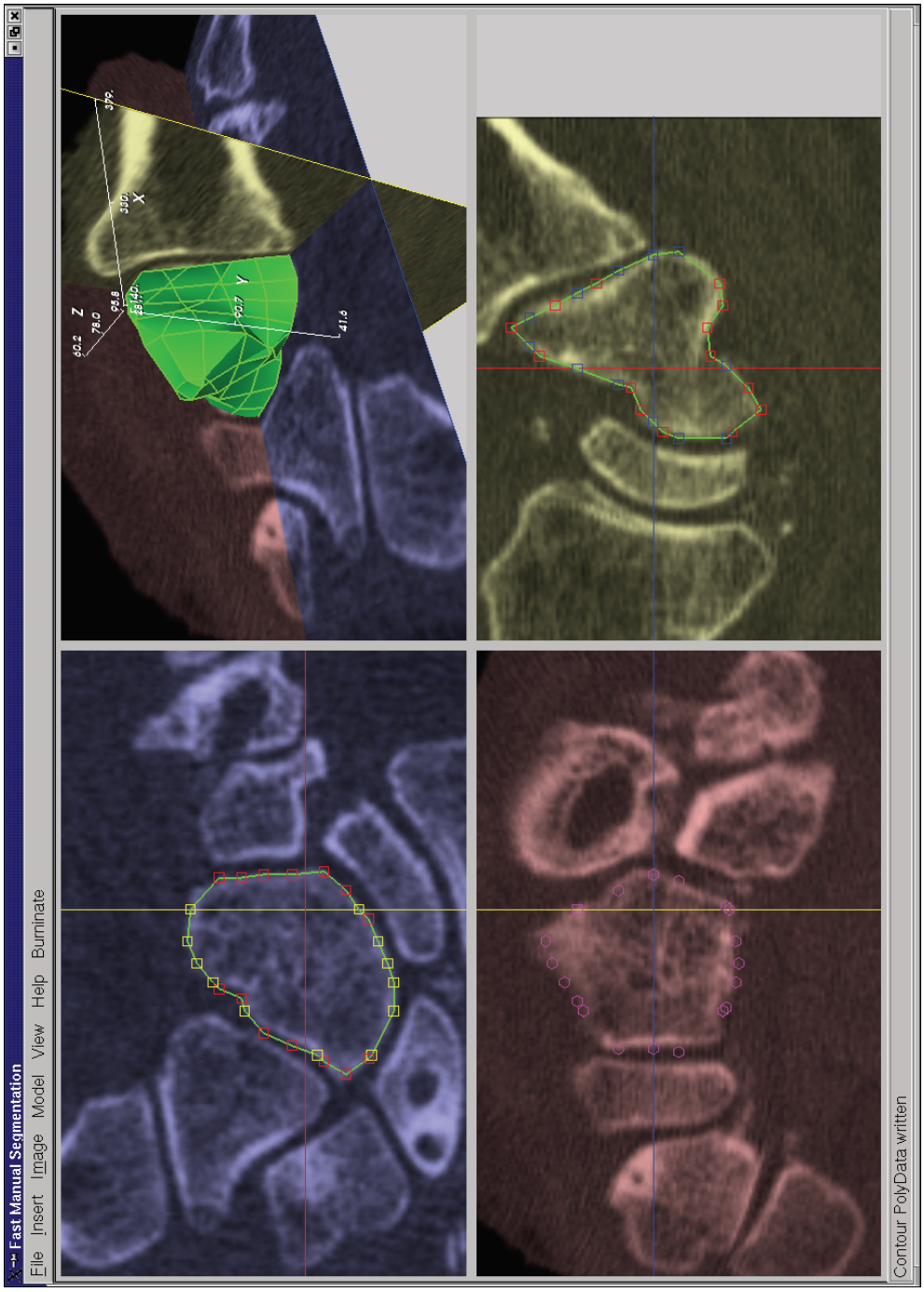


Figure 5.13: Screenshot of the segmentation application. A CT dataset of a human wrist and several contours are shown. The colour of the lines of the crosshair and the colour of each control point correspond to the colour of the intersecting orthogonal planes.

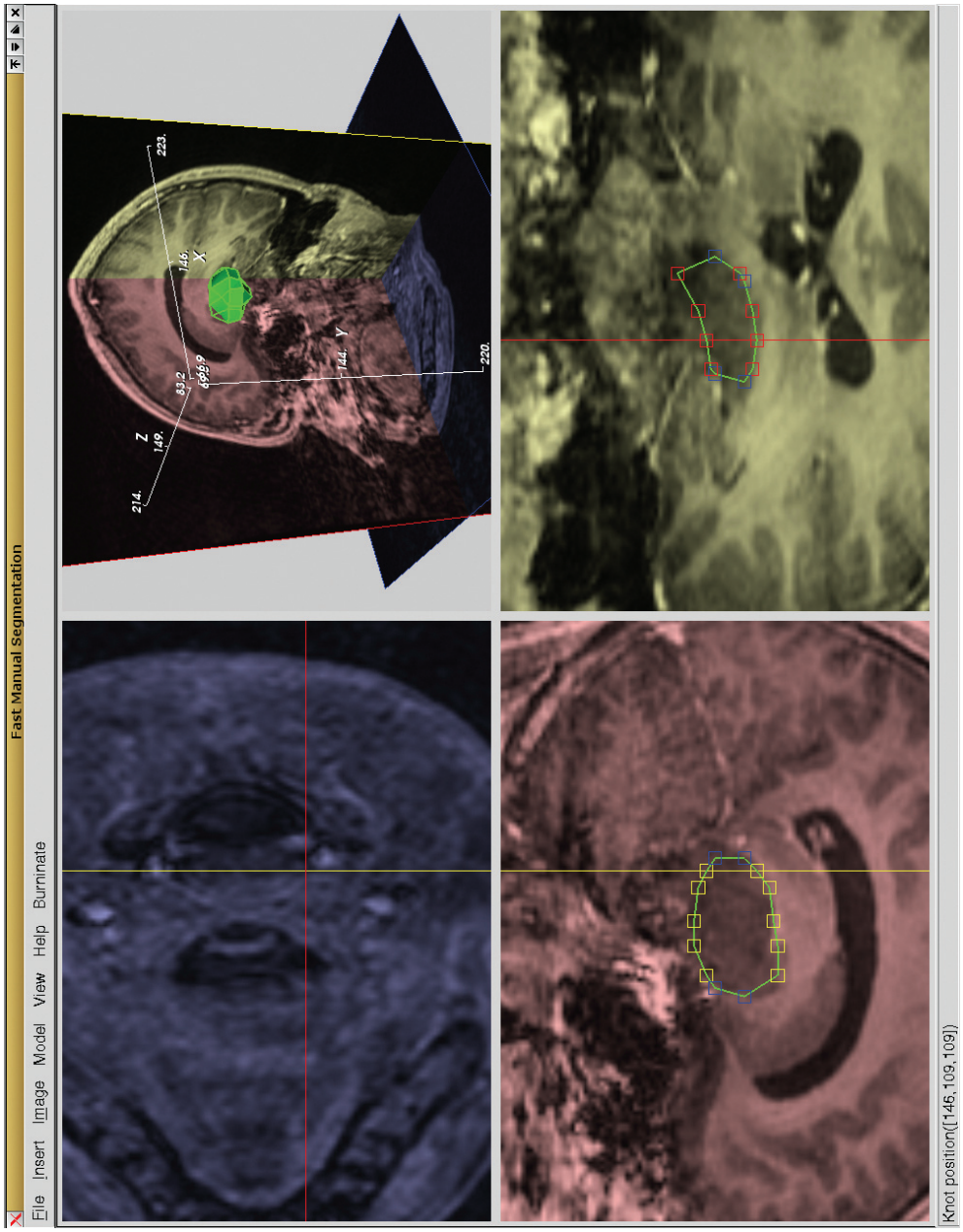


Figure 5.14: Segmentation of a tumor located in the brain (MRI dataset). The volume of the model of the tumor is 12.4cm^3.

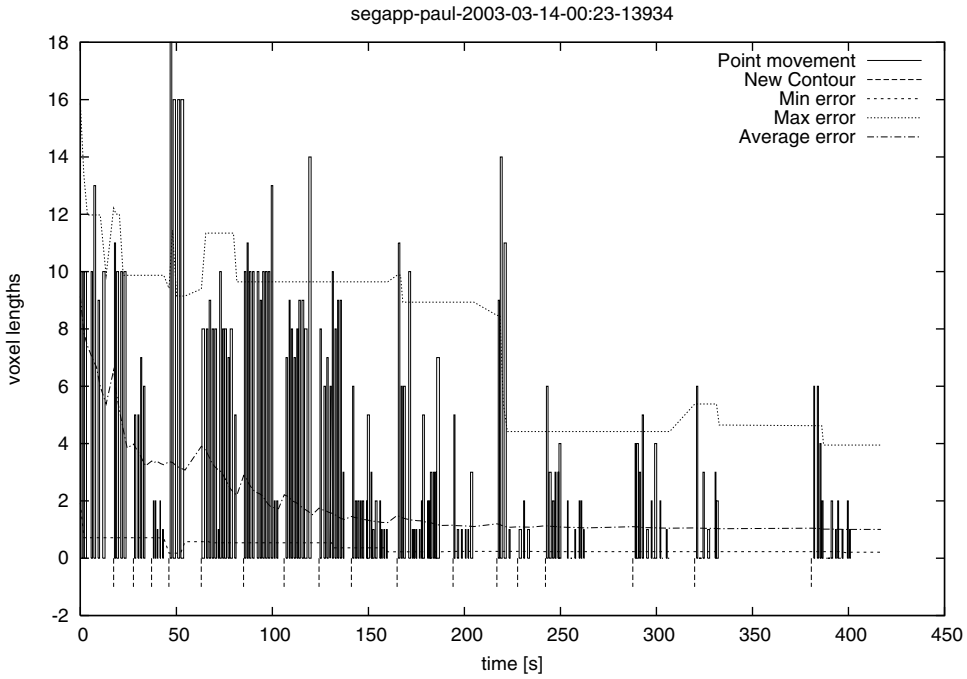


Figure 5.15: Graphical representation of a segmentation session. The bars indicate the distance a control point is moved, the dotted vertical lines indicate when a new contour is added, and the lines show the minimum, maximum and average approximation errors. The segmented object is an indented sphere, similar to Figure 5.11(b) on page 77.

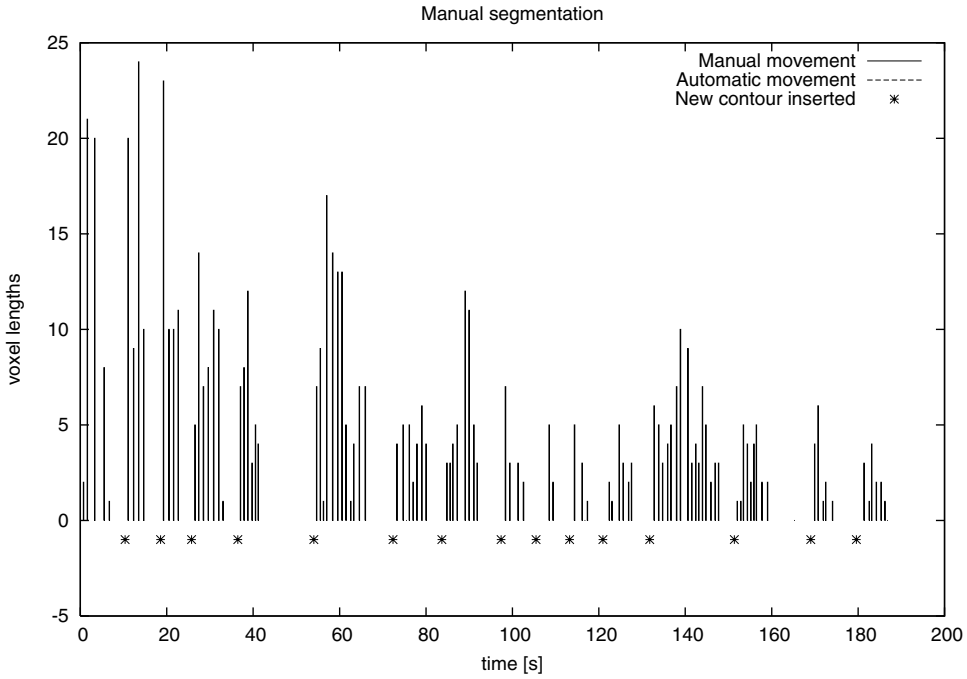
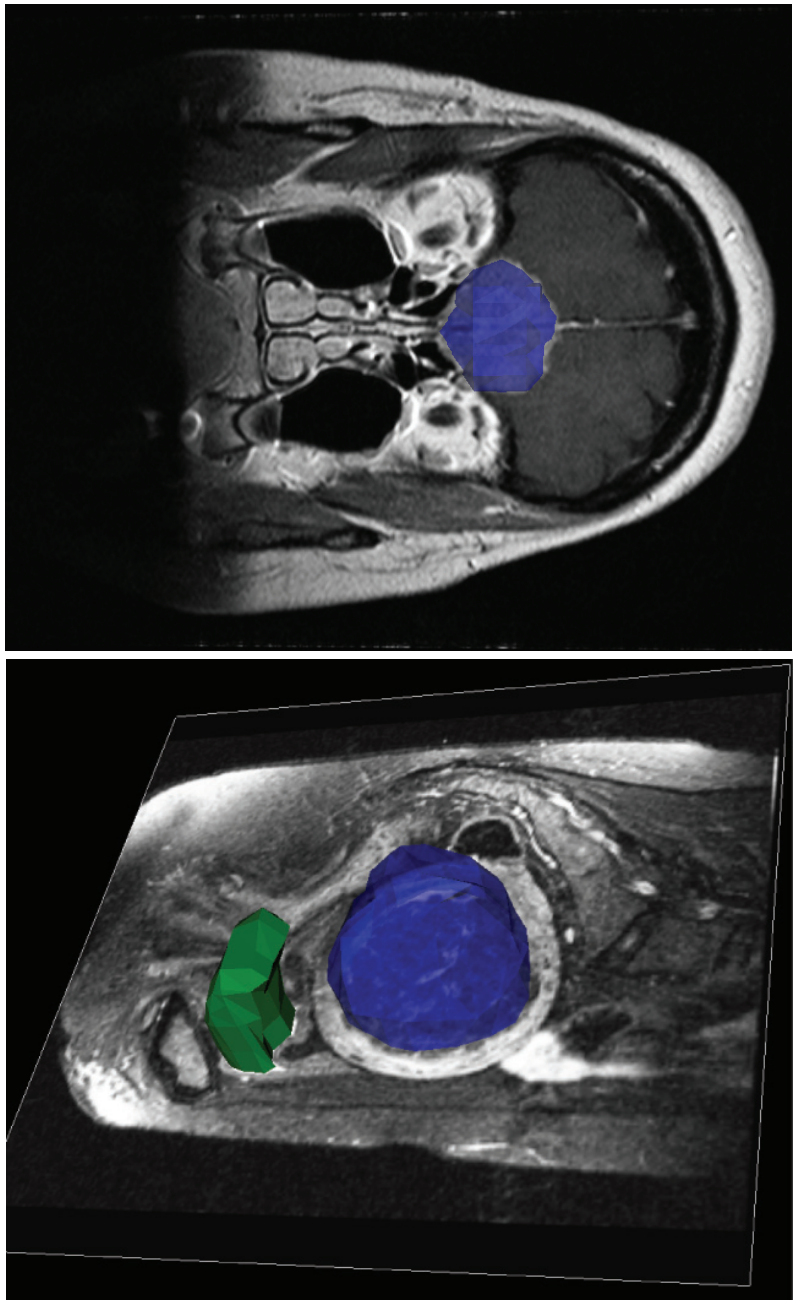


Figure 5.16: Manual segmentation of a carpal bone. A total of 104 separate manual control point relocations are performed, the total distance travelled by the control points is 545 voxels, and 15 contours have been inserted.



(a) Segmentation of a tumor located between the eyes (MRI dataset).

(b) End result of segmenting a large myoma (blue) and the bladder (green) in a CT dataset.

Figure 5.17: Manual segmentation examples.

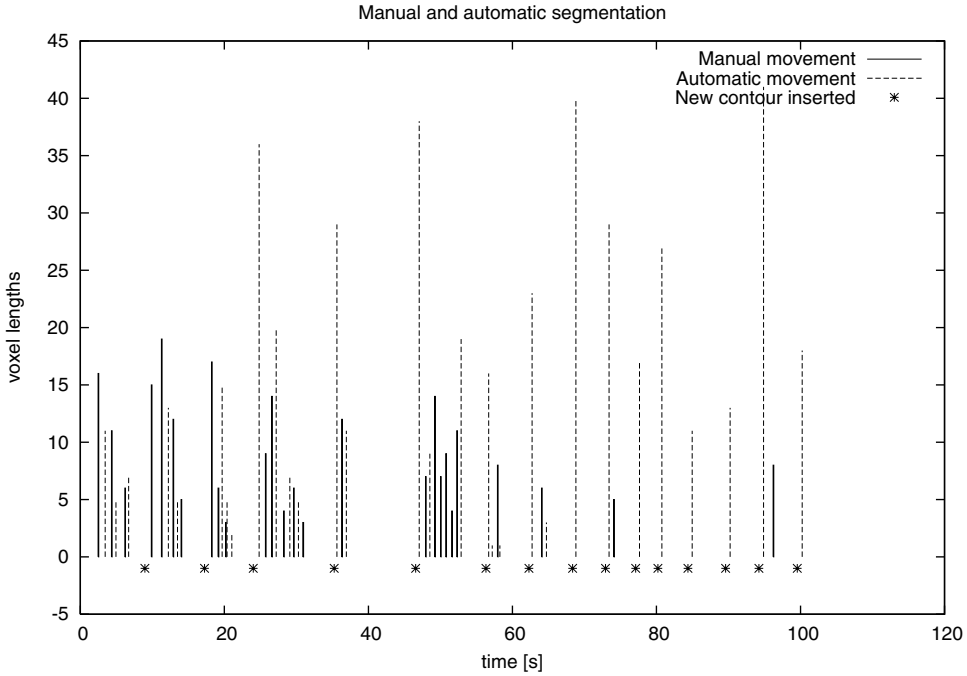


Figure 5.18: The effect of adding edge detection on the amount of manual control point movement. The amount of interaction decreases over time and is limited to a few corrections in the final stage. The vertical dashed lines indicate the *total* amount of automatic point movement per invocation. A total of 26 separate manual control point relocations are performed, the total manual distance travelled is 237 voxels, the total automatic distance is 477 voxels, and 15 contours have been inserted.

5.4 Conclusions and further research

We have described a new interactive segmentation method. Its data structure consists of a linked set of planar and orthogonal contours. Planar contours overlayed on image data are easily manipulated. User-interaction is kept to a minimum by two features. First, linked contours enable the user to model a 3D shape using 2D slices. Second, contours can be added by indicating a new intersecting plane with the object. The resulting intersection points are automatically linked according to the already present topology. The method can model objects with irregular geometry that are topologically equivalent to a sphere. The end result of the method is always free of geometrical and topological errors.

The use of colour-coding in our application assists a user to navigate and manipulate in a 3D environment. Novice users were able to segment a carpal bone in less than 10 minutes with an average accuracy less than 3 voxel lengths after a short training period of 20 minutes. The end results of the test subjects are similiar.

Augmenting the method by edge detection speeds up the segmentation process. The manual interaction using edge detection enhanced manual segmentation is limited to initialisation and corrective steering of the results of the edge detection.

Further research will focus on the following topics. First, adding template functionality, where the result of previous segmentation sessions can be reused. This allows building a library of reference objects which can be used for most segmentation tasks. Second, integration of (semi-)automatic segmentation methods to assist in relocating control points. Third, investigating the possibility of removing the orthogonality restriction. And finally, adding the possibility to model other topologies (objects with holes).

We have shown that manual segmentation using connected orthogonal contours has great advantages over conventional manual segmentation. Furthermore, effective feedback and control are provided by the method for steering semi-automatic segmentation methods.

Acknowledgement

This research is part of the MISIT (Minimally Invasive Surgery and Intervention Techniques) and the DIPEX (Development of an Improved endoProsthesis for the upper EXtremities) programmes of the Delft Interfaculty Research Center on Medical Engineering (DIOC-9). We would like to thank Mario Maas (AMC), Jaap Stoker (AMC), and the test subjects for their input and time.

Interactive matching of ultrasound and MRI for visualization during resection of myomata

6

P.W. de Bruin, F.M. Vos, F.H. Post, A.M. Vossepoel, and S. de Blok, Interactive matching of ultrasound and MR for visualization during myoma resection, *in* S. K. Mun, ed., *Proceedings of the SPIE International Symposium on Medical Imaging*, vol. 4681 - Visualization, Image-Guided Procedures and Display, 2002

Abstract

Piece-wise removal of myomata (fibroids, benign tumors) from the uterine wall is performed using a hystero-resectoscope inserted through the vagina. Only radical removal of the myoma prevents regrowth, whereas penetration of the uterine wall should be avoided.

Preoperative MR (Magnetic Resonance) images show the full outline and the location of the myoma in the uterine wall. Peroperative 3D US (ultrasound) reveals the part of the myoma that protrudes into the uterine cavity, but the part located in the uterine wall is almost invisible.

The images and extracted preoperative information must be registered to the peroperative situation. A pragmatic solution in the form of three point incremental matching is applied.

Three patients were scanned and the matching was performed on the data. The procedure provided a good match on two patients.

6.1 Introduction

Uterine myomata are the most common gynaecological pathology. A myoma is a benign tumor that grows in the uterine wall. The uterus is the female reproductive organ and is approximately $6 \times 5 \times 4$ cm^3 in size. Myomata range from a few millimeters in diameter to — in extreme cases — as large as a soccer ball. Such pathologies can cause excessive bleeding, pain and disturb pregnancy. Removal of myomata that protrude for more than 50% of their volume into the uterine cavity is usually performed via a minimally invasive procedure: a hystero-resectoscope is inserted into the uterus via the vagina and the myoma is removed piece-wise using a loop-wire electrode [16]. In Figure 6.1 several stages of the myoma removal process are shown.

From preoperative MR (Magnetic Resonance) images the full outline and the location of the myoma in the uterine wall can usually be assessed. Figure 6.2(a) on page 90 shows an example of an MR image of a uterus with a myoma. Peroperative US (ultrasound) reveals the part of the myoma that protrudes into the uterine cavity, but the part located *in* the uterine wall is difficult to see, if not invisible. Figure 6.2(b) on page 90 shows an example of a US image of a uterus with a myoma.

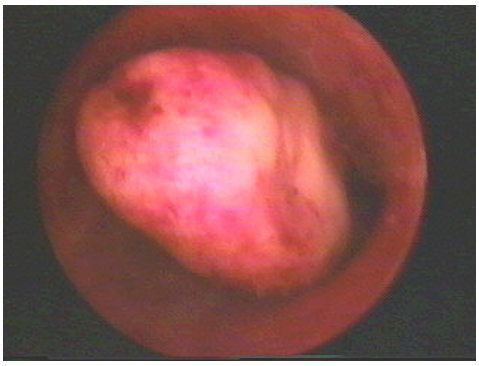
For several reasons the uterus is distended ("inflated") with salinated water during surgery: firstly, to be able to see through the resectoscope and secondly, as a medium for the removal of cut debris that would otherwise impair sight. Distension is dangerous because the distension fluid is absorbed into the bloodstream. If too much water is absorbed, the patient is effectively drowned.

Through the hystero-resectoscope and using peroperative abdominal US the surgeon is unable to see the part of the myoma that is located *in* the uterine wall (see Figure 6.1). The loop-wire electrode burns the cut tissue slightly, thereby making it difficult to discriminate between uterus and myoma based on tissue texture and color (see Figures 6.1(c) and 6.1(d)).

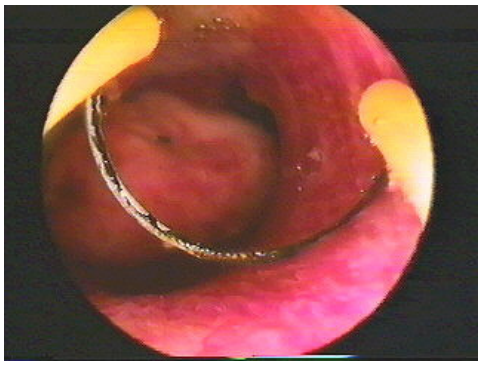
For complete recovery of the patient it is necessary to remove a sufficiently large part of the myoma. If too little is removed the myoma will regrow. On the other hand, removing too much risks penetrating the uterine wall.

Resection of myomata could be improved by giving the surgeon a view of the myoma that cannot be seen through the resectoscope or on (3D) US [37]. Specifically, information about how much of the myoma is removed is vital during surgery.

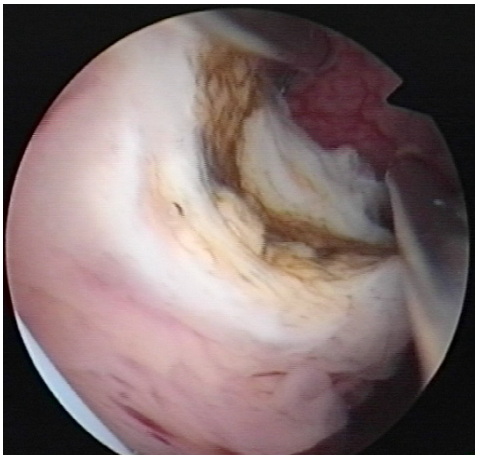
The goal of this work is to explore a method that "fits" a myoma extracted from preoperative MR into the peroperative (3D) US image. In order to perform this fit, a simple model of the uterus with myoma is registered with distinct landmarks and the visible part of the myoma in the US image. In this paper the first steps are taken to show that this is a feasible approach.



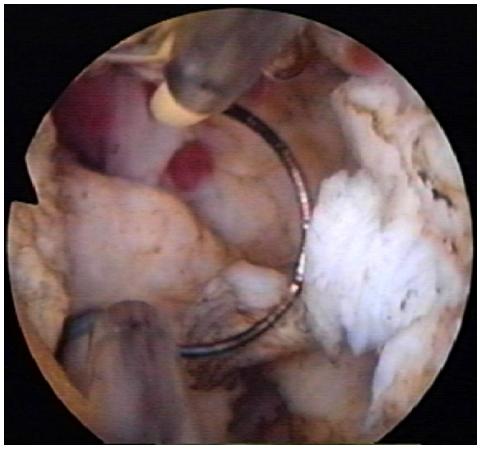
(a) A myoma seen through a resectoscope



(b) The loop electrode mounted on the resectoscope.

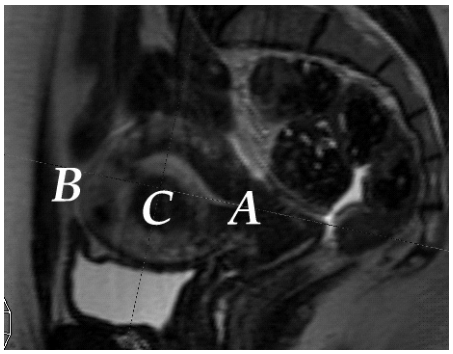


(c) After one cut with the electrode through the myoma.

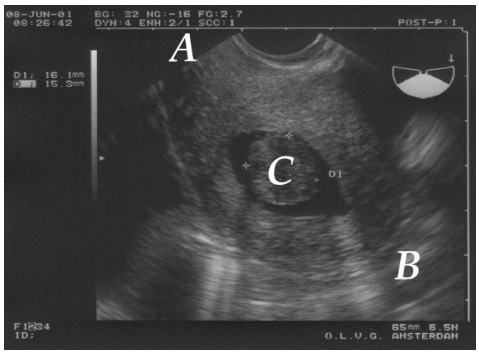


(d) After several cuts with the electrode.

Figure 6.1: Overview of myoma removal process (images obtained during a resection peformed by Sjoerd de Blok at the Onze Lieve Vrouwe Gasthuis).



(a) MR image (sagittal view) of a uterus with myoma.



(b) US image of distended uterus with myoma.

Figure 6.2: Example images of MR and US modalities. The indicated points correspond to: A – between corpus and cervix, B – top of the uterus and C – center of the myoma.

6.2 Techniques

6.2.1 Imaging modalities

MR can be used for preoperative assessment of the location and size of a myoma. The full outline and location of a myoma is clearly visible in an MR image (see Figure 6.2(a)). Using standard MR equipment (1.5T Philips Gyroscan NT) and a T2 weighted sequence, the myoma occupies approximately 25 voxels in the volume. The MR datasets have a spatial resolution of $1.175 \times 1.175 \times 4.0$ mm, and 12 bits in grey value. The volumes are $256 \times 256 \times 32$. During surgery, images are acquired using 3D ultrasound equipment (Kretz SA9900). The US voxels vary in size, but each B-scan is resampled at 0.35 mm^2. The data resolution is 7 bits. A typical volume size is $258 \times 216 \times 99$. Conventional 2D US images a single plane where 3D US can scan a set of planes by rotating the head of the probe. The volume is stored as a set of 2D planes where for each plane the acquisition angle is noted. This volume is a structured curvilinear grid which involves more complex processing than the regular orthogonal grid on which the MR data is defined.

Using 3D US the entire uterus can be imaged in about 20 seconds during which the patient must lie still. Averaging of several scans is usually performed to reduce noise in the scanned image. Due to the limited scanning time this averaging is not performed with 3D US, resulting in noisier images.

6.2.2 Matching

There is always an amount of time between acquisition of the MR data and the surgery. During this period, the location, orientation and shape of the uterus will not remain constant. Therefore, the images and extracted information from MR cannot be used without adapting it to the peroperative situation.

Deformation and orientation of the uterus is caused by the degree to which the bladder and colon are filled and the normal peristaltic movement of the organs in the abdomen. Also, the uterus is a muscle and can contract and expand.

If the patient is fasted before the MR scan, then the colon is in the same condition as during surgery. The bladder is filled completely during surgery to create a window for the ultrasound to look through. This situation is duplicated by asking the patient to drink a sufficient amount before MR scanning. Gravity can be ruled out as a factor because the position of the patient during MR and surgery is very similar.

Straightforward techniques [84] are prone to fail due to the noise in the MR and US datasets, the limited amount of matching information and due to the fact that the abdomen is not a rigid section of the human body. A pragmatic solution to the matching problem is a simple three point incremental matching. Here, the difficulty is defining corresponding landmarks in the two modalities. A landmark should be easy to locate in the images and it should be relocatable with a high degree of repeatability. Based on these criteria the following landmarks are selected for matching:

A Bending point between corpus and cervix;

B The top of the uterus;

C The center of the myoma.

The line A–B is the main axis of the uterus. The line from C perpendicular to the line A–B is the secundary axes. Distention can be described by scaling along these axes.

6.2.3 Extraction of landmarks

The extraction of landmarks from MR data is a user-guided process. In this approach a combination of seeded region-growing and the experience of a knowledgeable user is applied. The myomata are reasonably well distinguishable and region-growing is suitable for extraction of the outline of the myoma.

On the US images the exact outline of the myoma cannot be distinguished. The inner wall of the uterus is clearly visible due to the fact that the uterus is filled with water. The outer wall is almost invisible in the region farthest from the US probe.

Therefore, for the US images we rely on an expert to indicate the center of the myoma in the image.

The cervical canal is located by the hysteroscope that is inserted into the uterus. The hysteroscope is a metal instrument that renders a clear echo on the US image. For locating the top of the uterus we rely on an expert to indicate its position.

6.2.4 Incremental three point matching

The uterus consists of two sections: the cervix and the corpus. The cervix comprises approximately one third of the length of the uterus and connects the corpus to the vagina. The pear-shaped corpus occupies the remaining two thirds of the uterus.

Based on the experience of gynecologists the following assumptions about deformation of the uterus are made. The assumption for matching is that the movement of the corpus is confined to rigid-body rotation about a fixed point — the bending point between corpus and cervix (point A). Due to the deformation a straightforward three point matching is not possible. A least-squares approach will yield a low quality match because it minimizes the error at all three points. This is not desirable because the certainty at each point is not equal.

In our approach the point of rotation (point A) is translated first (see Figure 6.3). By doing so the points of rotation in each modality are aligned. Next, the top of the uterus (point B) in the US volume is rotated to align with the top of the uterus in the MR volume. Finally, the center of the myoma (point C) is rotated about the cervix–top axis to align.

The procedure of landmark indication is performed interactively using a small application written in OpenDX [82]. From the six landmarks the transformation matrix is calculated by compositing the translation and the two rotations described above.

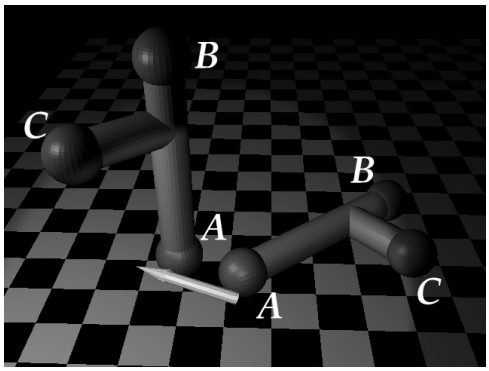
6.2.5 Visualization

Once the matching is complete the results are presented to the surgeon. From the preoperative MR the surface of the myoma is extracted. This surface can now be placed in the correct location in the US images using the calculated transform.

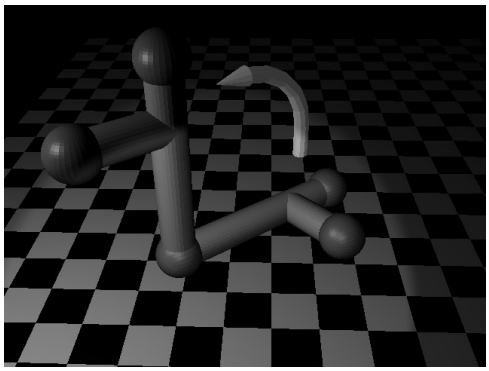
Arbitrary slices through each volume can be made to visualize the alignment at the intersection of the slices as is shown in Figures 6.4 and 6.5.

6.3 Results

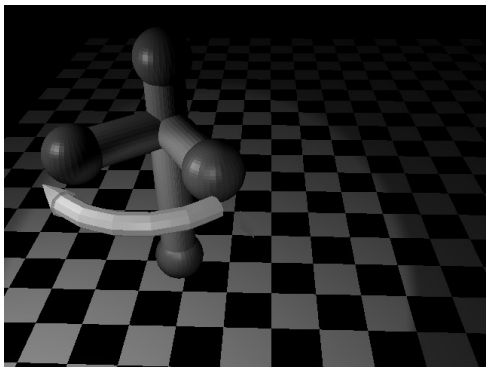
Three patients who were selected for removal of myomata were scanned. Between acquisition of MR data and surgery was a maximum time of two weeks.



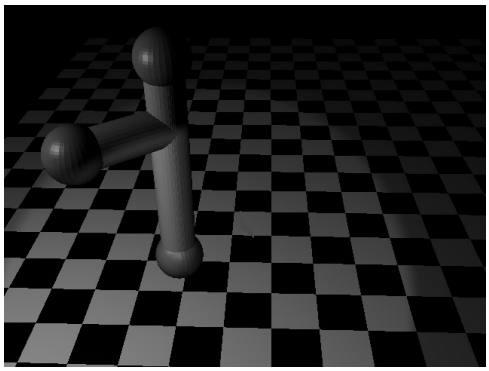
(a) Translation of the cervix.



(b) Rotation about the cervix to align the top.



(c) Rotation about the cervix–top axis to align the myoma.



(d) Final situation.

Figure 6.3: Overview of the incremental matching method. The stick figures represent the two imaging modalities. The three spheres on each stick figure are the landmarks to line up (see also Figure 6.1 on page 90)

Each of the patients had at least one myoma that was resectable using a minimally invasive procedure. On all three datasets the matching procedure was applied. On two of the three patients a visual alignment was achieved using the procedure. Figures 6.4 and 6.5 show the alignment by moving the US dataset through the MR dataset for two different patients. Figure 6.6 shows an ultrasound slice with the extracted meshes from the MR data.

In contrast to the first two patients, the bladder of the third patient was only half full during MR and completely full during surgery. The difference caused a change in deformation of the uterus that was too large to perform a match.

6.4 Conclusions and future work

Judging by the resultant visual alignment of the MR and US datasets the procedure is promising: two of the three patients could be matched. The third patient could not be matched as a result of a too large difference in bladder volume. This is avoided easily in the future. More work is necessary to evaluate the quality of the matching procedure.

Fully automatic US landmark extraction is difficult. However, we have found that this is not required, because it is a simple task for the gynaecologist to indicate the three landmarks.

Future research includes a thorough evaluation of the method using more data, minimizing user input and exploring various ways of real-time peroperative visualization.

Acknowledgements

This research is part of the MISIT (Minimally Invasive Surgery and Intervention Techniques) programme of the Delft Interfaculty Research Center on Medical Engineering (DIOC-9).

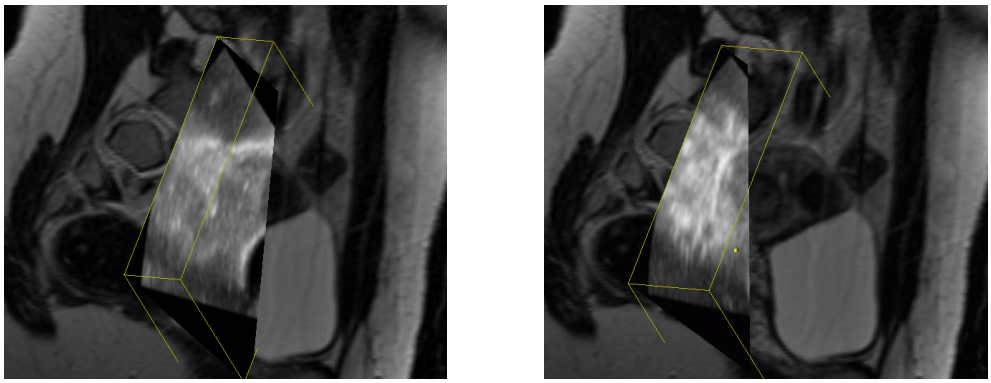


Figure 6.4: Matched MR and US of patient A. Two different slices of the US dataset through the MR volume are shown left and right.

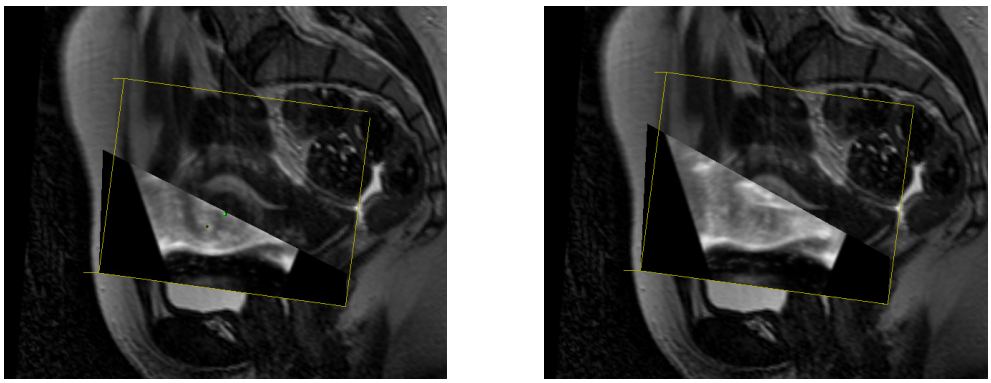


Figure 6.5: Matched MR and US of patient B. Two different slices of the US dataset through the MR volume are shown left and right.

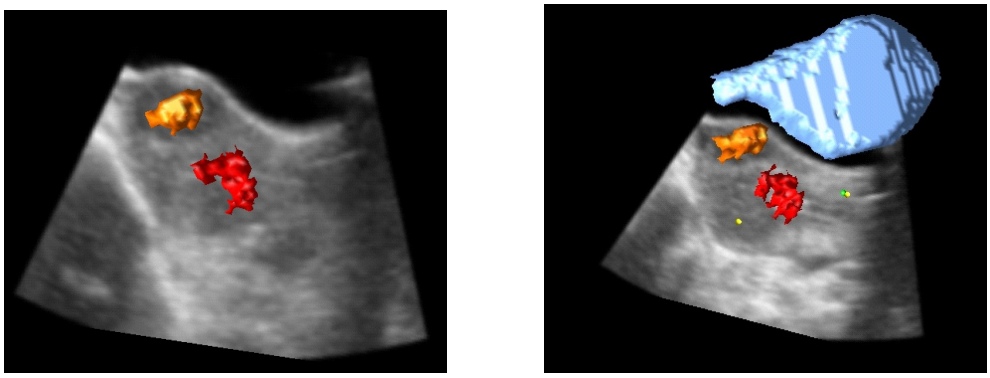


Figure 6.6: Example of visualization of meshes obtained from MR data in the ultrasound data. The meshes are myomata and the bladder.

A statistical shape model without using landmarks

7

F.M. Vos, P.W. de Bruin, J.M. Aubel, G.J. Streekstra, M. Maas, L.J. van Vliet, and A.M. Vossepoel, A statistical shape model without using landmarks, *in:* J. Kittler, M. Petrou, and M. Nixon, eds., *ICPR17, 17th International Conference on Pattern Recognition, Cambridge, UK, Aug. 23-26*. IEEE Computer Society Press, Los Alamitos, vol. 3, 2004 pp. 714–717.

Abstract

This paper describes the construction of a statistical shape model based on the iterative closest point algorithm. The method does not require manual nor automatic identification of explicit landmarks on example shapes. Corresponding features are found by retrieving the nearest points via interpolation along the surface. The application to analyse carpal bone shape renders evidence that the lunate bone occurs in distinct shapes.

Figure 7.1: X-ray image showing the left hand wrist in dorsal view. The carpal bones constitute the joint between the metacarpals and the ulna and radius.

7.1 Introduction

Statistical shape models have proven to be useful tools to study variation in anatomical shapes. A popular method captures shape by a sampled 3D point distribution model (PDM) [38]. These models are often created by manually indicating characteristic points on example shapes. However, indicating features by hand is tedious and prone to error. Moreover, many objects have only a few landmarks or have features that are not easily identifiable.

The carpal bones serve as a good example (see Figure 7.1). Physicians are interested in the shape of carpal bones, because a relation is expected to exist between bone shape and pathological wrist kinematics. Although the shape of each carpal bone is highly characteristic, it is not easy to identify characteristic points.

Preliminary medical studies report on shape features of the carpals in conventional X-ray images [112, 119]. The results indicate that there may be distinct types

of the lunate bone. Although it confirms observations from clinical practice, the analysis is limited due to the projective nature of the X-ray images.

Recently, several papers describe methods to analyze such shapes via 3D statistical shape models (see e.g., [23, 38, 40, 48, 107]. Although the outcomes are very promising, there is not yet a generally accepted framework.

The objective of this work is to construct a non-landmark based statistical shape model. Inspired by [23], an alternative way to find correspondences on multiple shapes is proposed (see Section 7.2.2). The method is applicable to any problem in which explicit shape features are not easily identifiable. The novelty is the adaptation of an iterative closest point algorithm and the application to analyse carpal bone shape.

7.2 Methods

We initially follow the procedure first described by Cootes and Taylor [38]. Shape vectors $\boldsymbol{x}_i$ are constructed of n characteristic points in each training image i. Thus, a d-dimensional image results in an nd-element shape vector. Translation, rotation and scale invariance is imposed by aligning the centers of gravity and minimizing the Euclidean distance between the shape vectors under rotation and scaling (Procrustes analysis).

The main shape variations are found by principal component analysis, under the assumption of Gaussian distributed data:

1. The mean vector $\boldsymbol{x}$ and covariance matrix $\boldsymbol{X}$ are calculated by

$$\bar{\boldsymbol{x}} = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i \quad X = \frac{1}{m-1}\sum_{i=1}^{m}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T \tag{7.1}$$

   in which m indicates the number of training shapes.

2. The eigenvectors ϕ_i and corresponding eigenvalues λ_i are computed and sorted so that $\lambda_i \geq \lambda_{i+1}$.

Let $\Phi = (\phi_1, \phi_2, \ldots, \phi_k)$ be the matrix of eigenvectors corresponding to the k largest eigenvalues (e.g., summing to 95% of total variance). Shape ($\boldsymbol{x}$) is modelled by:

$$\boldsymbol{x} = \bar{\boldsymbol{x}} + \Phi\boldsymbol{b} \tag{7.2}$$

in which $\boldsymbol{b} = (b_1, b_2, \ldots, b_k)$ is a vector containing the shape parameters. Clearly, $\boldsymbol{b} = \boldsymbol{0}$ yields the average shape. The shape variation along axis i is generated by varying b_i. Suitable limits are: $|b_i| \leq \pm 2\sqrt{\lambda_i}$ $(= 2\sqrt{\sigma_i})$ (σ_i indicates the variance along axis i).

At the basis of the analysis are corresponding points. A technique to find correspondences on two point sets is introduced next (Section 7.2.1). How to convey this method to a whole set of samples is described in Section 7.2.2.

7.2.1 Correspondence method

The iterative closest point (ICP) algorithm maps one point set (A) onto another (B) [13, 23] as follows:

1. Find the closest points to A on B: B'
2. Find the closest points to B on A: A'
3. Find the transformation Q that matches A with B' and B with A'
4. Transform all points via Q
5. Loop until converged

The transformation Q of a point $\boldsymbol{p}$ is defined by $Q(\boldsymbol{p}) = sR\boldsymbol{p} + \boldsymbol{t}$, in which s is a scale factor (scalar), R is a rotation matrix and $\boldsymbol{t}$ is a translation vector. An initial estimate for Q is obtained by registering the centers of gravity of the pointsets, aligning the first order moment vectors and equalizing the average distance of the points to the center.

The best match is found in step 3) by minimizing a symmetric distance measure on the point sets:

$$E^2 = \frac{1}{n_A}\sum_{i=1}^{n_A} |Q(A) - Q^{-1}(B')|^2 + \frac{1}{n_B}\sum_{i=1}^{n_B} |Q(A') - Q^{-1}(B)|^2 \tag{7.3}$$

Notice that the shapes are transformed into an intermediate frame, rather than transforming one shape onto the other. This way, preferential treatment of one of the shapes is avoided.

At last, points are mapped from one set on the other by finding the closest points.

7.2.2 Model construction

The previous method needs to be adjusted to find corresponding points in a whole set of examples. We assume the existence of segmented sample shapes represented by triangle meshes. An accurate surface representation is asserted, but the vertices may not correspond with landmarks. Additionally, mesh size and topology may vary. The meshes are to be "resampled" on corresponding positions to perform shape analysis

(Section 7.2, introduction). We implemented an iterative procedure based on the closest point algorithm (Section 7.2.1) to do so.

Initially, the mesh with the largest number of vertices ($n_{\max}$) is selected (the pivot mesh). Next, the other meshes are matched to the selected one. At that stage all meshes are resampled to have the same number of corresponding points ($n_{\max}$). Also, the resampled meshes inherit the topology of the pivot mesh. The mean shape is computed by averaging the resampled vertex positions and the whole procedure is repeated by matching the original meshes to this average shape. Thus, a preferential treatment of the mesh with the most vertices is avoided.

One should notice that the Procrustes analysis is embedded in the correspondence method (via Q). Consequently, systematic errors due to an initial misalignment are avoided.

7.3 Results

All 3D-CT images of the wrist over a period of exactly two years were collected from the Academic Medical Center Amsterdam. This resulted in wrist images from 134 patients, containing 103 right wrists and 107 left wrists From this set only those wrists were included that did not show signs of pathology according to a radiologist. Among the exclusion criteria were: fractures, arthritis, presence of bone fragments, and bone fusions.

Our focus was on the shape of the lunate and capitate bones since these are typically involved in pathologic wrist kinematics. Unfortunately, the entire wrist was not contained in each scan. Applying the exclusion criteria and considering only completely scanned carpal bones left us with 41 examples of the left lunate, 31 right lunates, 36 left capitate bones and 25 right capitates.

7.3.1 Model construction

A specially developed software tool [25] was employed to segment the sample shapes from the images. The output of the tool was a triangulated surface description. These meshes were aligned as described in (Section 7.2.2) and principal component analysis was performed on corresponding points (Section 7.2, introduction). Both picking the closest points (point-to-point) and interpolating to the nearest position on the triangles (point-to-surface) were considered to find the correspondences (Section 7.2.1).

Figure 7.2 on the next page illustrates the effect of both approaches on the cumulative relative variance for the left lunate bone (the other carpals had similar outcome). For instance, 80% of the total variance is explained by 23 eigenmodes using point-to-point correspondence and by only 10 eigenmodes via the point-to-surface method.

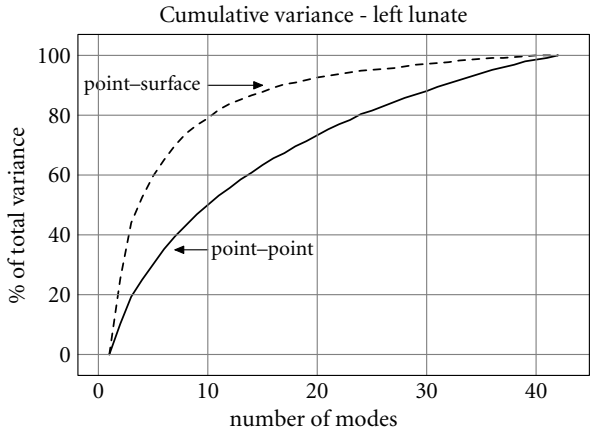


Figure 7.2: Cumulative relative variance of the left lunate. Variance calculated based on point-point correspondence is indicated by the solid line and point-surface correspondence by the dashed line.

Table 7.1: Mean distance between corresponding points.

Correspondence method	Mean E^2 (mm)
Point-point	1.9
Point-surface	0.9

The flat profile indicates that the shape analysis via a point-to-point correspondence is hampered by noise (Figure 7.2, left). This must be due to the irregular positioning of the vertices. The point-to-surface graph signifies an improvement on correspondence. Consequently, true shape description dominates over position errors (Figure 7.2, right). This is confirmed by the smaller mean distance between corresponding points over all shapes (mean E^2, cf. Equation 7.3 on page 100, see Table 7.3.1). Our further analysis employs point-to-surface correspondence.

The stability of the eigenvectors was assessed by their orientation and size as the sample size increased. The first 10, 20 and 30 shapes were selected from the complete set of (41) left lunate bones. Table 7.2 on the facing page shows the angles between the 4 largest eigenvectors calculated from the subsets and the ones determined from the full set. The table illustrates that the two largest eigenvectors tend to stablize with 30 samples. Higher order eigenvectors seem not to correspond which is indicated by the angles that approximate 0.5π radians.

Figure 7.3 on the next page illustrates the effect of sample size on the cumulative relative variance.

From Table 7.2 on the facing page and Figure 7.3 on the next page we concluded

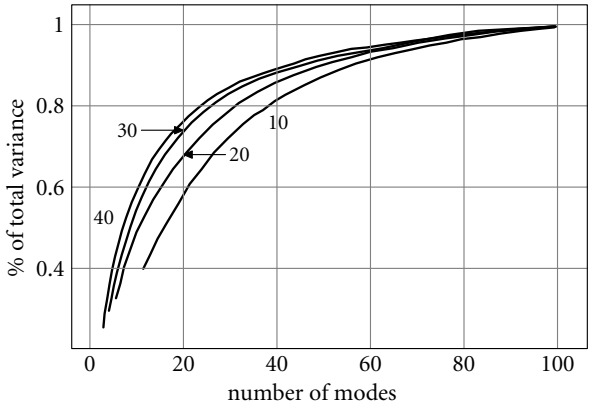


Figure 7.3: Cumulative relative variance for 10, 20, 30 and the full set of 40 samples.

Table 7.2: Angle between eigenvectors from smaller subsets and full set of samples (radians)

#sample shapes	eigenvector			
	1	2	3	4
10	.72	.68	1.2	1.6
20	.35	.38	1.6	1.8
30	.27	.28	.53	2.1

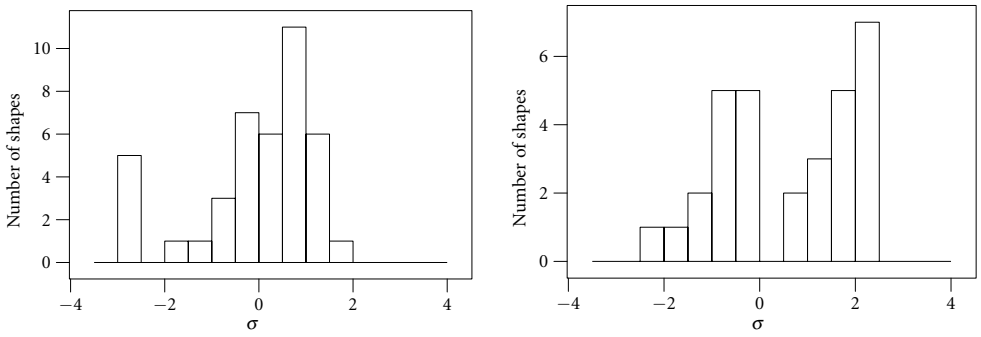
that the point-surface method enables a proper estimation of the first 2 eigenvectors.

7.3.2 Analysis

Earlier work indicates that the lunate bone appears in distinct types, implying multimodally distributed data. Our shape analysis, however, assumes Gaussian distributed data. The validity of this assumption was assessed by the projections of each lunate shape ($\boldsymbol{x}_i$) on the first eigenvector ϕ_1: Figure 7.4 on the following page

Clearly, the right lunate yields a bimodal distribution which confirms the existence of distinct shapes. Accordingly, our further analysis considered the two lunate types separately. The left lunate histogram does not justify the distinction. One might attribute this to small sample size. The outliers correspond to bones that have sharp protrusions. The other eigenmodes and the other bones did not deviate significantly from a Gaussian distribution.

The mean right carpal shapes (Figure 7.5 on the next page) are realistic representations that agree with how the bones are perceived by clinicians. The left drawing



(a) Left lunate (b) Right lunate

Figure 7.4: Histograms of lunate shapes projected on the first eigenvector.

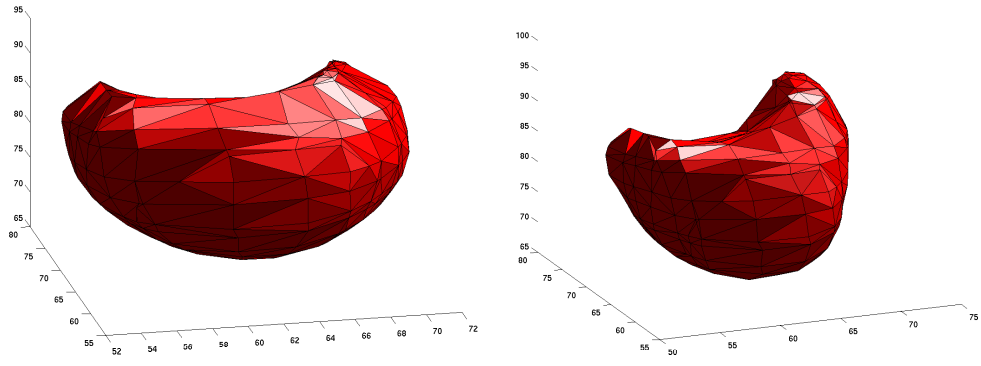


Figure 7.5: Two distinct right lunate shapes.

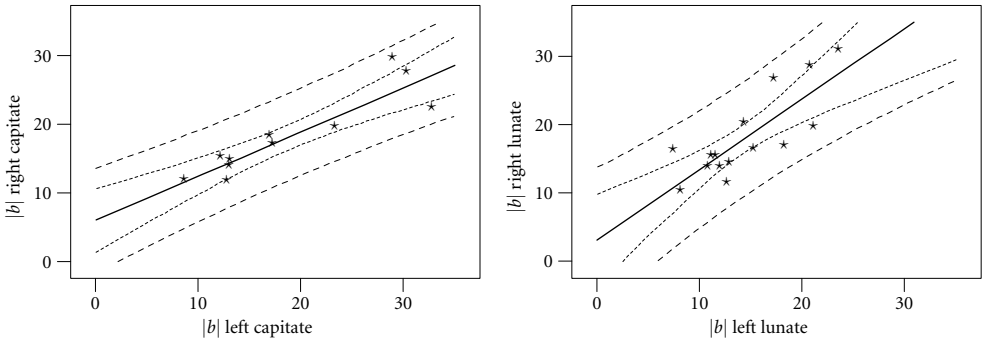


Figure 7.6: Comparison of carpal shapes in left and right wrist. The outer dashed lines indicate the 95% prediction interval, the inner dashed lines indicate the 95% confidence interval for the solid regression line.

corresponds to the left group in the histogram of Figure 7.5 on the facing page, the right shape with the right group. The two types of the lunate bone conform to the description of type I and type II lunates [112, 119].

The relation between different shapes was explored by way of the shape parameters (b, cf. Equation 7.2 on page 99). The graphs in Figure 7.6 plot $|\boldsymbol{b}|$ for the left versus the right lunate and capitate bones, respectively. The shaded area is the 95% prediction interval for the data computed from linear regression. The dashed lines represent the 95% confidence interval for the solid red regression line. Both graphs indicate that a deviation from the average shape in one hand correlates with a deviation in the other. Other relations (e.g. left lunate vs. left capitate, left lunate vs. age etc.) did not yield significant correlations.

7.4 Conclusion

We described a statistical shape model using an iterative closest point algorithm. The algorithm does not require manual identification of landmarks. A correspondence of features was established by retrieving nearest points on surfaces via interpolation. Such correspondence facilitated the construction of a precise shape model. The shape analysis of the carpal bones rendered further evidence that there are distinct lunate bone shapes. However, a final proof will require more example shapes. For further research, it would be interesting to correlate bone shape with left or right hand dominance as well as pathological kinematics.

We conclude that our approach provides a powerful tool, which adds to current methods for statistical shape analysis in the absence of natural landmarks.

Conclusions and future work 8

8.1 Conclusions

A common approach to visualisation of 3D medical data is to extract surfaces that describe the boundaries of anatomical objects. Such surfaces provide an interface for manipulation, navigation, modelling, and visualisation and help a clinician with diagnosis, treatment planning, and surgical simulation. For all these applications two requirements are important: accuracy and quality. Accuracy describes the conformance of a mesh to an object boundary, whereas quality relates to the shape of the triangles in the surface mesh. The quality of a triangle is usually defined by how much it resembles an equilateral triangle. The presence of elongated or degenerated triangles is undesirable in surgical simulation where these can cause errors in numerical solving methods (finite elements). In this thesis a set of methods and tools is presented to create accurate *and* high-quality surface meshes from volumetric medical datasets.

Chapter 2 gives an overview of surface extraction and mesh improvement techniques. A vast body of literature exists on both topics, and the overview is as a result limited to iso-surfacing, edge-detection, and Laplacian smoothing variants. The *integration* of extraction and improvement is not often found in the literature. Usually, mesh improvement techniques are applied *after* extraction of the surface. Although separate mesh improvement methods try to keep e.g., surface area or object volume constant, the relation between the mesh and the data from which it was generated is lost.

Iso-surfacing is the most common surface extraction technique. The SurfaceNets method, described in Chapter 3, combines iso-surface extraction with mesh smoothing. Initially, the SurfaceNets method was proposed to create smooth surface representations from binary data. In this work the basic principle is extended to the extraction of surfaces with smoothly varying triangles from grey-scale volume data. In order to do so, a mechanism is developed that alternately applies smoothing and surface relocation steps, which improves the quality of the mesh and maintains accuracy by ensuring that each node is located on the iso-surface. The meshes created by the SurfaceNets method are equal in accuracy and superior in triangle quality compared to the Marching Cubes iso-surface extraction algorithm. The improved meshes

contain less low quality and more high quality triangles. However, the application of an iso-surface method is limited to data that can be segmented by thresholding. For example, noisy data or bias-fields in MRI may prevent the generation of a suitable surface mesh using thresholding.

The ideas developed in Chapter 3 are taken a step further in Chapter 4 where a scale-space edge-detection method is applied for surface extraction. In order to prevent misdetection of edges and to reduce the effects of noise and the noise-enhancing effects of gradient filters, the scale-space approach is adopted. Particular consideration was given to the objectivity of the method and this has resulted in a minimal number of parameters that have to be tuned. The only requirements are an initial mesh and an estimate of the maximum deviation between this mesh and the actual object. Alternating edge-detection and mesh improvement steps proved to be a successful approach. By doing so, it is possible to fulfil both the accuracy and the mesh quality requirements. The accuracy is determined by applying the method to simulated data and was confirmed to lie within the range of inaccuracy of the edge detector. The method was applied to several medical datasets and quality of each initial mesh was improved greatly. Although we cannot claim that geometric errors are prevented completely, we found that our meshes rarely contain such an error.

The edge-detection method requires an initial mesh, and contrary to the iso-surface SurfaceNets method, this mesh cannot be created by thresholding. Chapter 5 describes a new interactive segmentation paradigm that is well-suited to generate such an initial mesh. Conventional "manual" segmentation methods operate by delineating a contour in each image of a stack of images. A 3D model is reconstructed by connecting the set of contours. The new fast interactive segmentation method circumvents the problems related to contour connection by operating directly in 3D. Starting with a template sphere that consists of three planar, connected, and orthogonal contours, a desired shape is created by moving contour points. The intersections of the object with orthogonal slices are linked automatically to create new contours. Effectively, the user can create contours automatically in three orthogonal directions and because the contours are linked the amount of "tedious" interaction is kept to a minimum. The combination of a linked structure and planar contours makes editing each contour a simple task and the effect of each operation is seen immediately on *each* affected contour. The final result is a mesh that is guaranteed to be free of topological and geometric errors. Apart from fully interactive segmentation, the method provides effective feedback and control for steering semi-automatic methods. Furthermore, after the addition of 2D edge detection, the amount of required interaction was reduced significantly; user actions are limited to initialisation, contour plane insertion and corrective repositioning of contour vertices. During the entire process the user has full interactive control over the location of *all* contour points. Experiments

show that novice users can work with the segmentation method after a short learning period. Finally, a color-coding scheme was used in the segmentation application that proved to be an effective aid for orientation and navigation during segmentation.

Chapter 6 contains an initial exploration of the possibilities of matching pre-operative MRI to per-operative 3D ultrasound images of the uterus. The goal is to aid a surgeon with the removal of fibroids from the uterus by providing real-time information about the location and (remaining) size of the tumor. Automatic matching is very difficult because the uterus is highly flexible and the 3D ultrasound images are very noisy. A simple method, consisting of landmark-based matching and rigid motion, is proposed, where the landmarks are chosen with the expected deformation in mind. The results show that although the uterus is oriented differently, the alignment appears to be visually correct.

Chapter 7 describes the construction of a statistical shape model of two carpal bones. Such a model provides a means to describe carpal bones as a weighted sum of principal shapes, and is useful for shape analysis and segmentation of unseen shapes. The method does not require manual identification of corresponding landmarks on the input shapes. Correspondences are established automatically by locating nearest points on surfaces using interpolation. Analysis of the shapes results in further evidence that there are two distinct shapes of the lunate bone. Our fast interactive segmentation method proved to be a valuable tool to create the input meshes from wrist CT data.

What sets this work apart from the large body of literature on surface extraction and mesh improvement techniques is the combination of edge detection/iso-surfacing and mesh improvement. Many mesh optimisation techniques operate by keeping certain measures of the mesh constant, for example volume or surface area. In this work this assumption is not made and the mesh is allowed to deform freely. A good trade-off between accuracy and high-quality meshes is achieved by our method of alternatingly applying surface extraction and mesh improvement techniques.

8.2 Future work

8.2.1 Surface extraction and improvement

Directions for research for the surface extraction and improvement methods presented in Chapters 3 and 4 are:

- other mesh improvement schemes;
- control of the trade-off between accuracy and quality;
- better control of interactive mesh coarsening and refinement;

- performance.

Not only is the current body of literature on mesh improvement techniques very large, new methods are published on a regular basis. This work has limited itself to variants of Laplacian smoothing and edge swapping. However, newer methods, such as anisotropic diffusion [42, 104] are well worth investigating. While our focus was on high accuracy, the methods could be extended to favour surface smoothness and triangle quality within an adjustable range of (in)accuracy. Finally, an obvious improvement is to provide the ability to adaptively and interactively apply mesh refinement and coarsening. By doing so, potentially large models can be reduced in size without sacrificing accuracy. For example, the SurfaceNets method always creates its data structure at the full resolution of the data grid, and consequently, a large number of triangles can be generated. A few preliminary steps have been taken for the edge detection method to refine meshes locally to conform to the detected surface. However, because of the interaction between mesh extraction and improvement steps, it is possible that a locally refined feature is relocated. As a result, control over refinement and coarsening is required. The focus of the implementations of the various methods was not on speed, and consequently, there is room for speed improvements and reductions of memory usage.

8.2.2 Fast interactive segmentation

Directions for research for the fast interactive segmentation method presented in Chapter 5 are:

- allow other topologies;
- better initialisation;
- relax restrictions on interaction;
- research interaction with semi-automatic methods.

To prevent opening a topological can of worms, we imposed a limit during development to only consider single connected objects, i.e., ones that are topologically equivalent to a sphere (or genus zero). However, other topologies are worth considering, for example, toroids and other objects with holes. In fact, the current method can be extended to handle closed surfaces of any genus. Segmentation using our current software implementation starts with the insertion of a template sphere. New contours are then added and the sphere is deformed to fit a desired object. Unfortunately, certain shapes are very difficult or impossible to create by this mechanism. A better method of initialisation could be to let a user indicate a number of feature points in

the data, e.g., local minima and maxima, and then automatically generate an initial set of contours from these points. Relaxing the condition that contours should be planar would make editing a contour difficult. However, an interesting direction of research is to lift the orthogonality condition. By doing so, more flexible surface fitting can be achieved. Finally, in our research we have only touched the beginning of interacting with semi-automatic segmentation methods. Interacting with, and possibly per-contour or per-contour segment application of different segmentation methods, are of interest.

8.2.3 Statistical shape models

The active shape model of carpal bones in Chapter 7 is only the beginning of a full wrist model. The strength of statistics lies in large quantities. The results we have achieved so far indicate that the quality of the model will improve if more input datasets are added. Unfortunately, the nature of the experiment is such that once we have a model that does not improve by adding new datasets, we know how many datasets we need.

This form of "statistical interpolation" has many applications and further research undoubtly will find many interesting features in the wrist data.

8.3 Concluding remarks

The prevalence of medical imaging modalities requires good quality methods for the analysis of medical data. For example, see-through visualisation of real-time data can help a surgeon during minimally-invasive surgery. Required components to achieve this are surface extraction, non-rigid multi-modality real-time registration and visualisation. In this thesis a part of a part of the required work is performed. Furthermore, surface extraction is not the end of line; after extraction new applications await. For example, creating biomechanical models of tissues, organs, and musculo-skeletal systems. For these applications high-quality volume meshes and quantitative analysis of the data to determine material properties are important.

Validation is the most important part of research and, unfortunately, it is also the most difficult part to perform *in vivo*. Validation, and all the other problems associated with technical medical research, require that clinicians and engineers work closely together; this interaction is crucial to explore new ideas and directions of research.

Bibliography

Nota bene: references are sorted in "Dutch" order, i.e., "de Bruin" is listed under "B" ("von", "van", etc. are handled similarly).

[1] Ainsworth, M. and P. Coggins. The stability of mixed hp-finite element methods for Stokes flow on high aspect ratio elements. *SIAM Journal of Numerical Analysis*, 2000. 38(5):1721–1761. Cited on page 16.

[2] Allamandri, F., P. Cignoni, C. Montani, and R. Scopigno. Adaptively adjusting Marching Cubes output to fit a trilinear reconstruction filter. In: D. Bartz, ed., *EG Workshop on Scientific Visualization '98 Conf. Proc.* Springer Wien, 1998 pp. 25–34. Cited on page 11.

[3] Ambrose, J. Computerized transverse axial scanning (tomography). Part 2: Clinical application. *British Journal of Radiology*, 1973. 46:1023–1047. Cited on page 3.

[4] Ambrose, J. and G. Hounsfield. Computerised tranverse axial tomography. *British Journal of Radiology*, 1973. 46:148–149. Cited on page 3.

[5] Anderson, L. I., ed. *Nikola Tesla: Lecture Before the New York Academy of Sciences April 6, 1897: The Streams of Lenard and Roentgen and Novel Apparatus for Their Production*. Twenty First Century Books, 1994. Cited on page 3.

[6] Babaud, J., A. P. Witkin, M. Baudin, and R. O. Duda. Uniqueness of the Gaussian Kernel for Scale Space Filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986. 8(1):26–33. Cited on pages: 13, 50.

[7] Bajaj, C. L., V. Pascucci, and D. R. Schikore. Fast Isocontouring for Improved Interactivity. In: *Proceedings of the 1996 symposium on Volume visualization*. 1996 pp. 39–. Cited on page 17.

[8] Bank, R. E. and R. K. Smith. Mesh Smoothing Using A Posteriori Error Estimates. *SIAM Journal on Numerical Analysis*, 1997. 34(3):979–997. Cited on pages: 19, 55.

[9] Barequet, G., D. Shapiro, and A. Tal. History consideration in reconstructing polyhedral surfaces from parallel slices. In: *Proceedings of the IEEE Visualization '96 Conference*. IEEE Computer Society Press, 1996 pp. 149–156. Cited on page 14.

[10] Baumgart, B. G. *Winged Edge Polyhedron Representation*. Tech. Rep. STAN-CS-320, Stanford University, 1972. Cited on page 17.

[11] Belyaev, A. and Y. Ohtake. A Comparison of Mesh Smoothing Methods. In: *Accepted for: The 4th Israel-Korea Bi-National Conference on Geometric Modeling and Computer Graphics Tel-Aviv, Israel, February 12-14, 2003*. 2003 . URL `http://mecadserv1.technion.ac.il/public_html/Korea/IK2003.htm` Cited on page 19.

[12] Berzins, M. Mesh Quality: A Function of Geometry, Error Estimates or Both. In: *7th International Meshing Roundtable, Sandia National Lab*. Oct. 1998 pp. 229–238. Cited on pages: 19, 54.

[13] Besl, P. J. and N. D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992. 14(2):239–256. Cited on page 100.

[14] Bhaniramka, P., R. Wenger, and R. Crawfis. Isosurfacing in Higher Dimensions. In: Ertl *et al.* [45], 2000 pp. 267–273. Cited on page 11.

[15] Blacker, T. D. and M. B. Stevenson. Paving: A New Approach To Automated Quadrilateral Mesh Generation. *International Journal for Numerical Methods in Engineering*, 1991. 32(4):811–847. Cited on page 24.

[16] de Blok, S., A. B. Dijkman, and D. J. Hemrika. Transcervical resection of fibroids (TCRM): results related to hysteroscopic classification. *Gynaecol. Endosc.*, 1995. 4:243–246. Cited on page 88.

[17] Bloomenthal, J. *Polygonization of Implicit Surfaces*. Tech. Rep. EDL-88-4, Xerox Corporation - Palo Alto Research Center, Dec. 1988. Cited on page 8.

[18] Boender, E. *Finite Element Mesh Generation from CSG Models*. Ph.D. thesis, Delft University of Technology, 1992. Cited on pages: 16, 30.

[19] Boender, E. Reliable Delaunay-Based Mesh Generation and Mesh Improvement. *Communications in Numerical Methods in Engineering*, 1994. 10:773–783. Cited on page 54.

[20] Boissonnat, J.-D. Shape reconstruction from planar cross sections. *Computer Vision, Graphics, and Image Processing*, 1988. 44(1):1–29. Cited on page 14.

[21] Bolle, R. M. and B. C. Vemuri. Three Dimensional Surface Reconstruction Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991. 13(1):1–14. Cited on page 7.

[22] Brakke, K. The Surface Evolver. *Experimental Mathematics*, 1992. 1(2):141–165. Cited on page 25.

[23] Brett, A. D. and C. J. Taylor. A method of automated landmark generation for automated 3D PDM construction. In: *Proceedings British Machine Vision Conference 1998*. 1998 pp. 914–923. Cited on pages: 99, 100.

[24] Bro-Nielsen, M. Finite Element Modelling in Surgery Simulation. *Proceedings of the IEEE Special Issue on Virtual & Augmented Reality in Medicine*, Mar. 1998. 86(3):490–503. Cited on page 34.

[25] de Bruin, P. W., V. J. Dercksen, F. H. Post, A. M. Vossepoel, G. J. Streekstra, and F. M. Vos. Interactive 3D segmentation using connected orthogonal contours. *Computers in Biology and Medicine*, 2004. Accepted for publication. Cited on pages: 6, 101.

[26] de Bruin, P. W., P. M. van Meeteren, F. M. Vos, F. H. Post, and A. M. Vossepoel. Accurate and High Quality Triangle Models from 3D Grey Scale Images. In: T. Dohi and R. Kikinis, eds., *Medical Image Computing and Computer-Assisted Intervention – MICCAI, Part II, LNCS #2489*. Springer, Sep. 2002 pp. 348–355. Tokyo, Japan. Cited on pages: 6, 75.

[27] de Bruin, P. W., F. M. Vos, F. H. Post, S. F. Frisken-Gibson, and A. M. Vossepoel. Improving Triangle Mesh Quality with SurfaceNets. In: S. L. Delp, A. M. DiGioia, and B. Jaramaz, eds., *Medical Image Computing and Computer-Assisted Intervention – MICCAI*. Oct. 2000 pp. 804–813. Third International Conference, Pittsburgh, PA, USA. Cited on pages: 6, 31, 50.

[28] de Bruin, P. W., F. M. Vos, F. H. Post, A. M. Vossepoel, and S. de Blok. Interactive matching of ultrasound and MR for visualization during myoma resection. In: S. K. Mun, ed., *Proceedings of the SPIE International Symposium on Medical Imaging*. vol. 4681 - Visualization, Image-Guided Procedures and Display, 2002 . Cited on page 6.

[29] Buell, W. R. and B. A. Bush. Mesh Generation – A Survey. *Transactions of ASME, Series B, Journal of Engineering for Industry*, Feb. 1973. 95(1):332–338. Cited on pages: 21, 24.

[30] Campagna, S., L. Kobbelt, and H.-P. Seidel. Directed Edges — A Scalable Representation for Triangle Meshes. *Journal of Graphics Tools: JGT*, 1998. 3(4):1–12. Cited on page 17.

[31] Cannan, S. A. An Approach to Combined Laplacian and Optimisation-Based Smoothing for Triangular, Quadrilateral, and Quad-dominant Meshes. In: *7th International Meshing Roundtable, Sandia National Lab*. Oct. 1998 pp. 479–494. Cited on page 24.

[32] Canny, J. A computational approach to edge-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986. 8(6):679–697. Cited on pages: 13, 51.

[33] Castleman, K. R. *Digital Image Processing*. Prentice Hall Inc., Upper Saddle River, New Jersey 07458, 1996. Cited on page 13.

[34] Catmull, E. and R. Rom. A Class of Local Interpolating Splines. In: R. Barnhill and R. Riesenfeld, eds., *Computer Aided Geometric Design, Salt Lake City*. Academic Press, Mar. 1974 pp. 317–326. Cited on page 69.

[35] Cavendish, J. C. Automatic triangulation of arbitrary planar domains for the finite element method. *International Journal for Numerical Methods in Engineering*, 1974. pp. 679–696. Cited on page 21.

[36] Cignoni, P., P. Marino, C. Montani, E. Puppo, and R. Scopigno. Speeding Up Isosurface Extraction Using Interval Trees. *IEEE Transactions on Visualization and Computer Graphics*, /1997. 3(2):158–170. Cited on page 11.

[37] Coccia, M. E., C. Becatini, G. L. Bracco, G. Bargelli, and G. Scarselli. Intraoperative Ultrasound Guidance for Operative Hysteroscopy. *The Journal of Reproductive Medicine*, May 2000. 5(45):413–418. Cited on page 88.

[38] Cootes, T. F., C. J. Taylor, and D. C. J. Graham. Active Shape Models - their training and application. *Computer Vision Image Understanding*, 1995. 61(1):38–59. Cited on pages: 98, 99.

[39] Cotin, S., H. Delingette, and N. Ayache. Real-Time Elastic Deformations of Soft Tissues for Surgery Simulation. *IEEE Transactions on Visualization and Computer Graphics*, 1998. 5(1):62–73. Cited on pages: 16, 34.

[40] Davies, R. H., C. J. Twining, T. F. Cootes, J. C. Waterton, and C. J. Taylor. A minimum description length approach to statistical shape modelling. *IEEE Transactions on Medical Imaging*, 2002. 21(5):525–537. Cited on page 99.

[41] Davison, N. E., H. Eviatar, and R. L. Somorjai. Snakes simplified. *Pattern Recognition*, 2000. (33):1651–1664. Cited on page 16.

[42] Desbrun, M., M. Meyer, P. Schröder, and A. H. Barr. Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow. In: *Siggraph '99 Conference Proceedings*. Aug. 1999 pp. 317–324. Cited on pages: 25, 27, 28, 110.

[43] Dubuisson, M. P. and A. K. Jain. A modified Hausdorff distance for object matching. In: *Proceedings, 12th IAPR International Conference on Pattern Recognition, Conference A (Jerusalem, Israel, October 9–13, 1994)*. IEEE Computer Society Press, Los Alamitos, CA, 1994, 566-568, Oct. 1994 pp. 566–568. Cited on page 40.

[44] Ekoule, A. B., F. C. Peyrin, and C. L. Odet. A Triangulation Algorithm from Arbitrary Shaped Multiple Planar Contours. *ACM Transactions on Graphics*, 1991. 10(2):182–199. Cited on pages: 14, 15, 66.

[45] Ertl, T., B. Hamann, and A. Varsheny, eds. *Proceedings Visualization 2000*. 2000. Cited on pages: 114, 125.

[46] Farin, G. *Cuves and Surfaces for Computer Aided Design: A Practical Guide*. Academic Press, New York, 1990. Cited on page 15.

[47] Field, D. A. Laplacian smoothing and Delaunay triangulations. *Communications in Applied Numerical Methods*, 1988. 4(6):709–712. Cited on pages: 21, 23, 37, 39.

[48] Fleute, M. and S. Lavalee. Nonrigid 3-D/2-D registration of images using statistical models. In: Taylor and Colchester [123], Sep. 1999 pp. 714–721. Second International Conference, Cambridge, UK. Cited on page 99.

[49] Florack, L. M. J., B. M. T. H. Romeny, J. J. Koenderink, and M. A. Viergever. Scale and the Differential Structure of Images. *Image and Vision Computing*, 1992. 10:376–388. Cited on pages: 13, 50, 52.

[50] Foley, J. D., A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley Publishing Company, second edition, revised fifth printing ed., 1990. Cited on page 69.

[51] Freitag, L. A. On Combining Laplacian and Optimization-Based Mesh Smoothing Techniques. In: *AMD Trends in Unstructured Mesh Generation*. ASME, vol. 220, 1997 pp. 37–43. Cited on pages: 19, 24, 53.

[52] Freitag, L. A. and C. Ollivier-Gooch. A Comparison of Tetrahedral Mesh Improvement Techniques. In: *5th International Meshing Roundtable, Sandia National Laboratories*. Oct. 1996 pp. 87–106. Cited on pages: 4, 18, 24.

[53] Frey, P. J. and P. George. *Mesh Generation, Application to Finite Elements*. Hermes Science Publishing, oxford ed., 2000. Cited on page 16.

[54] Frey, W. H. and D. A. Field. Mesh Relaxation: A New Technique for Improving Triangles. *International Journal for Numerical Methods in Engineering*, 1991. 31:1121–1133. Cited on pages: 19, 30, 54.

[55] Fuchs, H., Z. M. Kedem, and S. P. Uselton. Optimal Surface Reconstruction from Planar Contours. *Communications of the ACM*, 1977. 20(10):693–702. Cited on pages: 14, 66.

[56] Fujiwara, K. Eigenvalues of laplacians on a closed riemannian manifold and its nets. In: *Proceedings of the AMS*. 123, 1995 pp. 2585,2594. Cited on page 24.

[57] Garland, M. Multiresolution Modeling: Survey & Future Opportunities. In: B. Falcidieno and J. Rossignac, eds., *STAR – State of the Art Report, Eurographics 1999*. The Eurographics Associations, 1999 pp. 111–131. Cited on pages: 16, 18.

[58] Garland, M. *Quadric-Based Polygonal Surface Simplification*. Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1999. Software and thesis available at: `http://www.cs.cmu.edu/~garland/quadrics/`. Cited on pages: 17, 35.

[59] Gelder, A. V. and J. Wilhelms. Topological Considerations in Isosurface Generation. *ACM Transactions on Graphics*, 1994. 13(4):337–375. Cited on page 10.

[60] Gibson, S. Constrained Elastic SurfaceNets: generating smooth surfaces from binary sampled data. In: *Proceedings Medical Image Computation and Computer Assisted Interventions, MICCAI '98*. 1998 pp. 888–898. Cited on pages: 10, 31, 35, 39, 50.

[61] Gibson, S. Using Distance Maps for Accurate Surface Representation in Sampled Volumes. In: *Proceedings 1998 IEEE Symposium on Volume Visualization*. Oct. 1998 pp. 23–30. `http://www.merl.com/reports/TR99-25/`. Cited on pages: 12, 35.

[62] Harders, M., S. Wildermuth, and G. Székely. New paradigms for interactive 3D volume segmentation. *The Journal of Visualization and Computer Animation*, 2002. 13:85–95. Cited on page 66.

[63] Hoppe, H. Progressive Meshes. *Computer Graphics*, 1996. 30(Annual Conference Series):99–108. Cited on page 17.

[64] Hoppe, H., T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh Optimization. *Computer Graphics*, 1993. 27(Annual Conference Series, SIGGRAPH '93):19–26. Cited on pages: 17, 18.

[65] Hounsfield, G. Computerized transverse axial scanning (tomography). Part 1: Description of system. *British Journal of Radiology*, 1973. 46:1016–1022. Cited on page 3.

[66] Höwing, F., L. S. Dooley, and D. Wermser. Fuzzy Active Contour Model. *IEE Proceedings Vision, Image and Signal Processing*, 2000. (147):323–330. Cited on page 16.

[67] Hug, J., C. Brechbühler, and G. Székely. Tamed Snake: A Particle System for Robust Semi-Automatic Segmentation. In: Taylor and Colchester [123], Sep. 1999 Second International Conference, Cambridge, UK. Cited on page 16.

[68] John, V. Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 2002. 40(6):775–798. Cited on page 16.

[69] Ju, T., F. Losasso, S. Schaefer, and J. Warren. Dual Contouring of Hermite Data. In: *Siggraph 2002, Computer Graphics Proceedings*. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2002 pp. 339–346. Cited on pages: 46, 47.

[70] Kardestuncer, H., ed. *Finite Element Handbook*. McGraw-Hill, 1987. Cited on page 16.

[71] Kass, M., A. Witkin, and D. Terzopoulos. Active contour models. *International Journal of Computer Vision*, 1987. (1):321–331. Cited on page 16.

[72] Kobbelt, L., S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive Multi-Resolution Modeling on Arbitrary Meshes. *Computer Graphics*, 1998. 32(Annual Conference Series):105–114. Cited on page 12.

[73] Kobbelt, L. P., M. Botsch, U. Schwanecke, and H.-P. Seidel. Feature-Sensitive Surface Extraction From Volume Data. In: E. Fiume, ed., *SIGGRAPH 2001, Computer Graphics Proceedings*. ACM Press / ACM SIGGRAPH, 2001 pp. 57–66. Cited on pages: 11, 46.

[74] Koch, R. M., M. H. Gross, F. R. Carls, D. F. von Büren, and G. Fankhauser. Simulating facial surgery using finite element models. In: *Proc. ACM SIGGRAPH 96*. 1996 pp. 421–428. New Orleans, USA, August 4-9 1996, Computer Graphics Proceedings. Cited on pages: 16, 34.

[75] van Leemput, K., F. Maes, F. Bello, D. Vandermeulen, A. C. F. Colchester, and P. Suetens. Automated Segmentation of MS Lesions from Multi-channel MR Images. In: Taylor and Colchester [123], Sep. 1999 pp. 11–21. Second International Conference, Cambridge, UK. Cited on page 12.

[76] Lindeberg, T. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, Dordrecht, 1994. Cited on page 13.

[77] Lindeboom, G. A. *Andreas Vesalius 1514–1564. Een schets van zijn leven en werken*. Haarlem, de Erven F. Bohn, 1964. Uitgave van 500 genummerde exemplaren. Cited on page 1.

[78] Livnat, Y., H.-W. Shen, and C. R. Johnson. A Near Optimal Isosurface Extraction Algorithm Using the Span Space. *IEEE Transactions on Visualization and Computer Graphics*, 1996. 2(1):73–84. Cited on page 11.

[79] Lo, S. H. A New Mesh Generation Scheme For Arbitrary Planar Domains. *International Journal For Numerical Methods in Engineering*, 1985. 8(21):1403–1426. Cited on page 21.

[80] Lobregt, S. and M. A. Viergever. A discrete dynamic contour model. *IEEE Transactions on Medical Imaging*, Mar. 1995. 14(1):12–24. Cited on pages: 16, 50, 66.

[81] Lorensen, W. and H. Cline. Marching cubes: a high resolution 3D surface construction algorithm. In: *Proc. ACM SIGGRAPH'87*. Jul. 1987 pp. 163–169. Cited on pages: 8, 18, 34, 40, 50.

[82] Lucas, B., G. D. Abram, N. S. Collins, D. A. Epstein, D. L. Gresh, and K. P. McAuliffe. An Architecture for a Scientific Visualization System. In: *Proceedings IEEE Visualization '92*. Oct. 1992 pp. 107–113. Formerly IBM Visualization Data Explorer, now open source.
URL `http://www.opendx.org` Cited on page 92.

[83] Lutz, M. and D. Ascher. *Learning Python*. O'Reilly, 1999. ISBN 1-56592-464-9.
URL `http://www.python.org` Cited on page 73.

[84] Maintz, J. and M. Viergever. A survey of medical image registration. *Medical Image Analysis*, 1997. 2(1):1–36. Cited on page 91.

[85] Mansfield, P., I. L. Pykett, P. G. Morris, and S. P. Newbery. Human whole-body line-scan imaging by NMR. *British Journal of Radiology*, 1978. 51:921–922. Cited on page 3.

[86] McInerney, T. and D. Terzopoulos. Deformable Models in Medical Image Analysis: A Survey. *Medical Image Analysis*, 1996. 1(2):91–108. Cited on pages: 16, 50.

[87] McInerney, T. and D. Terzopoulos. Topology Adaptive Deformable Surfaces for Medical Volume Segmentation. *IEEE Transactions on Medical Imaging*, 1999. 18(10):840–850. Cited on page 16.

[88] McInerney, T. and D. Terzopoulos. T-Snakes: Topology adaptive snakes. *Medical Image Analysis*, 2000. 4:73–91. Cited on page 16.

[89] Meyers, D. and S. Skinner. Surfaces from Contours. *ACM Transaction on Graphics*, 1992. 11(3):228–258. Cited on page 14.

[90] Miller, J. V., D. E. Breen, W. E. Lorensen, R. M. O'Bara, and M. J. Wozny. Geometrically deformed models: a method for extracting closed geometric models form volume data. In: *Proceedings of SIGGRAPH '91*. ACM Press, 1991 pp. 217–226. Cited on page 66.

[91] Montani, C., R. Scateni, and R. Scopigno. Discretized Marching Cubes. In: R. D. Bergeron and A. E. Kaufman, eds., *Visualization '94*. IEEE Computer Society Press, 1994 pp. 281–287. Cited on page 11.

[92] Montani, C., R. Scateni, and R. Scopigno. Decreasing Isosurface Complexity via Discrete Fitting. *Computer-Aided Geometric Design*, Feb. 2000. 17(3):207–232. Cited on pages: 11, 35.

[93] Montignat, J. and H. Delingette. Globally constrained deformable models for 3D object recognition. *Signal Processing*, 1998. (71):173–186. Cited on page 16.

[94] Mukherjee, N. A Hybrid, Variational 3D Smoother For Orphaned Shell Meshes. In: *Proceedings, 11th International Meshing Roundtable, Sandia National Laboratories, September 15–18 2002*. 2002 pp. 379–390. Cited on page 25.

[95] Narkhede, A. and D. Manocha. Fast Polygon Triangulation Based on Seidel's Algorithm. In: A. W. Paeth, ed., *Graphics Gems V*. 1995 pp. 394–397. Cited on page 70.

[96] Nielson, G. M. On Marching Cubes. *IEEE Transactions on Visualization and Computer Graphics*, 2003. 9(3):283–297. Cited on page 11.

[97] Nielson, G. M. and B. Hamann. The asymptotic decider: resolving the ambiguity in marching cubes. In: *Proceedings IEEE Conference on Visualization '91*. Oct. 1991 pp. 88–91. Cited on page 11.

[98] Olabarriaga, S. D. *Human-Computer Interaction for the Segmentation of Medical Images*. Ph.D. thesis, Universiteit van Amsterdam, 1999. Cited on pages: 14, 66.

[99] Oppenheim, A. V., A. S. Willsky, and I. T. Young. *Signals and Systems*. Prentice-Hall, 1983. Cited on page 27.

[100] Osher, S. and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 1988. 79:12–40. Cited on page 46.

[101] Pardo, J. M., D. Cabello, and J. Heras. A snake for model-based segmentation of biomedical images. *Pattern Recognition Letters*, 1997. 14(18):1529–1538. Cited on pages: 16, 50.

[102] Pébay, P. P. and T. J. Baker. A Comparison Of Triangle Quality Measures. In: *10th International Meshing Roundtable, Sandia National Lab*. Oct. 2001 pp. 327–340. Cited on page 19.

[103] Pentecôte, N. and A. Vigliotti. Crashworthiness of helicopters on water: Test and simulation of a full-scale WG30 impacting on water. *International Journal of Crashworthiness*, 2003. 8(6). Cited on page 16.

[104] Perona, P. and J. Malik. Scale space and edge detection using anisotropic diffusion. In: *Proceedings of the IEEE Computer Society Workshop on Computer Vision, Miami FL.* 1987 pp. 16–27. Cited on page 110.

[105] Piegl, L. A. and W. Tiller. Cross-Sectional Design with Boundary Constraints. *Engineering With Computers*, 1999. 15(2):171–180. Cited on page 66.

[106] Rhodes, P. *An Outline History of Medicine*. Butterworths, 1985. Cited on page 2.

[107] Rueckert, D., A. F. Frangi, and J. A. Schnabel. Automatic construction of 3-D statistical deformation models of the brain using nonrigid registration. *IEEE Transactions on Medical Imaging*, 2003. 22(8):1014–1024. Cited on page 99.

[108] Ruestow, E. G. *The Microscope in the Dutch Republic: The Shaping of Discovery*. Cambridge University Press, 1996. Cited on page 3.

[109] Rusinkiewicz, S. and M. Levoy. QSplat: A Multiresolution Point Rendering System for Large Meshes. In: K. Akeley, ed., *Siggraph 2000, Computer Graphics Proceedings*. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000 pp. 343–352. Cited on page 17.

[110] Schroeder, W., K. Martin, and B. Lorensen. *The Visualization Toolkit*. Prentice Hall PTR, 2nd ed., 1999. ISBN 0-13-954694-4.
URL `http://www.vtk.org` Cited on page 73.

[111] Schroeder, W. J., J. A. Zarge, and W. E. Lorensen. Decimation of Triangle Meshes. *Computer Graphics*, 1992. 26(2):65–70. ACM SIGGRAPH '92. Cited on pages: 17, 18.

[112] Schuurman, A. H., M. Maas, P. F. Dijkstra, and J. M. G. Kauer. Ulnar variance and the shape of the lunate bone, a radiological investigation. *Acta Orthopaedica Belgica*, 2001. 67(5):464–467. Cited on pages: 98, 105.

[113] Shekhar, R., E. Fayyad, R. Yagel, and J. F. Cornhill. Octree-based decimation of marching cubes surfaces. In: *Proceedings of the conference on Visualization '96*. IEEE Computer Society Press. ISBN 0-89791-864-9, 1996 pp. 335–ff. Cited on page 11.

[114] Shen, H.-W., C. D. Hansen, Y. Livnat, and C. R. Johnson. Isosurfacing in Span Space with Utmost Efficiency (ISSUE). In: R. Yagel and G. M. Nielson, eds., *IEEE Visualization '96*. 1996 pp. 287–294. Cited on page 11.

[115] Snel, J. G. *Wrists in Space, Deformable models for segmentation and matching techniques for registration of 3D MR and CT images of the wrist*. Ph.D. thesis, Universiteit van Amsterdam, 2000. Cited on pages: 13, 16, 31, 50.

[116] Sutton, P., C. Hansen, H. Shen, and D. Schikore. A Case Study of Isosurface Extraction Algorithm Performance. In: W. de Leeuw and R. van Liere, eds., *Data Visualization 2000*. Springer, 2000 pp. 259–268. Cited on page 11.

[117] Swade, D. *The difference engine: Charles Babbage and the quest to build the first computer*. Viking Penguin, 2001. ISBN 0-670-91020-1. Cited on page v.

[118] Szymczak, A. and J. Vanderheyde. Extraction of Topologically Simple Isosurfaces from Volume Datasets. In: *IEEE Visualisation 2003*. 2003 pp. 67–74. Cited on page 63.

[119] Taleisnik, J. *The wrist*. Churchill Livingstone, 1985. Cited on pages: 98, 105.

[120] Taubin, G. A Signal Processing Approach to Fair Surface Design. *Computer Graphics*, 1995. 29(Annual Conference Series):351–358. Cited on pages: 24, 25, 27, 28.

[121] Taubin, G. Geometric Signal Processing on Polygonal Meshes. In: *STAR – State of the Art Report, Eurographics 2000*. 2000 . Cited on pages: 25, 27, 28.

[122] Taubin, G., T. Zhang, and G. H. Golub. Optimal Surface Smoothing as Filter Design. In: B. F. Buxton and R. Cipolla, eds., *Computer Vision - ECCV'96, 4th European Conference on Computer Vision, Cambridge, UK, April 15-18, 1996, Proceedings, Volume I*. Springer, vol. 1064 of *Lecture Notes in Computer Science*, 1996 pp. 283–292. Cited on page 28.

[123] Taylor, C. and A. Colchester, eds. *Medical Image Computing and Computer-Assisted Intervention – MICCAI*. Springer, Sep. 1999. Second International Conference, Cambridge, UK. Cited on pages: 117, 119, 120.

[124] Terzopoulos, D. and H. Qin. Dynamic NURBS with Geometric Constraints for Interactive Sculpting. *ACM Transactions on Graphics*, 1994. 13(2):103–136. Cited on page 66.

[125] Thompson, J. F., B. K. Soni, and N. P. Wheatherill. *Handbook of Grid Generation*. CRC Press LLC, 1998. Cited on pages: 16, 21.

[126] Turk, G. Re-tiling polygonal surfaces. *Computer Graphics*, 1992. 26(2):55–64. Cited on page 17.

[127] Várady, T., R. R. Martin, and J. Cox. Reverse engineering of geometric models–an introduction. *Computer-Aided Design*, 1997. 29(4):255–268. Cited on page 7.

[128] Verbeek, P. W. and L. J. van Vliet. On the location error of curved edges in low-pass filtered 2-D and 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1994. 16(7):726–733. Cited on page 52.

[129] Verbeek, P. W. and L. J. van Vliet. *Bandwidth of Gaussian scale parameter* σ. Tech. Rep. 199, Delft University of Technology, Pattern Recognition Group, unknown. Cited on page 13.

[130] Veron, P. and J. C. Leon. Shape Preserving Polyhedral Simplification With Bounded Error. *Computers & Graphics*, 1998. 22(5):565–585. Cited on page 40.

[131] Vos, F., P. de Bruin, J. Aubel, G. Streekstra, M. Maas, L. van Vliet, and A. Vossepoel. A Statistical Shape Model Without Using Landmarks. In: J. Kittler, M. Petrou, and M. Nixon, eds., *ICPR17, 17th International Conference on Pattern Recognition, Cambridge, UK, Aug. 23-26*. IEEE Computer Society Press, Los Alamitos, vol. 3, 2004 pp. 714–717. Cited on page 6.

[132] Webb, S., ed. *The Physics of Medical Imaging*. IOP Publishing Ltd, 1988. Reprinted with corrections 1992. Cited on page 3.

[133] Weigle, C. and D. C. Banks. Complex-valued contour meshing. In: *Proceedings of the conference on Visualization '96*. IEEE Computer Society Press. ISBN 0-89791-864-9, 1996 pp. 173–ff. Cited on page 11.

[134] Whitaker, R. T. Reducing Aliasing Artifacts in Iso-Surfaces of Binary Volumes. In: Ertl *et al.* [45], 2000 pp. 23–32. Cited on page 46.

[135] Whitworth, H. A., R. Bendidi, D. Marzougui, and R. Reiss. Finite element modeling of the crash performance of roadside barriers. *International Journal of Crashworthiness*, 2004. 9(1). Cited on page 16.

[136] Wilhelms, J. and A. Van Gelder. Octrees for faster isosurface generation. *ACM Transactions on Graphics*, 1992. 11(3):201–227. ISSN 0730-0301. Cited on page 11.

[137] Winslow, A. M. *Equipotential Zoning of Two-Dimensional Meshes*. Tech. Rep. UCRL-7312, July 1964, Lawrence Radiation Laboratory, University of California, Livermore, California, Jun. 1964. Available from NTIS, (National Technical Information Service) Springfield, Virginia, 22151. Cited on page 21.

[138] Winslow, A. M. *An Irregular Triangle Mesh Generator*. Tech. Rep. UCRL-7880, Aug. 1964, Lawrence Radiation Laboratory, University of California, Livermore, California, Aug. 1964. Available from NTIS, (National Technical Information Service) Springfield, Virginia, 22151. Cited on page 21.

[139] Witkin, A. P. Scale Space Filtering. In: *Proceedings, International Joint Conference on Artificial Intelligence*. 1983 pp. 1019–1023. Cited on page 52.

[140] Witkin, A. P. and P. S. Heckbert. Using Particles to Sample and Control Implicit Surfaces. In: *SIGGRAPH '94*. 1994 pp. 269–277. Cited on page 8.

[141] Yezzi, A., S. Kichenassamy, P. Olver, and A. Tannenbaum. A Geometric Snake Model for Segmentation of Medical Imagery. *IEEE Transactions on Medical Imaging*, 1997. 16(2):199–209. Cited on page 16.

[142] Zhou, T. and K. Shimada. An Angle-Based Approach to Two-Dimensional Mesh Smoothing. In: *Proceedings, 9th International Meshing Roundtable*. Sandia National Laboratories, Oct. 2000 pp. 373–384. Cited on pages: 24, 25, 60.

[143] Zhu, J. Z., O. C. Zienkiewicz, E. Hinton, and J. Wu. A New Approach to the Development of Automatic Quadrilateral Mesh Generation. *International Journal for Numerical Methods in Engineering*, 1991. 32(4):849–866. Cited on page 24.

[144] Ziou, D. and S. Tabbone. Edge Detection Techniques - An Overview. *International Journal of Pattern Recognition and Image Analysis*, 1998. 8:537–559. Cited on page 13.

Summary

This thesis investigates methods for medical visualisation. A common approach to visualisation of 3D data volumes from CT or MRI systems is to extract surfaces that describe the shape of anatomical objects. Such surfaces can help a clinician with diagnosis, planning, navigation, manipulation, visualisation and surgical simulation. For these purposes two requirements are important: accuracy and quality. Accuracy describes the conformance of a mesh to an object surface, whereas quality relates to the compactness and well-formedness of the mesh, e.g., well-shaped triangles with a high area to perimeter ratio are important for subsequent processing steps, for example finite element analysis.

Two methods were developed that meet both requirements: the first is an iso-surface based SurfaceNets method, the second is a scale-space edge-detection method. The basis of both methods is formed by the application of alternating mesh improvement and extraction techniques. Mesh improvement is performed by variants of Laplacian smoothing and edge swapping. The meshes created by the SurfaceNets method are equal in accuracy and superior in triangle quality compared to the Marching Cubes iso-surface extraction algorithm. The scale-space edge-detection method provides robust edge-detection and localisation in combination with a set of mesh improvement tools that improve the quality and prevent geometric and topological errors from occurring (e.g., holes and inverted mesh elements). For each surface extraction technique the robustness and effectiveness of the alternating approach is shown.

Another method that is presented in this thesis is a new interactive segmentation method. Conventionally, manual segmentation is performed by outlining a desired shape in a stack of 2D images. Our new method circumvents the problems related to contour connecting by operating directly in 3D. Starting with a template object that consists of three orthogonal, planar, and connected contours, a desired shape can be created by moving points of each contour. The user can add new contours in three orthogonal directions; the new contours are automatically linked at the intersections of the current model with a slicing plane. The linked contours reduce the amount of tedious manual interaction, and this is reduced even further by the addition of an automatic edge-detection method. It is shown that the fast interactive segmentation method provides effective feedback and control for semi-automatic segmentation methods.

Finally, a statistical shape analysis of two carpal bones is performed. Active shape

models are initialised by a large number of example shapes. From this set modi of variation are calculated that describe shared characteristics of the input shapes. Effectively, shapes are parametrised by a linear combination of each modus of variation. Conventionally, an explicit landmark correspondence is required between the shapes. In our approach this correspondence is determined automatically. By doing so, the initial shapes are allowed to be defined in a free-form manner. The fast interactive segmentation method proves to be a valuable tool for creating these initial shapes.

In conclusion, the set of described methods together provide the means to create 3D surface representations from 3D medical image data in a short amount of time, even if the data is noisy. The final result is an accurate *and* high-quality mesh that is well suited for further modelling and post-processing steps, such as visualisation, mechanical analysis, and simulation.

Samenvatting

Dit proefschrift onderzoekt methoden voor medische visualisatie. Een gebruikelijke aanpak om 3D medische beelddata, zoals CT of MRI, te visualiseren is de extractie van oppervlakken die de vorm van anatomische structuren en objecten beschrijven. Dergelijke oppervlakken kunnen een arts helpen bij diagnose, planning, navigatie, manipulatie, visualisatie en chirurgische simulatie. Voor deze toepassingen zijn twee eisen belangrijk: nauwkeurigheid en kwaliteit. Nauwkeurigheid geeft aan hoe goed een oppervlak een object beschrijft en kwaliteit geeft de compactheid en welgevormdheid van het oppervlak aan. Bijvoorbeeld, welgevormde driehoeken met een hoge oppervlak–omtrek-ratio zijn belangrijk voor verdere verwerking zoals eindige-elementensimulatie.

Twee methoden worden ontwikkeld die voldoen aan beide voorwaarden: de eerste is een op iso-oppervlakken gebaseerde SurfaceNets-methode, de tweede is een scale-space edge-detectie methode. De basis van beide methoden wordt gevormd door het alternerend toepassen van oppervlak verbeter- en extractietechnieken. Oppervlakverbetering wordt gedaan door varianten van Laplacian smoothing en edge-swapping. De oppervlakken die worden gemaakt door de SurfaceNets-methode zijn van dezelfde nauwkeurigheid en superieur in driehoekskwaliteit vergeleken met het MarchingC̃ubes iso-oppervlak extractie-algoritme. De scale-space edge-detectie methode levert robuuste edge-detectie en lokalisatie in combinatie met een set van gereedschap voor oppervlakverbetering waarmee de kwaliteit wordt verbeterd en geometrische fouten worden voorkomen (bijvoorbeeld gaten en omgekeerde elementen in het oppervlak). Voor beide oppervlakextractiemethoden wordt de robuustheid en effectiviteit van de alternerende aanpak aangetoond.

Een andere methode in dit proefschift is een nieuwe interactieve segmentatiemethode. Normaal gesproken wordt handmatige segmentatie gedaan door een gewenste vorm aan te geven in een stapel van 2D beelden. Onze nieuwe methode ondervangt de problemen die voorkomen bij het verbinden van de contouren door direct in 3D te werken. Door te starten met een sjabloonobject dat bestaat uit drie onderling verbonden, orthogonale en planaire contouren, kan een gewenste vorm worden gemaakt door de punten van elke contour te verplaatsen. De gebruiker kan contouren maken in drie orthogonale richtingen door automatisch de snijpunten te laten verbinden van de huidige vorm met een vlak. De verbonden contouren verminderen de hoeveelheid saaie handmatige interactie en dit wordt nog verder gereduceerd door de toevoeging van een edge-detectie methode. Aangetoond wordt dat de snelle handmatige interac-

tieve segmentatiemethode effectieve terugkoppeling en controle geeft voor toepassing van semi-automatische segmentatiemethoden.

Als laatste wordt een vormanalyse van twee carpalen gedaan. Active shape modellen worden geïnitialiseerd door een grote hoeveelheid voorbeeld datasets. Aan de hand van deze data worden modi van variatie berekend die overeenkomende karakteristieken beschrijven van de voorbeeldvormen. Effectief worden vormen geparametriseerd met een lineaire combinatie van elke modus van variatie. Gewoonlijk is een expliciete landmark-correspondentie vereist tussen de vormen. In onze aanpak wordt deze correspondentie automatisch bepaald. Op deze manier kunnen de voorbeeld datasets op een willekeurige manier worden beschreven. De snelle interactieve segmentatiemethode heeft bewezen een waardevolle applicatie te zijn bij het maken van de voorbeeld datasets.

Concluderend geeft de verzameling van beschreven methoden de mogelijkheid om een 3D oppervlak beschrijving te maken van 3D medische beelddata in een korte tijd, zelfs als de data ruizig is. Het eindresultaat is een nauwkeurig oppervlak van hoge kwaliteit, dat ook geschikt is voor verdere verwerking, bijvoorbeeld visualisatie, mechanische analyse en simulatie.

Curriculum Vitæ

Paul Wessel de Bruin was born on May 18, 1971 (a Saturday) in Gouda, The Netherlands. In 1990 he completed his secondary school at the Coornhert Gymnasium in Gouda. In the same year he started studying Mechanical Engineering at Delft University of Technology. He concluded his studies at the Laboratory for Measurement and Control in 1997 with a M.Sc. thesis on controlling a model of a musculosketal system using neural networks. During a short period he worked for the Rorema Group in Leidschendam, The Netherlands. In 1998 he started as a Ph.D. student at Delft University of Technology in the Visualisation group under supervision of Frits Post. During 1999/2000 he worked for four months at MERL in Cambridge, MA on SurfaceNets, under the guidance of Sarah Frisken.

From April 1, 2004, he is a researcher in the Department of Orthopædic Surgery of the Leiden University Medical Center in Leiden, the Netherlands.

Paul Wessel de Bruin is op 18 mei 1971 (een zaterdag) geboren te Gouda. In 1990 voltooide hij het eindexamen gymnasium β aan het Coornhert Gymnasium te Gouda. In dat jaar begon hij aan de studie Werktuigbouwkunde aan de Technische Universiteit Delft. Na zijn draai gevonden te hebben in de Vakgroep Meet- & Regeltechniek in de sectie van Henk Stassen studeerde hij af in 1997 op een onderzoek naar het aansturen en regelen van spier–skelet systemen met behulp van neurale netwerken. Gedurende een korte tijd werkte hij voor de Rorema Groep in Leidschendam. In 1998 trad hij in dienst als promovendus bij de Technische Universiteit Delft in de visualisatiegroep onder leiding van Frits Post. In 1999/2000 werkte hij gedurende vier maanden bij MERL in Cambridge MA aan SurfaceNets, onder de leiding van Sarah Frisken.

Sinds 1 april 2004 is hij werkzaam als onderzoeker bij de Afdeling Orthopædie van het Leids Universitair Medisch Centrum.