Matthias Harders

Surgical Scene Generation for Virtual Reality-Based Training in Medicine

Surgical Scene Generation for Virtual Reality-Based Training in Medicine

Matthias Harders

Surgical Scene Generation for Virtual Reality-Based Training in Medicine

Matthias Harders
ETH Zurich, Switzerland

To my family

Preface

Recent changes in patient awareness and acceptance of adverse effects in medicine and an increasing focus on patient safety have put traditional educational paradigms in the medical area to the test. Related to this, the great potential of training approaches using virtual reality (VR) techniques in surgical education has been recognized for a long time.

A key element of effective VR-based training is the ability to generate variable scenarios. Due to this, the adaptation of a trainee to a specific scene can be avoided and natural variation, which is encountered in most real life situations, can be included. Model generation in VR-based applications is in general a difficult task. The increase of computational power has enabled the display of larger and more detailed virtual environments, thus reinforcing the need for improved methods for model acquisition, enhancement, optimization, and adaptation.

In the context of surgical scene generation for training in medicine, this book examines the main components needed to define such a scenario. Three steps in the process are analyzed – the generation of the scene geometry, the modelling of organ appearance, and the definition of biomechanical parameters. This book provides an extensive overview of related work in these three different directions and introduces specific solutions in detail. Examples show the outcome and performance of the presented methods. The work represents an excellent resource for anybody involved in generating training scenarios in medical education, as well as in VR-based training in general.

This book is based on my *Habilitationsschrift* at ETH Zurich, Switzerland in 2007, and covers activities which have been carried out in the Virtual Reality in Medicine group during the development of various surgical simulator systems over a period of about eight years.

Particular gratitude is due to my colleagues who significantly contributed to the different chapters – Raimundo Sierra for the scene geometries, Rupert Paget, Volker Meier, and Johannes Hug for the object texturing, and Gerald Bianchi and Bryn Lloyd for the biomechanical parameter setting. I also would like to thank Stefan Tuchschmid and Daniel Bachofen, who worked hard to

make our envisioned surgical simulator system for hysteroscopy a reality, as well as Michael Bajka for his continuous support over the years, his medical expertise, and his trust in our developments. I am also thankful to the numerous other developers involved over the years in this area.

Sincere thanks go to Gabor Szekely, who has been my mentor and adviser during the long period at ETH. His continuing support, trust, and advice, which have been a cornerstone in my scientific development, are very much appreciated. I also would like to acknowledge Sebastien Ourselin, who generously hosted me during a research visit to his lab at CSIRO. Gratitude is also due to David Hellier and Dan Popescu, who provided a pleasant working atmosphere as well as enjoyable tennis matches. In addition, thanks go to Herve Delingette and Peter Bösiger for their support of this work.

Needless to say, I am also grateful to all the BiWi members for the pleasant and motivating atmosphere at our institute – especially to Philippe Cattin for being an excellent office mate. I also would like to express my gratitude to the editorial staff of Springer Verlag – especially Helen Desmond – for their support of this endeavor.

This list is certainly incomplete, and I apologize to anyone omitted.

Finally, heartfelt thanks go to all my friends and especially to my family for their support and encouragement over the years.

Zurich   *Matthias Harders*
                                                                    September 2007

Contents

List of Figures

List of Tables

List of Acronyms

Symbol	Meaning
1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
ABF	Angle-Based Flattening
AIDS	Acquired Immune Deficiency Syndrome
BEM	Boundary Element Method
CSG	Constructive Solid Geometry
CT	Computed Tomography
CAD	Computer-Aided Design
FDA	Food and Drug Administration
FEM	Finite Element Method
FFT	Fast Fourier Transform
GA	Genetic Algorithm
GE	General Electric
HSV	Hue, Saturation, Value
ICM	Iterative Conditional Modes
IOM	Institute of Medicine
JND	Just Noticeable Differences
LCPDF	Local Conditional Probability Density Function
LEM	Long Element Method
LSCM	Least Squares Conformal Maps
MIPS	Most Isometric Parameterizations
MIS	Minimally Invasive Surgery
MRI	Magnetic Resonance Imaging
MRF	Markov Random Field
MSM	Mass-Spring Model
NMR	Nuclear Magnetic Resonance
OB/GYN	Obstetrics/Gynecology
OR	Operating Room

PAFF	Point-Associated Finite Field
PCA	Principal Component Analysis
PDM	Point Distribution Model
RGB	Red, Green, Blue
RF	Radio Frequency
SPH	Smoothed Particle Hydrodynamics
TMM	Tensor Mass Model
TSVQ	Tree-Structured Vector Quantization
US	Ultrasound
VR	Virtual Reality

1

Introduction

1.1 Motivation

Recent years have brought about a drastic change in patient awareness and acceptance of adverse effects in medical care. The combination of this process with an increasing focus on patient safety has put traditional educational paradigms in the medical area to the test. Especially in the surgical domain, the time-honored concept of theoretical education followed by supervised clinical practice – often referred to as "see one, do one, teach one" – is becoming less and less acceptable [283], wherefore innovative and complimentary methods of teaching medical knowledge have to be sought. Further concerns are rooted in the high cost of teaching in a clinical environment [61].

Increased awareness of patient safety has for instance been stimulated by recent studies on surgical or medical errors, and the associated health care costs. According to a study carried out by the Institute of Medicine (IOM) of the United States National Academies in 1999 [145], medical errors cause the deaths of an estimated 44,000 patients in United States hospitals per year. The report states that mortality due to medical mistakes is higher than that due to highway accidents, AIDS, or breast cancer in the United States. Moreover, an additional \$8.8 billion of direct health care costs is expended, not taking into account secondary effects. While the report has been criticized for poor study methodology and vague differentiation between preventable and non-preventable adverse effects, it still reflects a need for reducing complications in the medical area. Similar findings have been reached in a study conducted in Canada in 2000 [13]. An estimated 70,000 preventable adverse events are reported, which occurred in Canadian hospitals and resulted in 1.1 million additional hospital days. Furthermore, it is estimated that about 9,000 to 24,000 of these events resulted in death. In [294] it is indicated that an additional 3.3 million days of hospitalization became necessary due to medical error. Similar trends have also been stated for a number of European countries [280, 197, 177].

Figure 1.1. Laparoscopic intervention in gynecology (courtesy of Clinic of Gyn., University Hospital Zurich, Prof. D. Fink)

According to [153], about a half of medical errors occur during surgical interventions, of which 75% are preventable. Unfortunately, exact numbers of surgical error are not easy to obtain, which makes the recording of these events an important initial step. Open, anonymous discussion of errors is a valuable means for quality assurance, which is for instance practiced in the pilot community. As an example, to address these shortcomings, a Critical Incident Reporting System was established in Switzerland in 1998, providing an online tool for recording adverse events [55].

One key factor, which has been identified as a cause of considerable numbers of surgical errors, is the rapid growth of medical knowledge and the very fast turnover of new technologies, which makes it hard for practitioners to keep up. Possibly the most significant change in recent times has been the advent of minimally invasive surgery (MIS) in the late 1980s. This technique minimizes the damage to healthy tissue during interventions on internal organs. An example of this procedure for laparoscopic interventions is depicted in Figure 1.1. The relatively large cuts in open surgery are replaced by small incisions, through which optical and surgical instruments are inserted. Reduced tissue injury and careful selection of the entry points results in a major gain in patient recovery after the operation as well as reduced scarring. The price for these advantages is paid by the surgeon, who loses direct contact with the operation site. Visual information is acquired via endoscopic cameras and displayed on a monitor, thus impairing normal hand-eye coordination. In addition, much of the manipulative freedom usually available in open surgery is lost. Therefore, performing operations under these conditions demands very specific capabilities of the surgeon, which can only be gained with extensive training. Nevertheless, MIS procedures have been superseding traditional approaches in several areas. For instance, laparoscopic gallbladder surgery has largely replaced conventional interventions. More than 90% of all cholecystectomies in the United States are done using this MIS technique. Taking all

these points into consideration, the need for improved medical training and continuing education becomes apparent.

The remainder of this chapter serves two purposes. Firstly, a broad overview of the area of surgical simulation is provided to lay the foundations for the following, more detailed discussion of training scenario definition. Secondly, the complete process of scene generation is introduced and the stage is set for the subsequent in-depth scrutiny of the individual process elements.

1.2 Surgical Education

1.2.1 Overview

A number of educational options currently exists. Next to the usually practiced "learning by doing," which involves supervised interventions on real patients, training can also be performed using cadavers, animals, lifelike mockups, actors, multimedia tools, and simulations. All of these have positive and negative aspects, which are summarized in Table 1.1.

These different options should usually be combined in a complementary fashion leveraging the individual advantages. The selection of the means of training needs to be governed by the specific skills that are to be trained. In this sense, one has to discriminate between basic manipulative vs. procedural skills. The former include, for instance, hand-eye coordination needed for handling of endoscopic cameras; while the latter relate to more cognitive processes, such as problem identification during intervention and selection of the appropriate reaction. As an example, rehearsal of knot tying does probably not require a highly sophisticated simulation environment. Instead, a simple box setup might be sufficient.

The need to provide alternatives to traditional education in health care is reflected in the recent trend towards establishment of simulation centers at hospitals. These activities have initially been set up by anesthesiology teams and focus on training with highly realistic patient mannequins (see, e.g., [90, 54]). However, the adaptation of medical curricula to include alternative means of training is still at a very early stage. This is especially true for Virtual Reality (VR)-based training. Nevertheless, it should be noted that in 2004 an unprecedented step has been taken by the United States Food and Drug Administration (FDA), which decided to approve simulation-based training for stenting procedures in the carotid artery using a specific product. In principle, physicians are now only allowed to perform this intervention after having trained with an approved simulator system [93]. The effect of this decision on the future of the field of VR-based surgical simulation remains to be seen.

Table 1.1. Advantages and disadvantages of non-VR training options

Paradigm	Pros	Cons
Training in OR	Real behavior of patient, actual environment, and tools	Risk to patient, subjective skill evaluation, additional OR time/cost, limited exposure to rare cases
Cadavers	Reduced time constraints, exact anatomy	Availability of cadavers, changed tissue behavior, no physiology, one-time use, special preparation required, inappropriate for MIS training
Animals	In vivo physiology and anatomy	Prohibited in some countries, limited availability of animals, ethical concerns, limited similarity to human anatomy, specialized facilities required, high cost, one-time use
Mock-ups	Relatively inexpensive, usually allows repeated training, effective rehearsal of basic manipulative skills	Needs replacement of components after training, tactile and biomechanical properties of tissue incorrect, limited physiology, low immersion
Non-interactive, PC-based tools	Detailed 3D visualizations, intraoperative movies, self-paced learning, low hardware requirements, limited cost	No tactile information, very limited interactivity, no evaluation, no immersion

1.2.2 VR-Based Training

Immersive training environments using VR technology have been a topic of research in diverse application areas, including fire fighting, vehicle driving, space missions, triage, emergency handling, maintenance, hazardous material handling, or military activities. Overviews of such systems can for instance be found in [43] or [279]. The major advantage of VR-based training is the interactive and engaging setting, enabling learning through personal experience. Generally, tasks are represented which would be too dangerous, too expensive, or too infeasible to undertake in reality. A very successful example, which in fact existed long before the term VR was initially coined, is flight simulation for pilot education. Since this technology has been fully accepted and integrated into training in aviation, it will be examined in more detail.

Initial attempts at using simulated environments for pilot education were undertaken a century ago at the inception of manned flight (Figure 1.2). From the early mechanical trainers, flight simulators have grown into highly com-
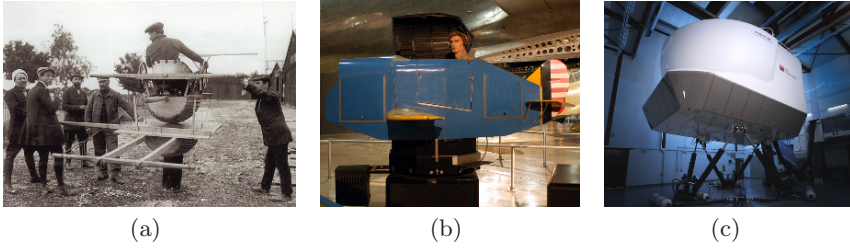
Figure 1.2. History of flight simulation devices. (a) Early synthetic flight training device from 1910, (b) Link Trainer from the 1930s (courtsey of National Museum of the United States Air Force), (c) Contemporary high fidelity flight simulator (courtesy of Swiss Aviation Training Ltd.)

plex, fully immersive computer-based tools for teaching in aviation. Nowadays simulator training is an indispensable element embedded into pilot education. Aviation authorities have issued regulations how simulator training hours can be logged. Furthermore, official certification procedures for new simulators have been established. In this context it is interesting to note that commercial pilots may be certified for new types of aircraft based on simulator experience alone.

Similar to other VR-based training environments, flight simulators allow practicing complex or emergency situations. The reduction of airplane accidents over the last decades is partly attributed to simulator training; the system developed by Edward Link in the 1930s has resulted in a significant reduction of accidents in instrument-only meteorological conditions.

Nevertheless, as with surgical simulation, in this field of research the necessary degree of realism for achieving specific training effects is not completely understood (see, e.g., [41]). Real-time simulation of aeroelasticity, i.e., modeling of high-velocity, turbulent, compressible flow over aeroelastic airfoils, requires significant computational resources. Also, the realistic modeling of weather phenomena affecting flight, e.g., local wind shear during a final approach to landing, is extremely complex. Therefore, physics in flight simulators is often only approximative.

1.2.3 Surgical Simulation

The great potential of training approaches using VR techniques in surgical education was recognized at the beginning of the 1990s, where initial attempts were reported, e.g., in [105] and [228]. A wide range of systems has been proposed and implemented in recent years. Several academic projects have been carried out, related, for instance, to diagnostic endoscopic investigations [281, 36, 15], laparoscopic surgical procedures [148, 19, 180, 263], arthroscopic interventions [310], eye surgery [75, 231], and radiological procedures [286, 110]. Apart from this nonexhaustive list, a growing number of commercial
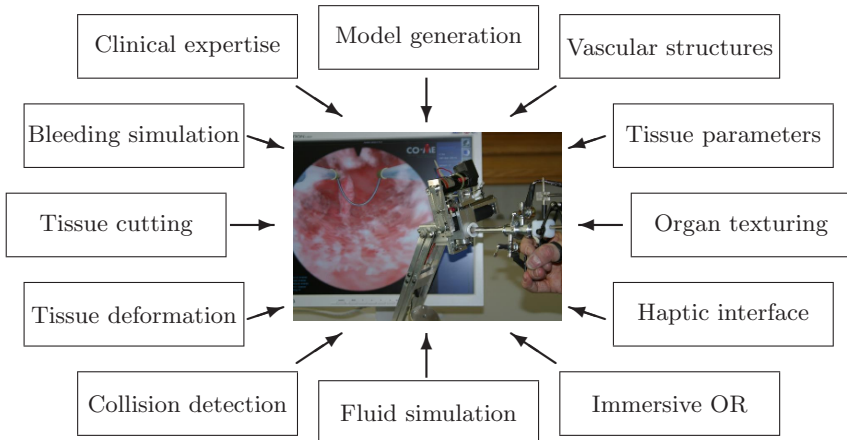
Figure 1.3. Components of a surgical simulator

products is available on the market. Further details can be found in recent overview papers [165, 17] or in the online surgical simulator repository [157] compiled by our lab.[1] Despite these developments, surgical simulation is still far from being included in the medical curriculum as a standard tool.

It should be noted that a number of further application areas of surgical simulation techniques have been envisioned in the past, including patient-specific rehearsal, testing of new interventions, patient education, practitioner certification, student admission, and planning. However, the technology is still too immature to provide any satisfying solution.

In order to build a highly realistic surgical simulator, several components have to be integrated. Figure 1.3 depicts all the elements which are typically required. The components belong to four major categories. First and foremost is the clinical background, which defines the problem area, identifies the training needs, and influences the system development at all stages. The next category focuses on the generation of surgical scenes. This includes models of healthy and pathological anatomy, organ textures, vessel structures, as well as tissue deformation parameters. These scenarios are used in the interactive simulation, which is composed of elements of the last two categories. The former denotes software modules, which are combined into the virtual representation of the surgery, while the latter comprises the interface elements controlling the simulation, for instance, the haptic device or the immersive training environment.

An in-depth discussion of all these elements is beyond the scope of this work; however, to illustrate the significance of the individual components – especially with respect to training scene generation – the elements of the last two categories will be shortly reviewed.

---

[1] `http://www.virtualsurgery.vision.ee.ethz.ch`

Haptic interface. The haptic device is the human-machine interface between the trainee and the simulator. It renders haptic feedback to the user based on the interaction with the virtual world. Moreover, it provides a natural means for controlling the simulation. In the optimal case, real surgical tools are modified to serve as input devices, avoiding interaction via mouse or keyboard.

Immersive environment. The sense of presence plays an important role in achieving a training effect. To enable user immersion into the environment, the surrounding and interaction metaphors should be the same as during a real intervention. Optimally, a complete OR is replicated for the simulation, also comprising auditory feedback. Moreover, by including an anesthesia simulator, a complete team training setting could be generated.

Fluid simulation. Several surgical procedures require the modeling of fluids. For instance, in hysteroscopy the uterus is distended with liquid media to access and visualize the uterine cavity. Therefore, the real-time modeling of fluid motion is required. The changing boundary conditions for the fluid solver are given by the diverse dynamic objects in the surgical scene.

Collision detection. In order to process the interaction between soft and rigid objects in a simulator, efficient collision detection algorithms are required. In addition, an appropriate response to the collisions needs to be determined. A geometrical representation of all scene objects has to be generated to this end.

Tissue deformation. Deformable modeling is a key component in any surgical simulator. Efficient and robust methods are required to deliver dynamic behavior of soft tissue at interactive speed. This usually necessitates the generation of meshes for all deformable objects.

Tissue cutting. Interactive simulation of soft tissue cutting is an integral part of surgery simulation. Smooth cuts into the organ models have to be created, while preserving the numerical stability of the underlying deformation computation. The process usually requires a remeshing of the organ models, as well as an update of scene textures.

Bleeding simulation. During cutting or strong collisions with organs, vessels can be damaged, which causes bleeding depending on the size of the vessel. Thus, a bleeding model is required to simulate different effects, e.g. oozing or fast spurts. The type of bleeding depends on the information coming from the vascular structure of the scene.

This list is certainly not exhaustive. Other possible elements exist, e.g., patient physiology, biochemical processes, models of neurology, and genetics. However, due to the already high complexity of the previously discussed elements, other simulation components currently receive little attention. The focus of this book is on the individual steps of the training scene generation, which will be addressed in more detail below.

Although the concepts presented in the book are applicable in general, the following discussions will mainly be carried out within the scope of two
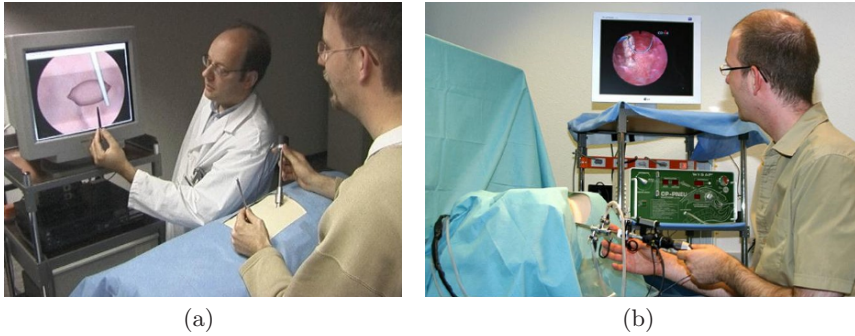
Figure 1.4. Overview of surgical simulator systems. (a) Laparoscopic simulator (LaSSo), (b) Hysteroscopic simulator (HystSim)

surgical simulator systems, which have been developed at the Computer Vision Lab of ETH Zurich – a simulator for diagnostic laparoscopy in gynecology[2] (LaSSo) [259], and a simulator for therapeutic hysteroscopy[3] (HystSim) [114]. Figure 1.4 provides an overview of the two systems.

In the LaSSo project (1996–2000), the limits of realism in surgical simulation were explored and extended, focusing on important components of VR-based endoscopic simulators. This included the construction of detailed geometric models of the anatomical site, the development of texture and blood vessel generation approaches, the study of different visualization and illumination methods, the development of a framework for physically-based simulation of elastomechanical tissue deformation, the design of a specialized parallel supercomputer to speed up the expensive calculations, and the integration of force-feedback devices.

The mantra of high realism has also been a key focus in the HystSim project (2001–2009). The driving application of this work is the development of a generic surgical training simulator for hysteroscopy. A key target is to go beyond the rehearsal of basic manipulative skills, and enable the training of procedural skills like decision making and problem solving. A prototype system has been created based on several modules, providing simulation of soft tissue deformation, collision detection and response, tissue cutting, distension fluid simulation, a hysteroscopy tool as input device to the simulator, and a replicated OR environment.

1.3 Training Scene Generation

A key element of effective VR-based training is the ability to generate variable scenarios. In this way, adaption of a trainee to a specific scene can be avoided

---

[2] More details on `http://www.vision.ee.ethz.ch/projects/Lasso/start.html`
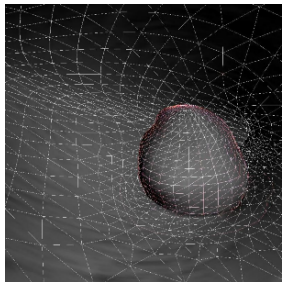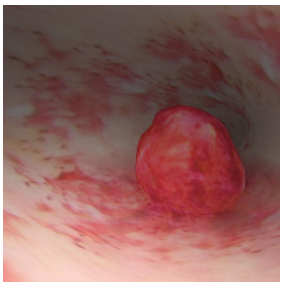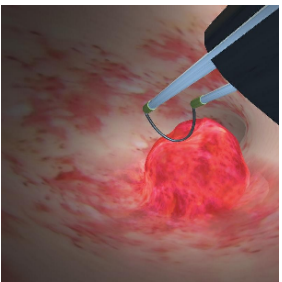
[3] More details on `http://www.hystsim.ethz.ch`

and natural variation, which is encountered in most real-life situations, can be included.

Model and content generation for VR-based applications is in general a difficult task. The increase of computational power has enabled the display of larger virtual environments, thus reinforcing the need for improved methods for model acquisition, enhancement, optimization, and adaptation. The objects in a VR scene can usually be categorized into either man-made or natural entities. Two general strategies are followed to create these objects – artificial generation or real-world-based acquisition. The former technique focuses on manual or semiautomatic design of virtual environments using computer-based modeling tools. In addition, some simple, exactly defined structures can be generated completely automatically (see, e.g., [189]). Until about a decade ago, synthetic model generation was the preferred approach; however, recent years have witnessed a paradigm shift towards real-world-based acquisition. This is mainly due to the slow, tedious, labor intensive, and costly processing of scenes. As an example, in [37] it is reported that building a model of a kitchen at quarter-inch accuracy required three man-years of work. Similar times are also reported in the movie industry, where virtual scenes are generated for special effects, e.g., the city models in the movie *Superman Returns* took 15 man-years to complete. Therefore, real-world-based model generation has received increasing attention recently. A typical technique in this class is the acquisition of real objects via sensors and their subsequent reconstruction (e.g., [276, 135]). Also, obtaining volumetric representations of organs using medical imaging technology and the subsequent extraction of their surfaces, based for instance on the *Marching Cubes* algorithm [170], falls into this category. Another possibility is the direct usage of CAD models of specific real-world counterparts. Since an increasing number of man-made entities undergo a computer-based design process, more and more highly detailed virtual representations are becoming available. Finally, formal analysis of real shapes or processes, trying to generate a compact mathematical description for model derivation, also belongs in the second category.

The resulting models often need further processing. For instance, noise introduced during the acquisition process has to be removed, while object mesh properties are maintained, thus requiring enhanced smoothing techniques (e.g., [262]). Moreover, mesh size is often quite large, making a polygon reduction step necessary. Again, during surface simplification, characteristic shape features have to be maintained (see, e.g., [95]).

The ability to present variable scenarios is also a crucial component in surgical training systems. In currently existing simulators this point is usually neglected. In order to replicate the day-to-day workload of a surgeon, different patients with varying manifestations of pathologies have to be integrated. Moreover, it should be possible to provide unusual cases. This is analogous to the scenario generation in flight simulation, where varying weather conditions, traffic situations, airports, airplane types, and emergencies can be selected. Optimally, a scene would be automatically generated according to

Training scene generation		
Geometry	**Appearance**	**Biomechanics**
• Variable models • Healthy anatomy • Tumor integration • Triangular surfaces • Tetrahedrization	• Image enhancement • Procedural textures • Texture synthesis • Texture mapping	• In vivo acquisition • Genetic optimization • Analytical derivation
Chapter 2	*Chapter 3*	*Chapter 4*

Figure 1.5. Overview of training scene generation steps covered in this study

the specifications of the medical expert overseeing the training session. This could for instance include key parameters of a patient, such as height, weight, or age. In the context of surgical scene generation for VR-based training in medicine, this book examines the main components needed to define a scenario. Three steps in the process are analyzed in detail – the generation of the scene geometry, the modeling of organ appearance, and the definition of biomechanical parameters. Figure 1.5 visualizes these different components.

1.4 Outline

In the following chapters the three main components of the scene generation framework are introduced in detail. First, Chapter 2 provides insight into the generation of the scene geometry, including the anatomy of the healthy uterus as well as typical uterine tumors. Thereafter, Chapter 3 sheds light on various approaches to synthesizing appropriate textures for organ surfaces, as well as the mapping of these to the previously created geometries. After this, Chapter 4 details techniques for setting the biomechanical parameters describing the deformation behavior of the soft tissue objects. An optimization-based

technique as well as analytical solutions are presented. Finally, Chapter 5 concludes the book with a brief overview of the application of the scene generation framework within the context of the hysteroscopy simulator, as well as an outline of future steps in this research area.

2

Geometry

2.1 Introduction

The first step in the surgical scenario definition process is the generation of the geometrical representation of the scene objects. The latter mainly consist of the human organs and possibly their pathological changes. A key component within this step is the ability to provide different scenes for every training session, reflecting the natural variability in cases and patients.

2.1.1 Definitions

Before the modeling process can be discussed, the appropriate paradigm for the geometric description of the objects has to be selected. Depending on the simulation module, different requirements have to be fulfilled. The most important of these shall be briefly discussed.

Visualization models. For the visual display of objects in a surgical scene the definition of their surfaces usually suffices. Currently available rendering hardware allows the usage of highly detailed models, while still fulfilling necessary update rates of about 30Hz to avoid flicker. It should be noted that advanced rendering techniques using surface texturing can be applied to further enhance the visual realism of an object. This will be discussed in more detail in the next chapter. Often the surface of an associated deformation model is used for the visual display; however, as discussed below, these usually have a much lower resolution, thus compromising visual fidelity. Instead of defining an object by its hollow surface shell, it is also possible to use volumetric representations, for instance derived directly from datasets acquired with medical imaging technology.

Deformation models. Different requirements have to be met for the models employed for the computation of soft tissue mechanics. Haptic interfaces are often integrated in medical simulators, which require a force update
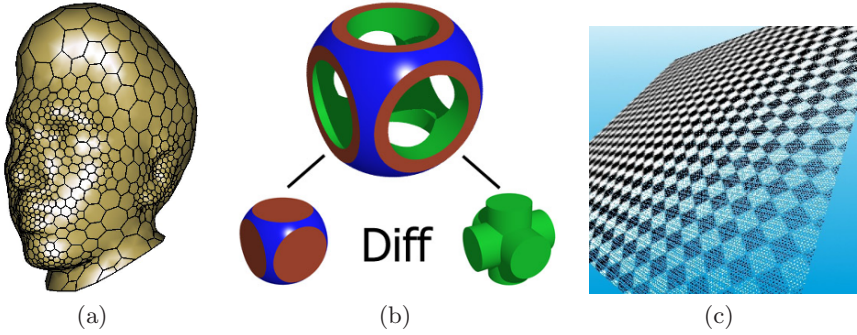
Figure 2.1. Object representation paradigms. (a) Simplex mesh of human face, (b) CSG tree example, (c) Plane sampled with surfels (from [210])

of up to 1 kHz. Since the rendered feedback is directly or indirectly coupled to the mechanical model, high update rates for the latter have to be satisfied, which limits the resolution of the mechanical models. Moreover, topological changes have to be supported by the model, which occur during interventional procedures such as tissue excisions. Finally, depending on the underlying model, further boundary conditions might have to be met, such as mesh quality, validity, or adaptability. The most common representation of the deformation models are tetrahedral meshes.

Collision models. One of the major bottlenecks of soft tissue simulation is the fast and accurate detection of object collisions. In the majority of cases this process can be based on the same models as those used for deformable objects. However, for the sake of real-time performance, using an amended representation might be advisable. For instance, rigid objects in a surgical scene could be approximated by much simpler geometric primitives. That is, the shaft of an endoscope can be represented in the collision module simply by a collection of lines. Also, simpler deformable models can be approximated by a coarse sampling of the object surface with a number of collision points.

It should be noted that the discussed models do not necessarily have to be distinct. While the usage of different object representations in the simulator modules has several advantages, it also increases the complexity of the system. For instance, cutting an object requires the coherent update of several models, which, in addition, has to be accomplished in real time. Nevertheless, the major advantage of such an approach is the independence of the individual resolutions, which generally outweighs the drawbacks. Regarding the model generation process, it is usually sufficient to obtain one of the representations, from which the others can be derived.

In the presented scenario definition scheme, the starting point is the surface representation of the scene objects for the visualization module. A number of

possible options for geometric surface description exist, which are outlined in the following.

Triangle meshes. With regard to visualization, the most popular approach to describing geometric entities are polygonal models. Objects are represented by piecewise linear approximations of their surfaces. The latter are sampled at specific locations, which represent the polygon vertices. The most common primitives are flat triangles. A mesh is defined by geometric and topological information $\mathcal{M} = (V, I)$, where its vertices are given by $V = \{\mathbf{v}_i \in I\!R^3 \,|\, i \in \{0, \ldots, n-1\}\}$, and I contains the topology of the points defining the mesh. In the simplest case only the indices of triangle vertices are specified, while more complex representations also contain further information, such as adjacency relationships or polygon edges. Piecewise linear triangular meshes exhibit several advantages, such as fast intersection tests or straightforward mesh refinement and simplification. Nevertheless, drawbacks also exist, such as the lack of smoothness, rendering artifacts, and difficulties in defining differential quantities. Also, the manual modification of the mesh at varying scales is problematic.

Simplex meshes. An alternative to represent three-dimensional surfaces is 2-simplex meshes, which are the topological dual of triangle meshes [72]. In such a mesh $\tilde{\mathcal{M}}$, a vertex $\tilde{\mathbf{v}}_i \in I\!R^3$ is connected to exactly three neighbors $\{\tilde{\mathbf{v}}_{N_1(i)}, \tilde{\mathbf{v}}_{N_2(i)}, \tilde{\mathbf{v}}_{N_3(i)}\}$. The osculating sphere given by these four points then defines the normal $\tilde{\mathbf{n}}_i$ and the mean curvature $\tilde{K}_i$ at vertex $\tilde{\mathbf{v}}_i$. Simplex meshes are slightly easier to control than triangular meshes, due to the constant vertex valence and the ability to consider the border as an empty face. The evaluation of finite difference operators is more robust as compared to triangle meshes. Furthermore, higher order continuity (i.e., C^2 continuity) can be enforced in a simple and efficient manner. A drawback of the representation is that the implicitly defined faces are usually not flat. Moreover, for rendering of the mesh on current graphics hardware, it is necessary to triangulate the simplex mesh. Direct usage of the dual triangle representation is inappropriate since vertex curvature would be strongly underestimated. Therefore, additional vertices need to be included for this step. An example of this representation is depicted in Figure 2.1(a).

Constructive solid geometry. The underlying idea of this object representation technique is the construction of new and more complex geometries by a combination of primitive shapes using Boolean algebra. The basic objects typically have a compact mathematical description, e.g., cuboids, cylinders, or spheres. Operators applied on the primitives are union, intersection, or difference. By combining multiple levels of CSG operations, relatively complex objects can be generated. However, this technique is only viable for man-made structures. Natural objects can not easily be approximated. In addition, as with most of the other approaches, a triangulation

of the objects for rendering is necessary [293]. A typical CSG tree is visualized in Figure 2.1(b).

Parametric polynomial surfaces. Instead of using piecewise linear patches, a parametric surface modeling approach allows the description of objects by higher order basis functions. These techniques are also called spline representations [70], the most common being the B-spline model. In the latter approach, a surface is defined by evaluating the tensor-product of univariate basis functions $S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_i^{<k>}(u) B_j^{<l>}(v) \mathbf{q}_{i,j}$ controlled by parameters u, v. Here, $\mathbf{q}_{i,j}$ are the $n \times m$ control points, and $B_i^{<k>}(u)$, $B_j^{<l>}(v)$ the polynomial B-spline basis functions of order $k-1$ and $l-1$ in u and v parameter direction, respectively. The B-spline basis function of order n can be obtained by convolving the basis function of order $n-1$ with the box function of order 0. A B-spline of order n has a continuity of C^{n-1}, i.e., a cubic spline is curvature-continuous. It should be noted, that the basis functions have local support, therefore control point modifications affect only a small portion of the spline. Typically, control points are only approximated; however, an interpolation of a given set of vertices can be determined by solving a linear equation system for the required control points. Nevertheless, control of fine detail or representation of high frequency structures with the B-spline model can be complicated.

Implicit descriptions. A compact definition of closed continuous shapes can be obtained with implicit functions. These are given by the set of points which satisfy an implicit equation $f(\mathbf{x}) = 0, \mathbf{x} \in I\!\!R^3$. Such a representation is resolution independent and has the advantage of simple inside/outside tests. Moreover, several of these functions can be combined to define more complex shapes. A popular subset of these representations are quadrics, which define a surface by polynomials up to 2nd degree. Given by the equation $\mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c = 0$, ten independent components define the 3×3 matrix $\mathbf{A}$, vector $\mathbf{b}$, and constant c, thus determining a quadric. This yields typical shapes, e.g., spheres, ellipsoids, paraboloids, or hyperboloids. An advantage of the functional definition is the ability to explicitly calculate exact surface properties such as curvature, tangents, or normals. However, a difficulty is the visualization of the surfaces, since no explicit surface elements are given. This can be tackled by uniformly distributing particles on the surfaces, which are rendered as disc-shaped primitives [296]. Nevertheless, obtaining a specific complex shape with fine surface detail is complicated. Further problems exist regarding intuitive editing of objects as well as interactive topology modifications.

Meshless representations. Similar to the application of particles in the previous approach, general surface elements can be used as an alternative, mesh-free object representation [210]. These techniques are based on surfels – point primitives without explicit connectivity. To this end the surfaces and textures of complex objects are sampled along several views. The approach is not optimal for man-made structures with flat surfaces
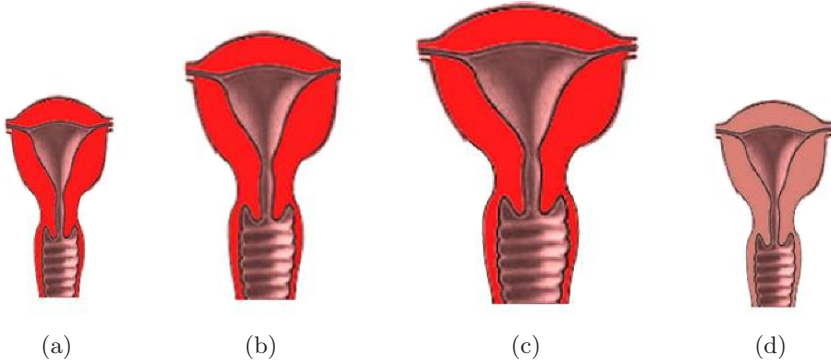
Figure 2.2. Variability of the uterine corpus size and shape at varying ages (adapted from EGONE e-learning internet portal, Zurich, ©2001-2007). (a) Adolescent female, (b) Nullipara, (c) Multipara, (d) Post-menopausal

and sharp corners; however, for natural shapes with high surface detail it exhibits some advantages. Multiresolution can be achieved by taking samples at several resolutions, thus obtaining a surfel mipmap. Also, interactive modifications of the representation are simplified, since no mesh has to be updated. An example visualization using surfels is shown in Figure 2.1(c).

The discussed object representation approaches exhibit inherent advantages and disadvantages. Nevertheless, all have been applied in the past to model organs. Simplex meshes have mainly been designed to support object reconstruction during segmentation. The technique has for instance been utilized to obtain models of the brain [14, 267], the liver [180], and complete thoracic and abdominal anatomy [46]. However, often only the initial steps of the segmentation rely on this representation, while for later phases, such as the actual simulation, different model paradigms are applied. Constructive solid geometry (CSG) is usually not used separately to model organ geometry; however, it has been applied in conjunction with other representations, e.g., [155] describes the combination with implicit surface modeling. Other previous approaches using implicit representations of organs are based on obtaining iso-surfaces from object skeletons [268, 88, 6]. Unfortunately, the resulting model geometries are too smooth for realistic simulations. Spline surfaces have been used to model organ geometries for surgical simulators, for instance in [261]. Other parametric descriptions of surfaces have been applied in object reconstruction tasks. These are often representations of closed surfaces for organs with spherical topology. In this context, surface expansion into a series of spherical harmonic functions [34] has for instance been applied to segmentation of MRI data [221]. Unfortunately, these methods are not appropriate for interactive applications. Promising alternatives for object
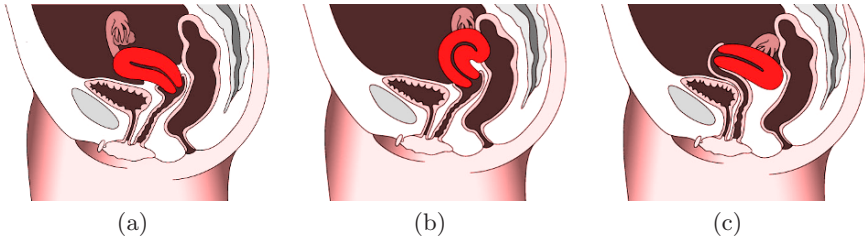
Figure 2.3. Variability of uterus positions (adapted from EGONE e-learning internet portal, Zurich, ©2001-2007). (a) Normal position, (b) Retroflexion, (c) Retroversion

modeling are meshless representations. These techniques are relatively new and have only recently received increased attention. Nevertheless, the majority of approaches focus on the representation of the physical model, while the object surface is still modeled by a triangle mesh [68].

While simplex meshes or meshless representations would have been reasonable options, the geometry representation in our scene generation process is mainly based on triangular surface models. In the following discussion the focus will be mainly on the geometry generation for hysteroscopy simulation. Most of the presented algorithms have been developed for this specific application. Nevertheless, the proposed schemes should also be applicable in other areas.

2.1.2 Relevance to Surgical Education

In order to achieve the fidelity needed for effective surgical training, the organ geometries of the training scene are a key element. Since the anatomy of any two patients will never be alike, the variability of the models is of major importance. Healthy volunteers exhibit considerable variability in their organ geometries – Figure 2.2 depicts typical uterine shapes at varying ages, while Figure 2.3 visualizes different positions of the uterus in the abdomen. The size of the uterus varies largely during pregnancy and will not return to its previous dimensions after childbirth. In addition, shape and size also change with the menstrual cycle.

Even larger variations can be found in pathological alterations. These can be malignant neoplasms, such as cervical or uterine carcinoma; adenomyosis, which is the ectopic growth of endometrial tissue within the myometrium; or benign neoplasms, such as myomas or polyps. The former are the most common benign tumors in female patients over 35 years of age. They are composed of smooth muscle and a variable amount of fibrous tissue, and are classified into several categories depending on their position. Figure 2.4 exemplifies the different manifestations of this kind of pathological alteration. Depending on the type of neoplasm encountered during inspection, varying

<p align="center">(a)   (b)   (c)</p>
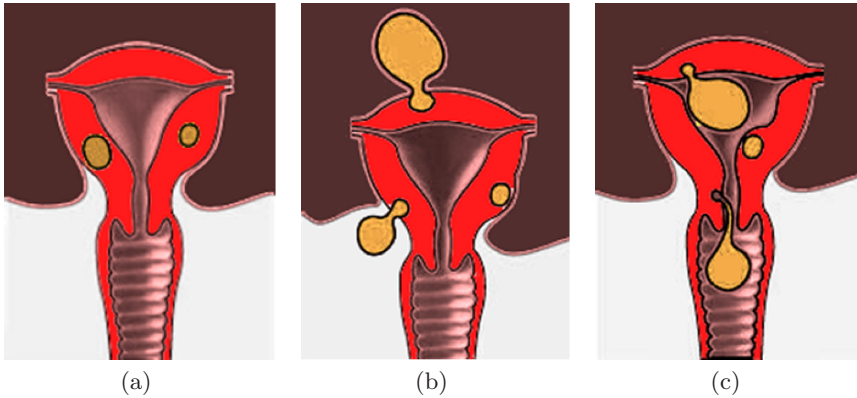
Figure 2.4. Myoma classification according to position relative to uterine wall (adapted from EGONE e-learning internet portal, Zurich, ©2001-2007). (a) Intramural myomas, (b) Subserous myomas, (c) Submucous myomas

surgical approaches have to be followed. Therefore, the ability to add different pathologies to a healthy anatomical geometry is a major component of surgical scene generation. In a sense, the pathologies can be considered as the training problems posed to the trainee. Analogous to flight simulation, where different weather conditions, airports, system malfunctions, etc., can be defined, surgical simulation also needs the same breadth of configurable training conditions. Further examples of pathology geometries from real interventions are depicted in Figure 2.5.

In conclusion, a simulator system has to be able to generate anatomical models incorporating the natural variability of the healthy anatomy, and to seamlessly integrate a wide spectrum of different pathologies according to the specifications from physicians. This is especially necessary for training of higher level skills, since repeated training with the same single organ model would obscure training effects due to user adaptation. Just as in real practice, the training scene has to be different from session to session.

2.1.3 Process Elements

In the next sections, all steps required for the generation of the training scene geometries will be discussed in detail. A twofold strategy is followed for this. The first part is the derivation of a statistical model of healthy anatomy, targeting the formal description of the macroscopic organ variability in individual patients. The second part is the artificial genesis of a wide spectrum of different pathologies and their seamless integration into the healthy organ according to the specifications from gynecologists.

The former process consists of four steps. The first element is the radiological data acquisition of different uteri to form a database for the statistical

(a)   (b)   (c)
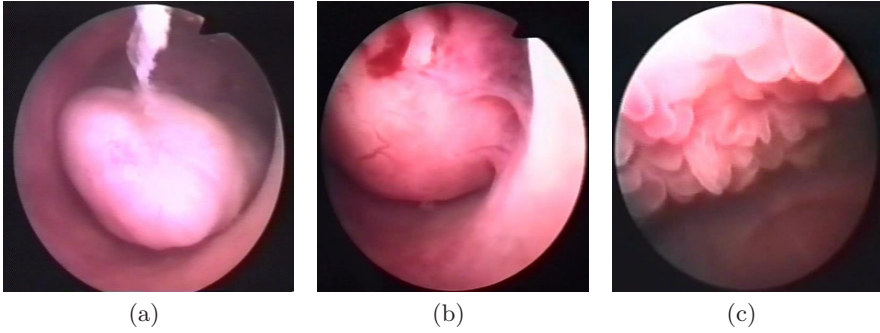
Figure 2.5. Geometries of different pathological alterations. (a) Polyp of uterine corpus, (b) Myoma located at fundus, (c) Polypoid endometrial hyperplasia

model. Next, these data are segmented, while correspondences between the segmentation results are maintained. Subsequently, statistical methods developed for the analysis of shapes are applied to encode the variability of healthy organs. Finally, new organ instances are derived based on standard clinical parameters by applying a constrained minimization.

The latter part aims at the addition of a wide variety of pathologies – the main focus being the modeling of benign neoplasms. Three different models for the generation of these tumors are scrutinized. First, a purely geometrical CAD-based algorithm, which artificially creates myomas or polyps by defining objects via skeletons and contour curves. Next, the growth of uterine leiomyomas is simulated with cellular automata. Finally, additional physiological information is integrated into the tumor growth process using particle-based models.

Subsequently, depending on the applied generation process, an additional step has to be carried out to merge the obtained geometries. Moreover, based on surface meshes, tetrahedral models for the deformation simulation also need to be determined. In addition, further mesh smoothing or reduction processes can also be applied.

In order to provide an overview of the main steps, the scene generation approach is visualized in Figure 2.6. More details of the different derivation schemes for appropriate geometries of the scene objects are given after a review of related work.

2.2 Previous Work Related to Surgical Simulation

Although the generation of variable training scenes is an indispensable part of surgical simulators, this topic has surprisingly received only little attention in the past. In the majority of projects only one static organ model is used. The geometry is usually derived from an exemplary dataset, which has been

Generation of healthy anatomy
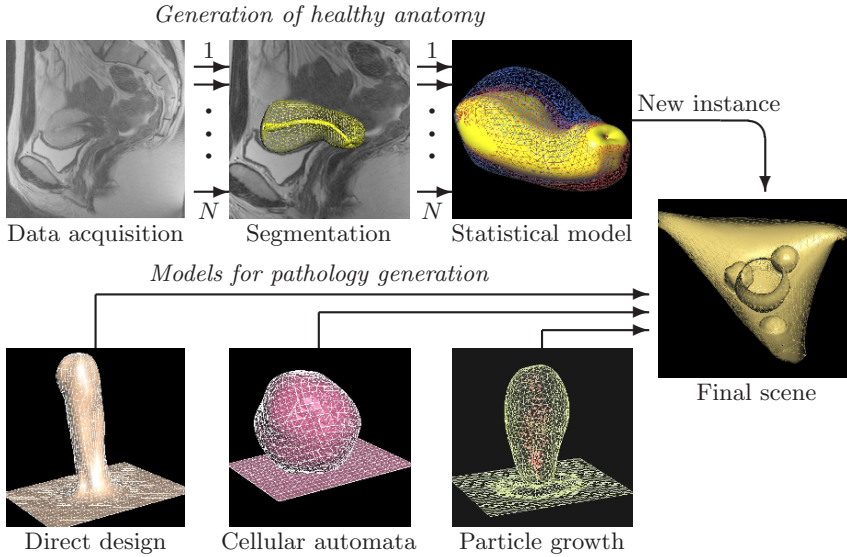
Figure 2.6. Overview of scene geometry generation process for hysteroscopy simulation

acquired with medical imaging technology from a patient or volunteer. For instance, several projects focusing on temporal bone surgery [40, 211, 3] derived a 3D volumetric model from CT data. Also, for arthroscopy simulators, the geometry of structures in the human knee has been obtained from MRI images [310, 102, 119]. Similar approaches have been followed in other surgical application domains, often involving labor and time intensive manual segmentation. Nevertheless, due to the single scene geometry, natural variations of human anatomy are not included. Moreover, it has to be noted that data are often acquired from healthy subjects, wherefore pathologies are not present in the models and thus have to be manually added. There are, however, also research endeavors which focus on obtaining larger organ databases. In a study described in [249], 30 models of the human liver were obtained by automatic segmentation of CT scans, including healthy volunteers, patients with lesions such as cysts or tumors, and patients after segmentectomy.

Several projects justify the limitation to a single organ model by the notion of patient specific simulation [47, 9]. While this might be appropriate for static offline surgical planning scenarios, it is questionable whether patient specific interactive surgery rehearsal is feasible with the currently available technologies. A major obstacle is the unavailability of acquisition methods for the biomechanical organ tissue parameters of a specific patient. In addition, the visual appearance of organs can usually not be obtained. Standard medical imaging does not provide the necessary information, since image generation is based on different physical properties such as tissue density. Moreover,
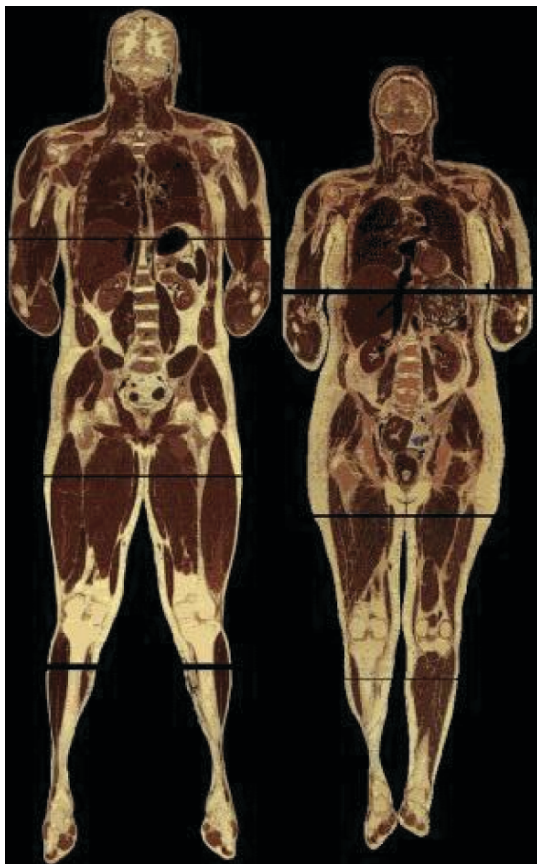
Figure 2.7. Coronal slices through Visible Human male and female cryosectional dataset (courtesy of Visible Human Project, US National Library of Medicine)

resolution, partial volume effects, noise, and other artifacts limit the accuracy of the model geometries. Therefore, a patient specific scene is at best only a rough approximation of the real situation, which will be detrimental to the surgery preparation, in the worst case even leading to negative training effects.

High resolution datasets obtained from cadavers are a special case. The first initiative in this direction is the Visible Human Project of the National Library of Medicine [195]. It focuses on the creation of complete, anatomically detailed, three-dimensional representations of human bodies. Medical data were acquired based on CT, MRI, and cryosectional slices from a male and a female volunteer (Figure 2.7). In related work, further datasets were obtained for Asian patient anatomies in the Chinese and Korean Visible Human projects [307, 143]. Due to the high resolution and good image quality of the Visible Human datasets, they were used by several projects as a source for anatomical models of high accuracy, e.g., [224, 180, 259, 263]. Similar to

(a)   (b)   (c)   (d)
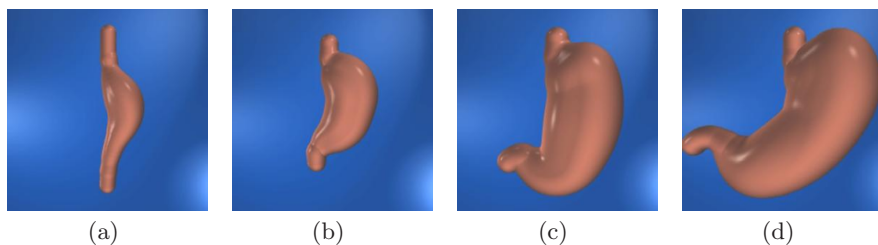
Figure 2.8. Animation steps of growth of embryonic gaster (from [77])

this, in [187] a cryosectional dataset of the cadaveric female pelvis has been utilized to obtain a detailed model for surgical simulation. Furthermore, the VOXEL-MAN software should be mentioned [230], which provides a detailed and fully annotated segmentation of the male data. In general, the Visible Human projects provide high resolution anatomy and to some extent also color information. However, the extensive efforts required for cadaver preparation and data segmentation render this approach impracticable for the generation of large numbers of training models.

In the context of patient-specific data acquisition, optical scanning and 3D reconstruction from photographs should also be mentioned. Vision-based extraction as well as 3D laser scanning has been used to acquire dental anatomy based on hydrocolloid imprints for orthodontic treatment planning in [152, 149, 4]. An approach focusing on surface scanning of soft tissue specimens is described in [10]. Ex vivo 3D scanning and reconstruction techniques are examined with respect to their performance and suitability to scan specimens such as a plastinated human heart or porcine organs. Unfortunately, all scanning systems experienced problems due to the highly reflective organ surfaces.

Instead of deriving model geometries from real anatomical examples, some attempts have been made to manually design organ shapes according to anatomical atlases. In [126] organ models have been created by hand, based on anatomical books and intraoperative videos using a commercial modeling tool. In [88] model geometry has been defined by implicit surfaces. A scene of abdominal organs is generated by manual definition of point or line primitives in space controlling the overall shape. Unfortunately, the resulting organs are very smooth and their visual appearance rather simplistic. A spline-based modeling system for creation of scenes containing deformable anatomical objects has been discussed in [147]. Freeform surfaces are used as basic primitives for geometry definition. An interactive scene editor allows the manual design of visualization and deformation meshes. Unfortunately, due to the required manual editing, these approaches are also not appropriate.

An interesting alternative is to model the actual growth process of human organs to obtain different anatomical shapes. First steps in this direction have been described in [77]. This work is the first attempt to geometrically model

(a)   (b)   (c)   (d)   (e)
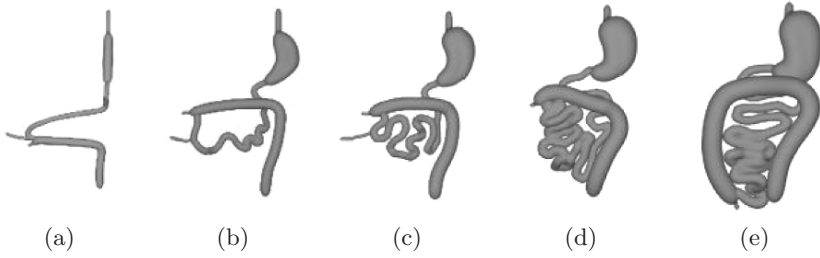
Figure 2.9. Animation steps of growth of embryonic intestine (from [76])

the development of healthy human anatomy. However, the main target is not the model generation, but the creation of rule-based animation sequences for the visualization of organ development. Nevertheless, the created models could also be used for simulation purposes. This approach models shapes by a functional representation (F-Rep), which is a generalization of traditional implicit surfaces combined with constructive solid geometry. The topology of an organ is represented by a skeleton in the form of a tree or cycled graph. The organ shape is approximated by ellipsoidal clusters located at selected points on the skeleton. Organ growth is encoded by changes of the skeleton segments according to continuous algebraic L-systems. The time-dependent growth model also takes surrounding organs into account. Some randomness is introduced by setting probabilities of the different L-system production rules. Anatomical knowledge is included in the model via statistical measurements obtained from anatomical atlases and cross-sectional images. These data include for instance growth speeds or diameters of inscribed ellipsoidal clusters. It is reported that manual parameter tuning is necessary to obtain appropriate results. Figure 2.8 shows steps in the resulting animation for the growth of the human embryonic gaster. An extension of the approach has been presented in [76]. The L-system-based growth is enhanced with dynamic boundary conditions due to gravity and colliding organs in the abdomen. The proposed model has for instance been applied to the growth animation of the digestive system of human embryos, as depicted in Figure 2.9. Related work is also discussed in [63], where a model for the growth of a human embryo brain is provided. Based on drawings found in anatomical atlases, skeletons are specified in 2D defining separate animation keyframes.

The work discussed so far mainly focuses on the generation of healthy anatomical models. Much less attention has been paid to providing the actual problem cases, i.e., the pathological alterations of the models.

One related area of research is the simulation and modeling of trauma, which mainly examines firearm injuries. In [81] a simulation of tissue damage due to small arm projectiles and fragments is presented. A three-dimensional nonlinear spring lattice is used to compute the soft tissue response to the ballistic penetration. Measurements on cadavers are carried out to obtain

stress-strain curves. The tissue damage of a low-speed bullet caused to a myocardium material slab has been simulated in [125]. Transversely isotropic hyperelastic tissue material as well as a strain-based failure model have been applied to represent the tissue. The Material Point Method was used to discretize the equations of motion. Another related example is the physics-based simulation of firearms injury to the human cranium [188]. The collision of a firearm projectile with and penetration into the human parietal bone is modeled based on Finite Elements. While the target of these projects is the analysis of the traumatic process, the discussed methods could also be used to generate models for specific simulators, for instance training systems for wound debridement.

Within the context of pathology generation, the highly active field of mathematical modeling of cancerous growth also has to be examined. A large body of research exists investigating tumor growth phenomena from sub-cellular (e.g., chemical transmitters) over cellular (e.g., cell proliferation rates) to macroscopic scales (e.g., biomechanical behavior of complete tumors). Varying objectives are followed, ranging from understanding the influence of individual substances to the prediction of future development for radiation treatment planning. A significant fraction of the work aims at modeling growth processes of malignant tumors in the human brain (e.g., [109, 256]). A complete review of this research field is out of the scope of this book. Nevertheless, the main directions followed in the past will be briefly examined. For more details the reader is referred to the excellent review given in [7].

Inspired by early findings on diffusion in tissues [120], one of the first mathematical formulations of tumor growth was presented in [44]. A diffusion model was used to model the oxygen distribution and the resulting growth of a spherical solid tumor. This work was later on extended in [108] by including adhesion and surface tension between cancerous cells. In the proposed model the evolution of the tumor radius is described by differential equations, while the concentration of nutrients and associated inhibitors is modeled by a reaction-diffusion equation. An important addition to these models is the modeling of metastatic processes. Based on experimental findings, a stochastic model of metastases formation was proposed in [164], which used a nonhomogeneous two-dimensional Markov model to predict development of metastatic cells. The influence of growth inhibitors on tumor development was later on also integrated into modeling approaches. To this end, in [1] a spatially-dependent mitotic control function was included. The last two decades have exhibited an exponential growth of enhanced mathematical tumor modeling methods. This includes for instance work on cell migration in multicell spheroids [181], angiogenic factors [51], multiphase models [215], and residual stress formation [5].

Several microscopic models of tumor growth do not take into account the interactions between cancerous cells and surrounding tissues. In order to link the microscopic proliferation with macroscopic effects, several groups examined the use of cellular automata. Initial work in this direction was reported in

[220]. A phenomenological description of tumor growth was developed based on the Gompertz growth law. The latter was introduced in 1825 and describes growth processes, which are initially exponential, but show asymptotic behavior later on. Also, in [247] a cellular automaton was applied to examine the morphological patterns obtained by the growth of several tumor seed cells. A 3D cellular automaton to describe the growth dynamics of a brain tumor was investigated in [137]. The underlying lattice is a Delaunay triangulation with varying element density, thus implicitly including growth speed. An extended model including normal and tumor cells, as well as necrotic space and microvessels is presented in [205]. Extracellular concentration is influenced by microvessels, thus steering the evolution of the automaton.

Further work deals with macroscopic, mechanical models of tumor growth focusing on appropriate modeling of internal pressures and surrounding tissue deformation, e.g., [287], [151], and [56]. 2D finite element analysis is performed on the brain anatomy using nonhomogeneous material properties.

The aforementioned approaches to modeling of tumor growth capture several aspects of the underlying process. However, they can usually not be applied to surgical scene generation. Problems arise due to the often significant computation time, different model representations, or unknown material parameters. Nevertheless, some of the presented concepts provide a starting point from which application-specific methods will be developed.

2.3 Data Acquisition

2.3.1 Overview

The first step in the geometry generation chain is the radiological acquisition of the in vivo organ shape. This aims at building a database of several uteri to provide a ground truth for the statistical analysis of the healthy anatomy. It should be noted that due to the considerable number of possible pathological alterations, as well as the large range of their manifestations, a similar statistical approach for pathology generation is infeasible.

Several medical imaging modalities are available to obtain the necessary 3D anatomical data. The most suitable techniques for this task are computer tomography (CT), magnetic resonance imaging (MRI), and 3D ultrasound (US) imaging. Table 2.1 provides an overview of the main characteristics of these methods, while their main principles are briefly reviewed in the following.

Computer tomography. The underlying theoretical principle of this X-ray-based imaging technique is the reconstruction of an object from its projections. Based on work of Johann Radon in 1917, a tomographic reconstruction algorithm was proposed in 1960 by Allan Cormack. His findings provided a solution to the problem of superposition in conventional X-ray imaging. Independently of this work, Godfrey Hounsfield developed

Table 2.1. Short overview of medical imaging modalities

	CT	**MRI**	**US**
Imaging principle	Attenuation of electro-magnetic radiation	Spin relaxation in nuclear magnetic resonance	Reflection of sound waves
Physical properties	Photon beams 10-100 keV, effective radiation dose 0.2-13 mSv	Magnetic field 0.2-9 T, radio frequency pulses 10-100 MHz	Sound waves 1-12 MHz
Typical resolution	$0.33\,\text{mm}^3$ isotropic	plane $0.6\,\text{mm}^2$, slice 2 mm	axial 0.5 mm, lateral 1.0 mm
Drawbacks	Ionizing radiation, large amount of data, cost	Motion artifacts, strong magnetic field, acoustic noise, cost	Limited resolution & penetration depth, image noise and distortion, bone & air interference
Important dates	Introduced 1968, Nobel prize 1979 (Hounsfield & Cormack)	NMR discovered 1945, Nobel prize 1952 (Bloch & Purcell); MRI principles introduced 1973, patented 1974 (Damadian), Nobel prize 2003 (Lauterbur & Mansfield); Nobel prize 1991 for NMR spectroscopy (Ernst)	Introduced 1940s

an approach on his own at the end of the 1960s using the so-called inverse Radon-transformation. Moreover, he constructed the first clinically usable CT scanner with which in 1972 he presented cross-sectional images of the human brain. Thereafter, the applied principles and technologies have been continuously improved over several generations of scanners. Multi-detector-row tomographs available today are capable of acquiring isotropic imaging data in a few seconds during a single breath-hold. In current medical practice, CT has become an indispensable imaging tool and is the gold standard in the diagnosis of several pathologies. A typical CT image is shown in Figure 2.10(a).

Magnetic resonance imaging. This tomographic technique reconstructs images based on the magnetic resonance of atomic nuclei. In 1945 Felix Bloch and Edward Purcell discovered that nuclei with non-zero spin numbers exhibit magnetic properties and align themselves parallel to a strong

(a)   (b)   (c)
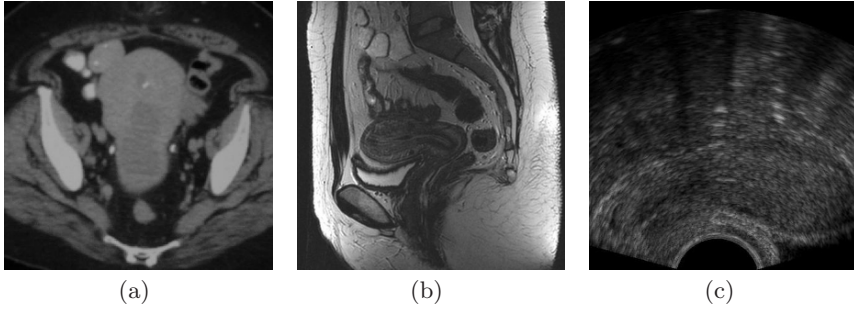
Figure 2.10. Uterus region displayed using various medical imaging modalities. (a) Transverse CT image of pelvic region, (b) Sagittal MRI image of abdomen, (c) US scan of uterus

uniform magnetic field. Applying radio frequency (RF) pulses in a plane parallel to the magnetic field causes nuclei to temporarily assume a non-aligned high-energy state. After the RF excitation, the nuclei return to equilibrium. During this process, energy is emitted which can be recorded and used for imaging purposes. Raymond Damadian recognized the potential of the nuclear magnetic resonance (NMR) and suggested in 1971 its use for cancer detection. In 1973 Paul Lauterbur suggested the addition of a gradient field to enable the spatial location of signals. Moreover, he applied back-projection methods to create the first MRI images. Peter Mansfield provided the mathematical analysis of the gradient fields. Medical MRI usually obtains images based on the relaxation process of excited hydrogen nuclei in soft tissue. Several properties can be used to generate images, including proton density, spin-grid relaxation (T1), spin-spin relaxation (T2), and spin-echo techniques without refocusing (T2*). A complete coverage of this imaging modality is beyond the scope of this work; the reader is referred to the related literature, e.g., [30, 292]. An image of an MRI cross-sectional scan is depicted in Figure 2.10(b).

Ultrasound imaging. The detection of ultrasound waves reflected at borders of changing acoustic density is the key component of sonography. Sound pulses are usually created by a phased array of piezoelectric transducers. The physical principles of this technology, such as the piezoelectric effect or the sound propagation in materials were discovered in the 19th century. The first application of ultrasound in medicine focused on therapeutic applications of high intensity pulses, for instance in neurosurgery as an attempt to treat Parkinson's disease. In the early 1940s Karl Theo Dussik created the first ultrasound images. Since then medical sonography has evolved into the second most utilized medical imaging modality in diagnostics. Although the technique is relatively safe and inexpensive, the imaging quality is rather low due to noise and interference effects. A typical US scan is visualized in Figure 2.10(c).
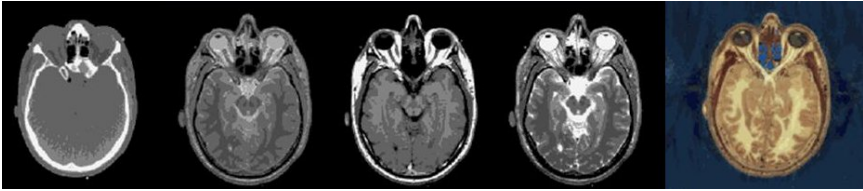
Figure 2.11. Cross-sections of Visible Human male (CT, MRI modes, cryosection – courtesy of Visible Human Project, US National Library of Medicine)

Apart from the standard medical imaging modalities, the previously mentioned acquisition of anatomical cross-sections from cadavers should also be scrutinized. While such a data source is not appropriate for acquiring the necessary instances for a statistical shape analysis, it nevertheless has been used to obtain the organ models in several surgical simulation endeavors – including the laparoscopic simulator system developed in our lab [259]. In order to obtain the Visible Human data, the donated bodies of a 39-year old male and a 59-year old female were first imaged using CT and MRI in the unfrozen state. Thereafter, the bodies were fixated using a foaming agent as well as gelatin, and then frozen. The cryomacrotomy was performed with a specialized milling device, shaving off individual slices of the cadaver block at 1 mm and 0.33 mm intervals for the male and female specimen, respectively. After each cut, color photographs of the top of the blocks were taken. This approach resulted in approximately 1,700 axial color images of high resolution – 0.33 mm^2×1 mm for the male and 0.33 mm^3 for the female dataset. Further details of the acquisition process concerning the Visible Human male are described in [251]. Images of the male dataset obtained with different imaging modalities are shown in Figure 2.11.

The detailed cross-sectional data of the Visible Human female were used as the basis of an exemplary model of the female abdomen and pelvis for the laparoscopic simulator. More than 300 structures of interest, including organs, blood vessels, bones, muscles, and ligaments were outlined in the images using semiautomatic segmentation, and were thereafter three-dimensionally reconstructed [12]. The resulting segmented model of the internal female genital organs and ligaments is shown in Figure 2.12. However, in order to obtain a statistical shape model a larger number of samples is needed. The respective acquisition process will be outlined in the following.

2.3.2 Uterine Image Acquisition

Medical datasets were obtained from 26 volunteers. These were selected to cover the range in age and number of pregnancies that is usually encountered in patients undergoing hysteroscopy. Ten of the participants were older than 40 years, 12 without children, and 7 postmenopausal. Because of the danger of radiation, CT was ruled out for image acquisition. Nevertheless, MRI as well
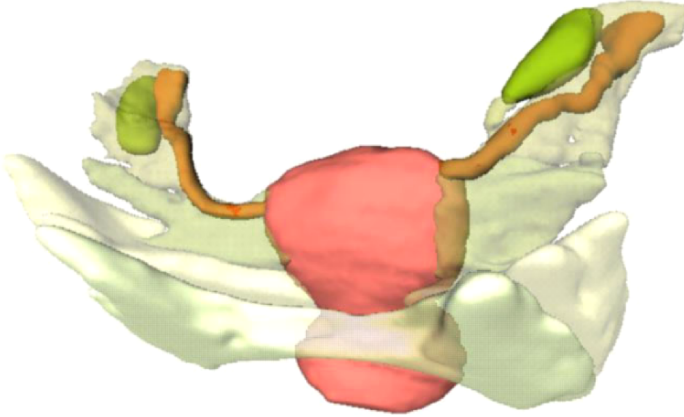
Figure 2.12. Ventral view of the Visible Human female uterus (red), uterine tubes (orange), and ovaries (olive). Ligaments are rendered semitransparent

US data were obtained. The MRI scans consisted of 16–25 slices in the sagittal and transversal planes, respectively. In addition, scans were also carried out with an oblique orientation aligned with the uterine corpus. Imaging was performed with a GE 1.5 Tesla machine yielding a resolution of $0.46\,\mathrm{mm}^2 \times 3\,\mathrm{mm}$. Scanning time was less than 10 mins due to the small number of slices, thus avoiding motion artifacts. Moreover, Buscopan was administered to further minimize noise due to gastrointestinal activity. Nevertheless, the filling of the bladder led to some intrasubject variability between scans. However, this was found to be negligible. In addition, transvaginal ultrasound images were captured in 2D and 3D. However, due to deformation induced by the probe, as well as the generally low image contrast, these scans were not used for the shape analysis of the healthy anatomy.

2.4 Healthy Anatomy Generation

2.4.1 Outline

The underlying idea of the geometry generation of healthy organs is based on the derivation of a statistical model describing the natural variability of the organ shape. The main target is to find, for a given population of object instances, the shape space which most compactly describes the variations occurring in the sample dataset. Moreover, to allow scene definition by physicians, an intuitive model instantiation process has to be provided. Therefore, a constrained optimization has to be included which provides shapes matching user-controlled intuitive parameters.

Under the assumption that the variability of uterus shape descriptors is Gaussian and the dependencies of the shape parameters are only up to second order, the principal component analysis (PCA) can be applied to decorrelate the parameters and reduce the dimensionality of the description. Linear transformation maps the data to a new coordinate system, thus providing a linear vector space spanned by a complete set of orthogonal basis vectors in which the main variations of object shape are separated. This technique has been applied in several research areas, for instance focusing on segmentation of medical imagery [253, 277, 140]. A number of approaches can be followed to represent organ shape by a parametric model. Using the vertex coordinates of polygonal object representations leads to the so-called point distribution model (PDM) [57, 50, 123].

A key requirement to obtain meaningful results from the PCA is the correspondence between the shape parameters of the individual instances of the population. In the case of PDMs, vertices located at the same anatomical feature should have the same index in all instances of the training set. Moreover, all training samples have to be expressed in the same coordinate system. Several methods have been proposed to establish the described correspondences. In the spherical harmonic object representation described in [34], the correspondence is implicitly defined. To obtain the latter, the uniform area parameterization and the coordinate system is aligned with the axes of the first-order ellipsoid of the object. Another approach is to perform a global optimization of the determinant of the covariance matrix of the study population, as described in [146]. In [66] a strategy inspired by information theory has been followed. The idea of this technique is to find correspondences that result in an optimal model in terms of generalization ability, compactness, and specificity. An objective function is defined for the genetic optimization, describing the quality of the statistical shape model. While these methods are fully automatic, a semiautomatic paradigm can also be followed, focusing on initialization of subdivision surfaces by a user, based on generalized landmarks [255]. Directly ensuring point correspondences during the necessary segmentation process results in a statistical shape description of reasonable quality. Therefore, this approach has been followed for the statistical model of the healthy uterus [242].

2.4.2 Segmentation

In order to maintain point correspondences, a coarse-to-fine subdivision scheme is applied, which ensures similarity of vertex indices between different segmentations.

The shape extraction process starts with the rigid registration of the MRI datasets in order to ensure that the slices are aligned in space with sufficient accuracy [111]. Three main strategies can be followed to this end: landmark-based semiautomatic registration, surface-based registration of segmented shapes, and intensity-based registration. We apply the latter technique,
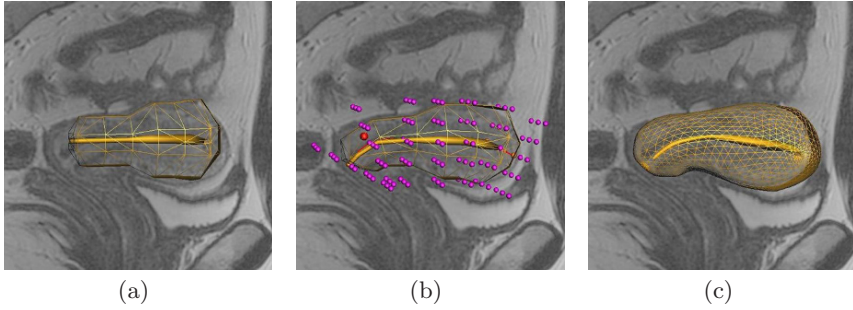
Figure 2.13. Steps of uterus segmentation. (a) Coarse mesh for initialization, (b) Free-form deformation, (c) Final segmentation after subdivision and smoothing

obtaining the rigid transformation based on the optimization of mutual information as introduced in [175]. First, the mutual histogram of two volumes is calculated. This allows us to obtain the mutual information based on the marginal probabilities of the histograms. In the registration algorithm the mutual information is then maximized by searching for the optimal rigid body transformation. It has to be mentioned that due to bladder filling the registration of intrapatient volumes is in principle nonrigid; however, the observed changes were mostly below scan resolution and thus negligible.

Thereafter, the segmentation process is carried out in a semiautomatic fashion. Prior knowledge about the anatomical shape of the uterus allows us to provide a coarse surface mesh as an initial guess for the organ shape. This root mesh has been manually designed and contains openings for the cervix as well as the tubal ostia (Figure 2.13(a)). Thus, from a topological point of view, the surface is represented by two concentric spherical shapes connected by three short cylindrical tunnels. By choosing mesh dimensions according to standard anatomical measurements of the uterus, a reasonable initialization can be achieved for the subsequent steps. The initial guess is refined using a combination of interactive segmentation techniques – global free-form deformation, local mesh adaptation, and mesh optimization. The first permits the deformation of a control mesh of user-defined size to globally adapt the root mesh. This allows us to coarsely align the initial guess with the actual shape of the uterus (Figure 2.13(b)). The second technique enables modification of local sections of the segmentation mesh. This is useful to displace vertices of the sparse mesh to match characteristic anatomical landmarks of the uterus. Finally, mesh optimization can be carried out to remove C^1 discontinuities on the surface due to the manual modifications. Apart from these tools, the initial coarse mesh can be further subdivided following a quaternary scheme, during which correspondence between vertices is maintained (Figure 2.13(c)). By combining all these steps, triangular surface meshes of the uteri have been segmented from all MRI datasets. Depending on the quality of the medical
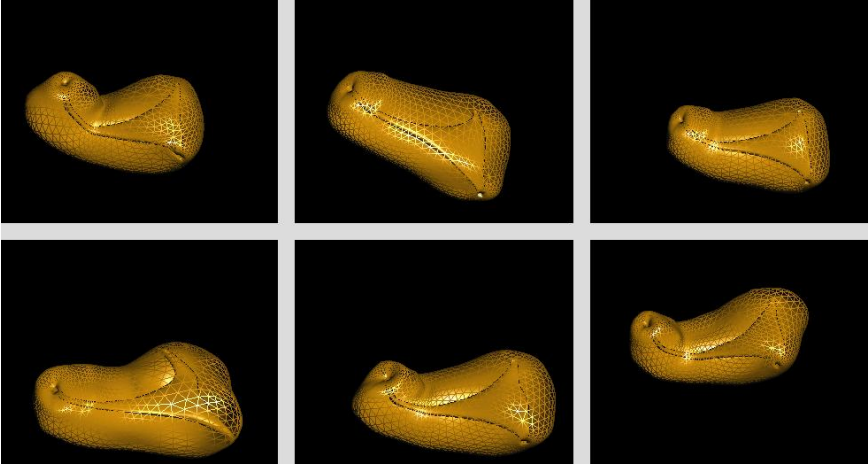
Figure 2.14. Example meshes of uteri segmented from MRI datasets

data and the skill of the user, the segmentation can be carried out in several minutes. A subset of the segmented shapes is depicted in Figure 2.14.

2.4.3 Statistical Model

Before the statistical analysis can be performed, the instances of the study population have to be transformed into a common coordinate system. The three natural orifices of the uterine cavity serve as a basis for defining the right-handed coordinate system. The origin of the system is given by the right tubal ostium. The axes are rotated such that the x-axis passes through the cervical ostium and the left tubal ostium lies on the xy-plane in the negative y direction. The z-axis is then determined by the cross-product of the other axes. Apart from defining a common coordinate frame, no spatial normalization has been applied.

After this step, the statistical model is obtained by employing the principal component analysis (PCA) on all object instances in our database. The used population of object instances consists of the 23 segmented surfaces of the uteri. For N instances, given as polygonal models with M vertices $\mathbf{p}_i = [x_i^{[1]}, y_i^{[1]}, z_i^{[1]}, \ldots, x_i^{[M]}, y_i^{[M]}, z_i^{[M]}]^T$, the parameter signals of the shapes are centered by calculating the average model $\overline{\mathbf{p}}$

$$\overline{\mathbf{p}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{p}_i \tag{2.1}$$

and an instance-specific difference vector

$$\Delta \mathbf{p}_i = \mathbf{p}_i - \overline{\mathbf{p}} \,. \tag{2.2}$$
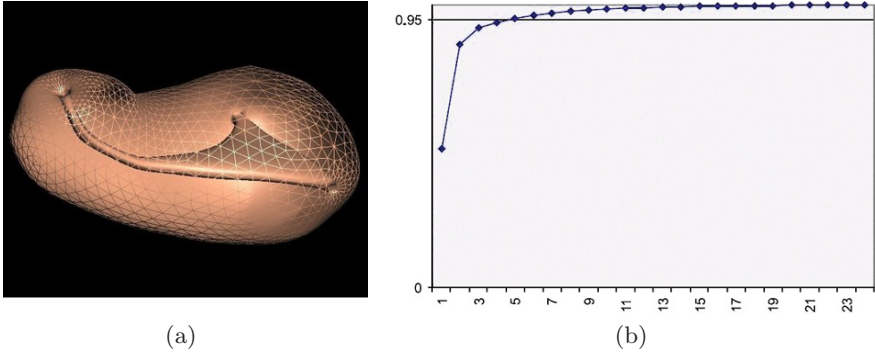
Figure 2.15. Statistical model of natural variability. (a) Mean uterus shape, (b) Cumulative histogram of eigenvalues

The covariance matrix $\mathbf{\Sigma}$ and the resulting PCA given by the eigensystem of that matrix can subsequently be determined according to

$$\mathbf{\Sigma} = \frac{1}{N-1}\sum_{i=1}^{N} \Delta\mathbf{p}_i \Delta\mathbf{p}_i^T \overset{\text{PCA}}{=} \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T ; \qquad (2.3)$$

with $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_{N-1})$ containing the eigenvalues. A new instance can then be derived by a combination of weighted eigenshapes and the mean shape.

$$\tilde{\mathbf{p}} = \overline{\mathbf{p}} + \sum_{j=1}^{k} b_j \mathbf{U}_j \qquad (2.4)$$

In general the first $k \ll N$ eigenshapes corresponding to the largest eigenvalues capture most of the shape variability. Thus, the linear combination of a small number of eigenshapes results in a compact description of the population. The shape of a new instance is fully defined by the parameter set $\mathbf{b} = \{b_1, \ldots, b_k\}$. In Figure 2.15 the mean shape of the uterus as well as the influence of the eigenvalues is shown. The first five eigenshapes capture 95% of the overall variability.

The trade-off between tractability and accuracy of the model description is influenced by the number of parameters k. Unfortunately, the shape changes obtained when varying the individual parameters b_j do not necessarily map to intuitive anatomical metrics, thus making it complicated to control the object appearance based only on these parameters. Instead, it should be possible to control the shape changes by adjusting standard anatomical metrics, e.g., length, width, or depth of the uterine fundus. To achieve this, an approach of progressive elimination of variation can be followed. This technique was initially introduced in [123] in the context of medical image segmentation. The underlying idea is the fixation of control points at anatomical landmarks,

followed by the derivation of a shape in variation space with minimal deviation from the mean of the model.

In the context of PCA, this means choosing the model with minimal Mahalanobis distance D_m. One solution to this task is to find three basis vectors $\mathbf{r}_{x_j}$, $\mathbf{r}_{y_j}$, and $\mathbf{r}_{z_j}$ in variation space describing decoupled x-, y-, and z-translations of a given point j in object space with minimal overall variation. Once these vectors are found, all possible boundary conditions can be satisfied by adding these appropriately weighted vectors to the mean shape. The Mahalanobis length of the vectors is given by

$$D_m(\mathbf{r}_w) = (\mathbf{U}\mathbf{r}_w)^T \boldsymbol{\Sigma}^{-1} \mathbf{U}\mathbf{r}_w = \mathbf{r}_w^T \boldsymbol{\Lambda}^{-1} \mathbf{r}_w, \quad w \in \{x_j, y_j, z_j\}. \tag{2.5}$$

The unit-translation vectors can now be determined by minimizing D_m. The necessary constrained optimization can be carried out using Lagrange multipliers. The objective function is given by

$$L(\mathbf{r}_w, \mathbf{l}_w) = \sum_{e=1}^{N-1} \frac{\left(r_w^{[e]}\right)^2}{\lambda_e} - \mathbf{l}_w^T \left[\mathbf{U}_j \mathbf{r}_w - \mathbf{e}_w\right], \quad w \in \{x_j, y_j, z_j\} \tag{2.6}$$

with $\mathbf{e}_w$ being unit translations into x-, y-, and z-directions. Moreover, submatrix $\mathbf{U}_j$ contains those three rows of $\mathbf{U}$ that the coordinates x_j, y_j, and z_j depend on. Solving the system yields the basis vectors $\mathbf{r}_w$ according to

$$\mathbf{R}_j = \left[\mathbf{r}_{x_j}\,\mathbf{r}_{y_j}\,\mathbf{r}_{z_j}\right] = \boldsymbol{\Lambda}\,\mathbf{U}_j^T \left[\mathbf{U}_j\,\boldsymbol{\Lambda}\,\mathbf{U}_j^T\right]^{-1}. \tag{2.7}$$

Thus, the most probable shape $\check{\mathbf{p}}$ given the displacement Δ_j between actual and mean landmark j is given by

$$\check{\mathbf{p}} = \overline{\mathbf{p}} + \mathbf{U}\mathbf{R}_j\Delta_j = \overline{\mathbf{p}} + \mathbf{U}\boldsymbol{\Lambda}\,\mathbf{U}_j^T \left[\mathbf{U}_j\,\boldsymbol{\Lambda}\,\mathbf{U}_j^T\right]^{-1}\Delta_j. \tag{2.8}$$

After obtaining the most probable shape for a given landmark, it has to be ensured that subsequent modifications do not alter the adjusted vertex. Therefore, the components from the statistic that cause a displacement of the point have to be removed. To this end the basis vectors are subtracted from the parametric representation of each instance i, weighted by the landmark specific displacement.

$$\mathbf{b}_i^{\hat{j}} = \mathbf{b}_i - \mathbf{R}_j\Delta_j = \mathbf{b}_i - \mathbf{R}_j\,\mathbf{U}_j\,\mathbf{b}_i, \qquad \forall i \in \{1, \ldots, N\}. \tag{2.9}$$

Doing so for all instances yields a new description of the population $\mathbf{b}_i^{\hat{j}}$ which is invariant with respect to point j.

This technique can now be extended to satisfy the requirements in the surgical scene generation process. The prediction scheme has to be modified so that anatomical distances in the uterus can be used as constraints for

the optimization. The predictors are selected as the squared measurements of length, width, and depth at the uterine fundus and cervix, respectively, as well as the overall length from cervical ostium to fundal tip. Variation of the latter value allows one to control the angle between cervix and fundus. This leads to a nonlinear prediction scheme, in which l Euclidean distance constraints ξ_r between landmark vertices $\mathbf{v}^{[i]_r}$ and $\mathbf{v}^{[j]_r}$ have to be fulfilled. Thus, the most probable shape has to be found which meets the constraints

$$\left(D_e(\hat{\mathbf{p}})\right)^2 = \|\mathbf{v}^{[i]_r} - \mathbf{v}^{[j]_r}\|^2 = \xi_r, \quad r = 1, \ldots, l \tag{2.10}$$

and exhibits the smallest Mahalanobis distance from the mean $D_m(\bar{\mathbf{p}} - \hat{\mathbf{p}})$. Again, this constrained minimization problem can be solved using Lagrangian optimization. However, two problems are encountered in the current formulation. The number of variables $(3M)$ is much larger than the number of observations (N). Due to the linear dependency of the difference vectors $\sum_{i=0}^{N} \Delta\mathbf{p}_i = 0$, the covariance matrix is rank-deficient. In addition, due to the large number of variables, i.e., mesh vertices, the numerical methods become quite slow and unstable. To avoid this, the number of variables can be reduced by representing all population instances in a reduced linear subspace spanned by the mean shape and a limited number of eigenshapes. Thus, the transformed shape representation is given by

$$\mathbf{q}_i = U^T \Delta\mathbf{p}_i, \quad i = 1, \ldots, N. \tag{2.11}$$

The goal function of the minimization problem in the new subspace becomes

$$f(\mathbf{q}) = \mathbf{q}^T \mathbf{\Lambda}^{-1} \mathbf{q}, \tag{2.12}$$

with the new constraints

$$\left(D_e(\bar{\mathbf{p}} + \mathbf{Uq})\right)^2 = \xi_r, \quad r = 1, \ldots, l. \tag{2.13}$$

Now, Lagrangian optimization can be applied in the reduced subspace to obtain the best fitting parameter vector fulfilling the given constraints. The Lagrange function has the form

$$F(\mathbf{q}, \mathbf{l}) = f(\mathbf{q}) - \mathbf{l}\left(D_e(\bar{\mathbf{p}} + \mathbf{Uq})^2 - \xi_r\right). \tag{2.14}$$

A numerical solution to this problem can be determined using the Newton-Raphson method. After retrieving the optimal parameter vector $\hat{\mathbf{b}}$, the corresponding organ shape can be determined by projecting back into the full point distribution space:

$$\hat{\mathbf{p}} = \bar{\mathbf{p}} + \mathbf{U}\hat{\mathbf{b}}. \tag{2.15}$$

(a)	(b)	(c)

(d)	(e)	(f)
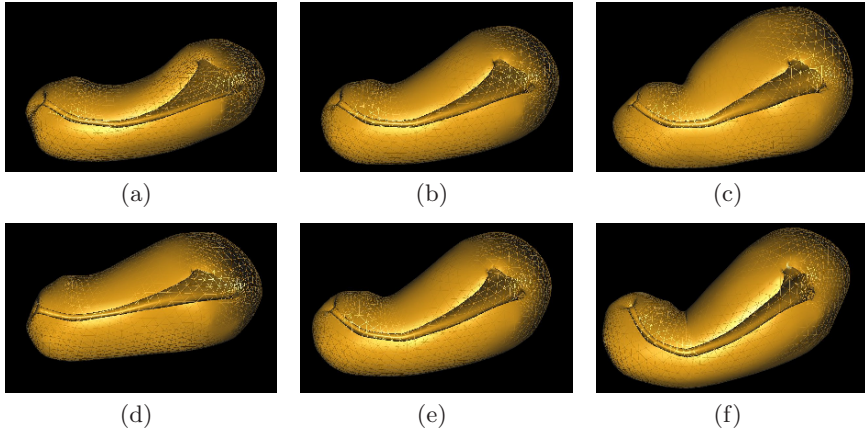
Figure 2.16. Creation of new instances according to variation of two different predictors. (a) Fundus depth = 25 mm, (b) Fundus depth = 36 mm, (c) Fundus depth = 49 mm, (d) Fundus length = 44 mm, (e) Fundus length = 52 mm, (f) Fundus length = 61 mm

2.4.4 Shape Prediction

Based on the described process, a tool for clinical users allowing the intuitive derivation of new healthy organ instances according to anatomical measurements has been created. The selection process is controlled by manually adjusting the seven anatomical predictors mentioned above. Due to the fast convergence of the Newton-Raphson scheme, the resulting shapes are obtained, and thus also displayed, in real time, providing a convenient interaction environment to the user. The underlying statistical analysis of the uterus shape database has to be carried out before the derivation of new instances can be performed. However, this offline step has only to be done once and is performed in a few seconds on standard PC hardware. Figure 2.16 shows the variation of the fundus depth and length, respectively. Since the overall length of the uterus is kept constant, an increasing angle is introduced in the latter case between cervix and fundus to accommodate for the increasing fundus length.

While the described tool allows the convenient definition of new instances by nonexpert users, the question remains how well the obtained shape description captures the natural variation of the healthy uterus. According to [66], three main criteria should be met by a shape model: generality, compactness, and validity.

The first term denotes the ability of the model to represent any instance of the examined shape – i.e., it should not be limited to only those in the training set. In order to evaluate the generality of the description, a leave-one-out experiment has been performed. To this end, one uterus instance of the organ database at a time is excluded from the statistical analysis. The shape model thus obtained is then tested by deriving a new instance according to the

anatomical measurements of the excluded shape. The quality of the obtained approximation of the original is then examined according to an error metric, which is based on distances of corresponding mesh vertices after rigid shape registration. The leave-one-out experiment for the uterus dataset resulted in the worst case in a maximum vertex distance of 12.44 mm, while the mean errors of the individual experiments ranged from 1.12 mm to 3.81 mm.

In this context it has to be pointed out that the proposed process does not necessarily provide an optimal approximation of the excluded shape. This is due to the fact that the predicted shape has a minimal Mahalanobis distance to the mean shape, but not necessarily to the projection of the excluded shape into the spanned subspace. This is due to the imposed nonlinear shape constraints, which can yield a shape with minimal distance to the projected original, but suboptimal distance to the mean shape. Nevertheless, the difference between these instances is usually small and within the order of magnitude of the projection of the original shape. Thus, it can be assumed that in the reduced subspace the excluded shape can be approximated sufficiently precisely.

The second characteristic refers to the ability of the model to represent shapes with a minimal set of parameters. This is assessed by obtaining the statistical model using all shape instances and quantifying the remaining variability after specification of an increasing number of shape constraints. To this end, the variability is determined according to maximum, minimum, mean, and variation of the corresponding vertex distances, averaged over the database. In case of the uteri it is observed that the length metric has the largest influence on shape variability. The eigenvector associated with this predictor, obtained from the eigen decomposition of the predictors, already accounts for 80% of the overall variability. For completeness, it should be mentioned that none of the eigenvectors is sufficiently well aligned with the descriptors, thus they cannot be used for the envisioned intuitive interaction system.

The final attribute focuses on the validity of the instances generated with the shape model. Not all combinations of predictor values yield a valid shape; for instance, variation of the fundus or cervix width predictor independently of the others results in anatomically invalid shapes. Additional adjustments of the remaining metrics have to be done to alleviate the situation. This is also an indication that there is still some interdependence left between the predictors. Nevertheless, to determine the model validity, an assessment of randomly generated uterus instances by medical experts has been undertaken. This study indicated that the Mahalanobis distance D_m with respect to the eigenspace of the predictors can be used as an indicator of shape validity. New instances with distance $D_m < 16$ from the mean were usually classified as valid. Therefore, this measure can be used as a threshold during instance creation. As mentioned, not all combinations of predictors provide a valid shape during interactive manipulation by a user. Thus, in such a case, the

last valid result is maintained during interaction until the threshold is satisfied again.

2.5 Pathology Integration

2.5.1 Outline

After the generation of new instances of the healthy organ, the pathological variation needs to be added. Although a number of conditions warrant a hysteroscopic intervention, at the current stage the focus of our work is on benign tumors – polyps and myomas. Both are a cause of abnormal bleeding and their suspected presence is a common indication for hysteroscopy. While there is some similarity between their visual appearance, the medical treatment is quite different. While the resection of a polyp at its stem is a viable approach, the same technique applied to a pedunculated myoma would cause major complications. Therefore, it is necessary to correctly identify the type of tumor and follow an appropriate surgical procedure.

Myomas consist of dense fibrotic tissue, causing them to grow relatively independently of the surrounding tissue, keeping a spherical shape. Their size ranges from a few millimeters to several decimeters. With respect to the uterine wall they can be confined to the myometrium, grow into the uterine cavity, and in extreme cases prolapse through the cervix into the vagina. A myoma is described in medical terms according to its location in the uterus (fundus, corpus, cervix), degree of protrusion into the cavity (type 0, I, II), and the lengths of the main axes. In contrast to this, polyps originate from a benign overgrowth of endometrial tissue, usually protruding into the uterine cavity. They are softer than myomas and reach sizes between a millimeter and a few centimeters. The medical description of polyps is usually done according to their location (cervix or fundus). However, size and shape are usually not classified.

With regard to the scene generation process, there are a number of requirements which have to be met. The main aim is to generate a realistic shape of the tumors – optimally in an automatic process according to medical terms. Moreover, to accommodate for repeated training, some degree of randomness during the generation process has to be included. In addition to this, during the model development, it should be possible to track and/or include further information, such as surface textures, blood perfusion, and biomechanical properties. Finally, the seamless integration into the healthy organ model also has to be ensured. The focus of related work has mainly been on aspects which are less critical for surgical simulation, such as exact interaction on the cellular level, patient-specific modeling of growth, or observation of the actual development process.

In order to generate pathologies, three approaches have been developed, differing in level of interaction and control, realism, computation time, and degree of variation, which are discussed in more detail below.
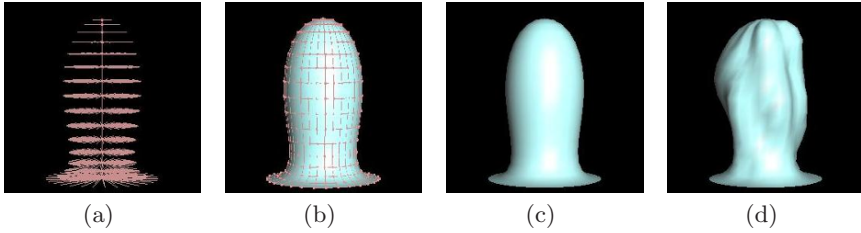
Figure 2.17. Skeleton-based design of a polyp. (a) Medial axis with spokes of mesh vertices, (b) Elements of 4-connected surface mesh, (c) Generated base polyp, (d) Distortion applied to mesh

2.5.2 Skeleton-Based Design

The first pathology generation approach follows the strategy of interactive design based on the notion of shape characterization by local symmetry and axial growth. The application of these concepts to the description of biological objects has been a focus of research for several decades. At the beginning of the last century the first studies of the laws and diversity of biological shape were carried out in [266]. While being mainly descriptive, the work recognized the role of axial growth processes in the formation of the shape of living objects. These ideas were later formalized in [29], providing a rigorous mathematical framework of shape description. It was demonstrated that segments of symmetry axes and corresponding radius functions form a versatile shape taxonomy. As discussed earlier, skeletons have been successfully applied to the simulation of growth processes of a number of human organs [77]. Also, for the design of the pathologies under scrutiny, the description via skeletons and corresponding radii can be utilized [239].

The shape of a myoma or polyp can be described by a collection of connected linear segments, with possible branching at the end of the skeleton, as well as associated radii around the medial axis. As a simplification, the medial axis is composed of only one linear segment in the pathology generation tool; however, the discussed design mechanism can easily be extended to a more complex composition. In the first phase an initial base shape is obtained, which can later be further modified by the user.

The design process starts with the selection of the type of pathology to be generated. Based on images of real pathologies as well as medical descriptions, piece-wise profiles have been heuristically defined, which capture the typical shapes of myomas or polyps. Depending on the type of pathology, the profiles show characteristic differences – for instance, myomas having a spherical outline, and polyps having an elongated stalk. Thereafter, the user selects the desired height of the tumor. As mentioned, the profile of the pathologies is made up of a different number of sections. Depending on the user-defined height, varying profile sections are included, thus giving the pathology a characteristic outline matching its specified size. This for instance allows one to
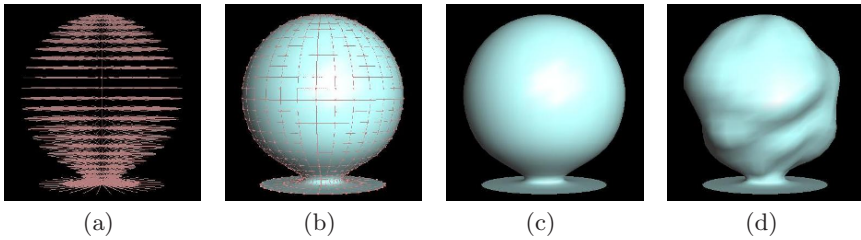
Figure 2.18. Skeleton-based design of a myoma. (a) Medial axis with spokes of mesh vertices, (b) Elements of 4-connected surface mesh, (c) Generated base myoma, (d) Distortion applied to mesh

differentiate between sessile and pedunculated myomas. After the shape of the pathology has been selected, further adjustments can be made by changing the width/depth ratio. The result of this phase is the definition of the pathology surface by a 4-connected grid of vertices. The latter are obtained by a regular discretization of the profile surface. In a final step, randomness can be introduced by distorting the obtained surface. To this end mesh vertices are randomly displaced along their normal directions. Thereafter, to obtain smooth variations, a Gaussian filter is applied to the surface. Moreover, to avoid large perturbations at the peduncle, the distortion is additionally scaled by the radius at the respective skeleton position. Figure 2.17 shows an example polyp designed with the described approach, while Figure 2.18 depicts the result for a myoma.

After the mesh of the pathology has been defined, it needs to be merged with that of a healthy uterus. To this end, first a vertex of the uterus mesh is selected, where the pathology should be located. The tumor shape is then reoriented to parallel the vertex normal, and translated to the specified position. Thereafter, all vertices of the cavity mesh whose distance to the selected tumor position is smaller than the radius of the pathology base are removed, thus leaving a hole in the mesh. In the next step, the resolution along the border of the coarser mesh is refined, such as to match that of the other rim. Next, the gap between the two mesh borders is triangulated, thus creating links between both meshes. The final step is the smoothing of the transition zone by the optimization of a local mass-spring system. To this end, the edges of the pathology as well as the newly created links are considered as linear springs. The individual steps of this process are shown in Figure 2.19.

This approach allows the interactive generation of new myomas and polyps by novice users in a few seconds. Unfortunately, the underlying methodology relies on heuristic derivation of the tumor shape and does not cover any physiological details of the formation process. Moreover, the approach does not take into account any interaction with the surrounding tissue. While this could be integrated belatedly by modifying the meshes after the placement of the generated pathology, the influence during growth itself would not be accounted

(a)   (b)   (c)   (d)
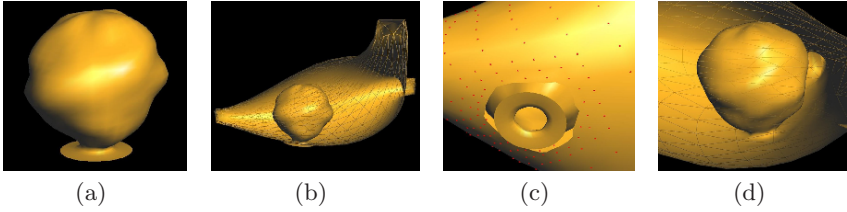
Figure 2.19. Merging example of myoma with uterine cavity. (a) Generated myoma mesh, (b) Transformation of myoma mesh to user defined position, (c) Deletion of neighboring vertices/triangles, (d) Integrated mesh after triangularization and smoothing

for. Nevertheless, the full control of the generation process at interactive rates is a desired characteristic from the point of view of physicians who want to design a specific training scenario. Nevertheless, as an alternative, the tumor growth process itself can be modeled to obtain the desired pathologies. To this end growth models can be used, which will be discussed in the following.

2.5.3 Cellular Automata Growth Model

The modeling of tumor growth with cellular automata has been an active topic of research for more than a decade (see, e.g., [220, 287, 137]). However, existing work usually follows different targets than scene generation for surgical simulation. From a theoretical point of view, cellular automata were initially introduced in the late 1940s by John von Neumann [282]. However, the first practical application of the theory is probably the renowned *Game of Life* by John Horton (see, e.g., [184]). In general, cellular automata are used for modeling discrete dynamical processes without explicitly relying on differential equations. An advantage of cellular automata is their simplicity and extendability. The growth process can be described based on a small number of rules acting on a regular lattice. In addition, all intermediate stages of the evolution can be observed if desired. Moreover, simulation with cellular automata is intrinsically stable. The modelling of myoma growth with such a system [240] is outlined in this section. It should be noted that the number of actual cells in a myoma is too large to be handled at interactive rates. Therefore, individual cells are not simulated, but instead conglomerations of cells. Thus, in the following the term "cell" will denote a node of the cellular automaton, and not a single biological cell.

Individual cells of a cellular automaton are located at positions $\mathbf{p} = (x, y, z)$ on a regular, three-dimensional cubic lattice $\mathcal{L}$. At a specific time step t during the simulation, each cell at position $\mathbf{p}$ has two characteristic states associated with it – a tumor and a tissue state. The former is described by the discrete value $S_{tumor}^t(\mathbf{p}) \in \{0, \frac{1}{n}, \frac{2}{n}, ..., 1\}$, while the latter is represented by the continuous value $S_{tissue}^t(\mathbf{p}) \in [0, 1]$. These states are altered

during the simulation by a set of predefined rules. Based on the states, three different cell classes can exist. Positions with $S_{tumor}^t(\mathbf{p}) = 1$ are considered to be part of the tumor. The remaining cells having $S_{tissue}^t \mathbf{p} > 0$ are part of the healthy tissue, while the rest is background, i.e., free space inside the uterine cavity.

The dynamic evolution of the system is controlled by local space- and time-independent transition rules R_i. The latter can be either deterministic or probabilistic. In order to model tumor growth, the conventional cellular automaton is extended by introducing a global rule R_{global}, which controls the application of the rules R_i. The former is time-dependent, thus allowing one to change the sequence of application of the rules R_i during tumor evolution. The individual rules are applied simultaneously on all cells at every time step. The interaction neighborhood $\mathcal{N}(\mathbf{p})$ defines the set of neighboring nodes which will be considered when one of the rules is applied to position $\mathbf{p}$. Typical neighborhoods used in 3D are the von Neumann neighborhood $\mathcal{N}_6$ and the Moore neighborhood $\mathcal{N}_{26}$.

Four specific rules are sufficient to model the growth process of a myoma. The first rule of the tumor growth automaton describes the growth of the cancerous tissue. The tumor state of a cell at location $\mathbf{p}$ is increased if one of the nodes $\mathbf{q}$ in its neighborhood has a non-zero tumor state.

$$R_{grow} : S_{tumor}^{t+1}(\mathbf{p}) = \min\left[1, S_{tumor}^t(\mathbf{p}) + \tfrac{1}{n}\right],$$
$$\text{if } \quad S_{tumor}^t(\mathbf{q}) > 0, \quad \mathbf{q} \in \mathcal{N}_{26}(\mathbf{p}) . \tag{2.16}$$

In order to introduce some randomness in this process, the rule is only applied with a certain probability, depending on the neighbor location. If $\mathbf{q}$ is in $\mathcal{N}_6$, then the probability is high, otherwise it is low.

The next rule models the migration of the tumor inside the healthy tissue. The tumor states of all cells are moved in one of the six major directions based on a global cost function determined by the tissue states.

$$R_{move} : S_{tumor}^{t+1}(\mathbf{p}) = S_{tumor}^t(\mathbf{p} - \mathbf{d})$$
$$\arg\min_{\mathbf{d}} C(\mathbf{d}), \quad \mathbf{d} = (\mathbf{q} - \mathbf{p}), \quad \mathbf{q} \in \mathcal{N}_6(\mathbf{p}) .$$

The global cost function C sums up the tissue states of the neighbors of all cells with $S_{tumor} > 0$ in one of the six major directions. Thus, there are accordingly six different associated costs.

The displacement of healthy tissue by the growing tumor is produced by the third rule. The neighbor of a current cell position with the largest tumor value moves its tissue to the latter.

(a)   (b)   (c)   (d)
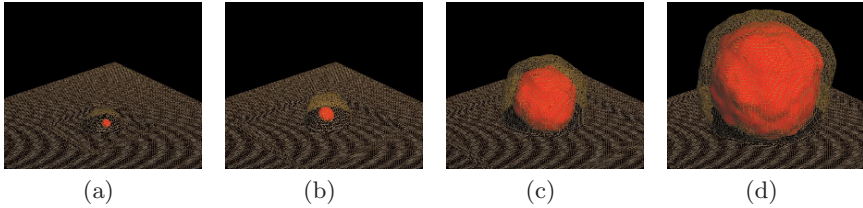
Figure 2.20. Steps of cellular automata-based myoma growth (outer tissue surface shown in beige wireframe, inner tumor surface in solid red). (a) Iteration t=20, (b) Iteration t=30, (c) Iteration t=40, (d) Iteration t=50

$$R_{displace} : \tilde{\mathbf{q}} = \underset{\mathbf{q} \in \mathcal{N}_{26}(\mathbf{p})}{\arg\max}\, S_{tumor}^{t}(\mathbf{q})$$
$$S_{tissue}^{t+1}(\mathbf{p}) = \min\left[1, S_{tissue}^{t}(\tilde{\mathbf{q}}) + S_{tissue}^{t}(\mathbf{p})\right] \tag{2.17}$$
$$S_{tissue}^{t+1}(\tilde{\mathbf{q}}) = 0\,.$$

After the displacement the tissue states are smoothed with a Gaussian filter. Thereafter, a final rule R_{close} ensures that the tumor cells are covered by a cell layer of healthy tissue.

For the growth simulation, the automaton is initialized by defining background and tissue cells. The states of the tissue cells are set proportional to the normalized distance from their position to the tissue/background interface (i.e., $0 < S_{tissue}^{0} < 1$). One of the tissue cells is then initialized with a non-zero tumor state. By placing this single tumor cell into the layer of healthy cells representing the myometrium the growth is started.

According to R_{global} different growth intervals are specified. At the start of the simulation a faster movement of the tumor is allowed. As its size increases the application of R_{move} can be reduced. The tumor volume can be approximately determined by multiplying R_{grow} with the respective probability of $\mathcal{N}_6$. This measure can also be used as a stopping criterion for the algorithm. An example of a growth process for a myoma based on the described cellular automaton on a 100^3 lattice is shown in Figure 2.20. Note that the depicted surfaces have been extracted from the volumetric lattice using the Marching Cubes algorithm.

The developed approach is able to simulate the growth of different kinds of myoma. A medical expert has to define the starting location of the tumor growth as well as the desired size. Thereafter, the generation process runs automatically. A resulting tumor geometry can be obtained in a few seconds on standard PC hardware. Nevertheless, a number of limitations exist. As mentioned, the outcome of the algorithm is a volume of labeled voxels describing the pathology. For visualization purposes, first a surface has to be extracted. Moreover, as in the previous approach, the generated geometry has to be merged with that of the healthy anatomy. One viable approach to this task would be to voxelize the part of the cavity where the tumor is going to

be located. After the growth process, a surface mesh can be obtained and merged with the remaining parts. However, due to the involved voxelization step some surface detail might be lost, depending on the underlying lattice resolution.

A further limitation of the cellular automaton is the problem of collision handling. During the evolution there are no explicitly defined tissue surfaces. Cells do not differentiate which part of the endometrium they originate from. Therefore, collisions of the endometrium covering the tumor with other parts of the uterine surface cannot easily be detected. A possible solution to this would be to keep track of the distance between tissue cells; however, this would considerably increase the computation time. Therefore, the algorithm is limited to generating spherically shaped myomas, which grow almost independently of the surrounding tissue. Finally, using a regular lattice is a limitation of the growth process. An extension to irregular grids (see, e.g., [199]) could possible improve the simulation outcome. This characteristic is an intrinsic part of the following growth model.

2.5.4 Particle System Growth Model

In order to avoid some of the drawbacks associated with a regular computational grid as well as alleviating the limitations with collisions, the modeling of tumor genesis with particle systems has been scrutinized. The latter approach has its roots in two initially unrelated areas – computer graphics and computational fluid dynamics. In computer graphics, particle systems were introduced in 1983 as a method to render entities and effects such as smoke or grass [223]. Numerous extensions were later suggested, for instance techniques for modeling of waterfalls [245] or flocking behavior [225]. In parallel, a similar method, Smoothed Particle Hydrodynamics (SPH), was introduced in the 1970s to model compressible fluid in astrophysical problems [103, 172]. This work was later extensively studied and extended in the field of computational fluid dynamics. More details can be found in [166]. In the following, the notion of particle systems will be introduced from the more heuristic point of view of computer graphics.

Particle systems consist of a set of ideal point masses $\mathcal{P} = \{\mathbf{p}_0, \ldots, \mathbf{p}_n\}$ located in space at positions (x_i, y_i, z_i) with associated attributes, such as mass, lifetime, or radius. No explicit mesh, topology, or lattice is given. The movement of the particles is governed by Newton's second law of motion $\mathbf{f} = m \cdot \mathbf{a}$, where the motion of the particles is described by six degrees of freedom, i.e., position and velocity.

Inspired by molecular dynamics, the interaction between particles is defined by a potential function Φ showing short-range repulsive and long-range attractive behavior. Forces between two particles $\mathbf{p}_i$ and $\mathbf{p}_j$ are proportional to the interparticle vector distance $\mathbf{d}_{ij}$ based on the gradient of the potential function

(a)   (b)
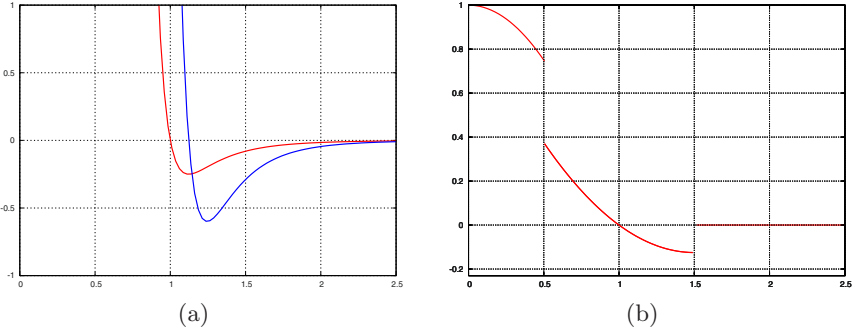
Figure 2.21. Potential functions and force profiles for particle systems. (a) Φ_{LJ} (red) and $-\nabla\Phi_{LJ}$ (blue) with $A = 1, B = 1$, (b) Λ_2 with $\mathbf{d}_0 = 1$

$$\mathbf{f}_{ij} = -\nabla\Phi(\|\mathbf{d}_{ij}\|) . \tag{2.18}$$

A number of typical potential profiles have been applied in the past, the most common one being the Lennard-Jones function [156], which models van der Waals attraction and Pauli repulsion between atoms.

$$\Phi_{LJ} = \frac{B}{\mathbf{d}^n} - \frac{A}{\mathbf{d}^m} , \tag{2.19}$$

with A, B, n, m being predefined constants with $n > m > 3$. $n = 12$ and $m = 6$ is the traditional form of the Lennard-Jones potential. In this equation the first term models the repulsion and the second the attraction. The interaction profile exhibits an equilibrium distance $\mathbf{d}_0$ at which the gradient is zero. The Lennard-Jones potential as well as its gradient is depicted in Figure 2.21(a).

The equations of motion describing the movement of the particles according to Newtonian physics have to be solved numerically. A straightforward approach to integrate the decoupled first-order equations is the explicit Euler method. Due to the explicit formulation, an appropriate time step δt has to be selected to ensure computational stability.

Since in the straightforward implementation all particles interact with each other, the solution of the system is of the order $O(n^2)$. However, some acceleration methods have been proposed in the past which allow for a much faster computation. For instance, with the method outlined in [107] a particle system with homogeneously distributed particles can be solved in $O(n)$. The algorithm is based on the introduction of pseudoparticles replacing groups of mass points. Spatial subdivision determines the location of the former. Moreover, a Taylor approximation of the potential is used, which further reduces the computation time.

Based on the introduced particle systems, the growth of polyps can be simulated. The ability of these systems to describe polyp-shaped surfaces has been shown in a different context in [260]. Since polyps are much softer than myomas, the surrounding tissue influences their growth. Therefore, collisions

between the pathology and the endometrium have to be included in the model. As in the cellular automata approach, particles in the growth model represent collections of cells instead of individual entities.

A critical element of the approach is the selection of an appropriate potential or force profile. The typical basic properties should be given, such as short-range repulsion and long-range attraction. To model cell interaction during polyp growth, a force profile function can be empirically selected, e.g., according to [60].

$$
\Lambda_2(\mathbf{d}) = \begin{cases} 0 & : & \mathbf{d} > \frac{3}{2}\mathbf{d}_0 \\ \frac{1}{2}(1 - \frac{\mathbf{d}}{\mathbf{d}_0})(2 - \frac{\mathbf{d}}{\mathbf{d}_0}) & : & \frac{1}{2}\mathbf{d}_0 \leq \mathbf{d} \leq \frac{3}{2}\mathbf{d}_0 \\ 1 - (\frac{\mathbf{d}}{\mathbf{d}_0})^2 & : & 0 \leq \mathbf{d} < \frac{1}{2}\mathbf{d}_0 \end{cases} \qquad (2.20)
$$

where $\mathbf{d}_0$ is the predefined interparticle equilibrium distance. The resulting profile is shown in Figure 2.21(b). Discontinuities exist at $\mathbf{d} = \frac{1}{2}\mathbf{d}_0$ and $\mathbf{d} = \frac{3}{2}\mathbf{d}_0$; however, this does not cause any instabilities during computation. The first represents rapidly increasing forces due to the incompressible behavior of tissue, while the second models tissue failure due to extensive loading. A key difference from the Lennard-Jones potential is the bounded maximum force at close ranges, which allows larger time steps during simulation. Nevertheless, the profile selection is quite heuristic and there might be functions which better approximate the behavior of particle interaction during tumor growth. However, as shown below, reasonable results can be obtained with the selected relationship.

The particle positions are updated based on a quasistatic solution of the motion equations using Euler integration. The reduced formulation results from setting the damping coefficient to $\gamma = \frac{2m}{h}$.

$$
\mathbf{p}^{t+\delta t} = \mathbf{p}^t + \frac{h^2}{m}\mathbf{f}^t . \qquad (2.21)
$$

The set of particles is divided into three different categories. The myometrium is represented by a constant number of fixed particles $\mathcal{P}_{myo}$, which are located at the base of the tumor. The polyp is modeled by a set of freely movable particles $\mathcal{P}_{tumor}$. Initially only one single particle is located within the healthy tissue, which subsequently continues to divide to model the tumor growth. Finally, all other tissue is covered by the endometrium, which is represented by particles $\mathcal{P}_{endo}$ with explicitly defined connectivity, forming a triangular surface. This extension of conventional particle systems is necessary to prevent particles from passing through the endometrial membrane. Instead, the growth process causes the particle surface to deform. To this end, the connections between the endometrial particles are modeled by linear springs, enabling surface adaptation by limited deformations. The iterative growth algorithm has four independent stages – tumor particle subdivision, tumor particle adaptation, endometrial surface adaptation, and surface fairing.
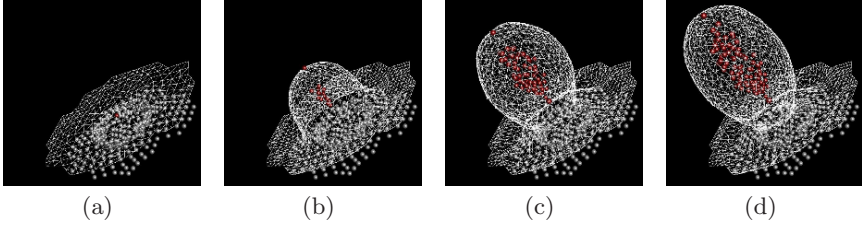
Figure 2.22. Steps of particle system-based polyp growth (endometrial surface shown in white wireframe, tumor particles as red, and myometrial particles as gray spheres). (a) Initial state with 1 tumor particle, (b) Growth equilibrium with 13 particles, (c) Growth equilibrium with 53 particles, (d) Growth equilibrium with 82 particles

The first step models the increased proliferation of tumor cells. For a polyp, the initial particle $\mathbf{p}_0 \in \mathcal{P}_{tumor}$ is divided at each time step, thus constantly increasing the number of tumor particles in the model.

$$\mathcal{P}_{tumor}^{t+\delta t} = \mathcal{P}_{tumor}^t \cup \{\mathbf{p}_{n+1}\} \qquad |\mathcal{P}_{tumor}^t| = n$$
$$\mathbf{p}_{n+1} = \mathbf{p}_0 - \lambda \mathbf{n}_{prox} \qquad \lambda \ll \|\mathbf{d}_0\| , \tag{2.22}$$

where $\mathbf{n}_{prox}$ is the normal of the surface patch closest to the dividing tumor particle $\mathbf{p}_0$. Due to the particle division strategy, the initial particle is always located at the tip of the growing polyp, while new particles are generated nearby. In order to model the spherical growth of a myoma, the previous strategy can be adapted, allowing each tumor particle to randomly divide.

Due to the proliferation of the tumor, it is occupying an increasing amount of space. This process is modeled by computing the interaction forces between the particles, according to which the latter are displaced in the second step.

$$\mathbf{f}_i = \sum_{\mathbf{p}_j \in \mathcal{P}_{tumor}, i \neq j} \Lambda_2(\mathbf{d}_{ij}) \frac{\mathbf{d}_{ij}}{\|\mathbf{d}_{ij}\|} . \tag{2.23}$$

In addition, it has to be ensured that all particles stay inside the endometrial surface. Therefore additional repulsive forces between particles of $\mathcal{P}_{tumor}$ and $\mathcal{P}_{endo}$ are introduced. A different force profile is used in this case, which creates only repulsive forces to mimic collisions with the surface .

$$\tilde{\mathbf{f}}_i = \sum_{\mathbf{p}_j \in \mathcal{P}_{endo}} \frac{1}{1 + \tilde{\mathbf{d}}_{ij}^4} \qquad \mathbf{p}_j \in \mathcal{P}_{tumor} . \tag{2.24}$$

Finally, the fixed particles at the base of the tumor $\mathcal{P}_{myo}$ are included in the computation to avoid tumor particles passing into the myometrium. Thereafter, the endometrial surface is updated according to the new particle positions in order to cover all tumor tissue. To this end, forces proportional to the distance between the tumor particles and their projections onto the
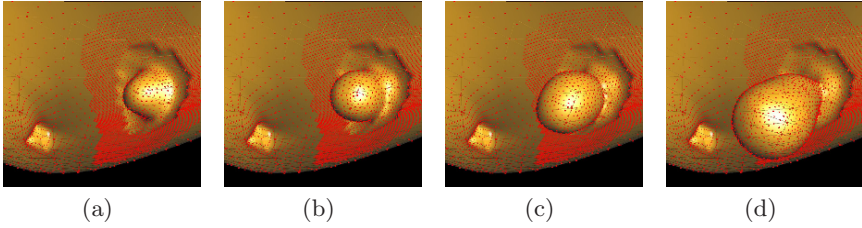
Figure 2.23. Particle system polyp growth in uterine cavity. (a) Iteration 8, (b) Iteration 25, (c) Iteration 80, (d) Iteration 140

nearest surface triangle are determined in the third step. These forces are distributed to the nodes of the respective closest triangle. Based on this, the static equilibrium of the mass-spring network of the surface mesh is determined.

The previous process is carried out without any constraints on surface deformation, thus causing irregularities to emerge. Therefore, an additional step for surface fairing has to be performed. This includes triangle subdivision as well as mesh improvement by edge swapping and/or edge collapse.

Triangles, which due to the surface deformation have grown larger than a predefined threshold, are subdivided using a quaternary scheme. This also requires the subdivision of neighboring triangles to avoid singular, nonmanifold vertices. Thereafter, edges between two triangles are flipped if this improves their quality metric Q. The latter is based on the ratio of the radius of the inscribed circle to that of the circumscribed circle.

$$Q = 2\frac{r_{in}}{r_{out}} = \frac{(b+c-a)(c+a-b)(a+b-c)}{abc} \,, \tag{2.25}$$

where a, b, c are the edge lengths of a triangle. An optimal (equilateral) triangle has $Q = 1$, while Q is approaching zero for almost degenerate cases. Usually, triangles with $Q > 0.5$ are considered to be of sufficient quality.

Finally, very small triangles generated during the remeshing process are removed by collapsing their shortest edge. This step is applied only when the triangle vertices as well as all their neighboring vertices have a valence greater than three.

After all these steps have been carried out, the mass-spring system is updated with new rest lengths corresponding to the current edge lengths, thus obtaining a new stationary equilibrium in the growth simulation. The described process stages are then iterated until the desired size of the polyp is achieved. Individual steps during the growth simulation are depicted in Figure 2.22.

By using the triangle mesh and vertices of the healthy anatomy to define the endometrial particles, the growth process can be carried out directly within the uterine mesh. Therefore, an explicit integration of the pathology is not required. Furthermore, collisions with the uterine surface can be handled

similarly to the interaction of the tumor particles with the endometrial surface. Repulsive forces are added based on the triangles of the cavity mesh. Figure 2.23 shows the development of a polyp inside the uterine cavity.

A significant drawback of this approach is simulation stability and the appropriate tuning of growth parameters. For instance, spring stiffnesses of the endometrial surface have to be heuristically adjusted to obtain the desired result. Setting them too low would result in a strongly distorted surface, while setting them too high would make the mesh too stiff, preventing a natural growth of the endometrium. Similar problems are encountered regarding the remaining system parameters. Unfortunately, the behavior of the system cannot easily be predicted, thus often leading to unsatisfactory results necessitating a restart of the growth simulation. In addition, the computation time is considerably longer as compared to the other methods. Depending on parameter selection, it can be on the order of hours, thus rendering the approach not interactive. This is a clear limitation, since a medical expert would need to see the result for the training scene generation without large delays. Nevertheless, improvements to the algorithm, for instance using Greengard's fast multipole acceleration or parallelization techniques, could help reduce the computation times to interactive rates.

2.6 Volumetric Representation

The techniques discussed so far generate triangular surface representations of the organ geometries. In the hysteroscopy simulator these will be used for the visualization of the scene. However, other modules of the simulation system require a volumetric representation of the objects based on tetrahedral meshes. This mainly includes the tissue deformation [265], the collision detection [118], and the tissue cutting subsystems [115]. The stability of these is highly dependent on element quality, therefore optimal 3D tetrahedral meshes need to be generated. During simulation the triangular surface is attached to the volumetric model and updated according to its deformation. This strategy allows a lower mesh resolution for the deformation model without compromising rendering quality.

Mesh generation is an active field of research extending into several application areas. A comprehensive survey of existing techniques can be found for instance in [202]. Methods focusing on tetrahedral mesh generation usually fall into one of three main categories – 3D Delaunay, Octree, or advancing front techniques.

The first class of algorithms is based on the Delaunay criterion [71], which is also sometimes referred to as the *empty sphere* property. It states that the circumcircle or -sphere of a triangle or tetrahedron, respectively, should not contain any other node of the mesh. An overview of Delaunay meshing techniques as well as of their limitations for the 2D case is provided in [238].

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

(e)   (f)   (g)   (h)
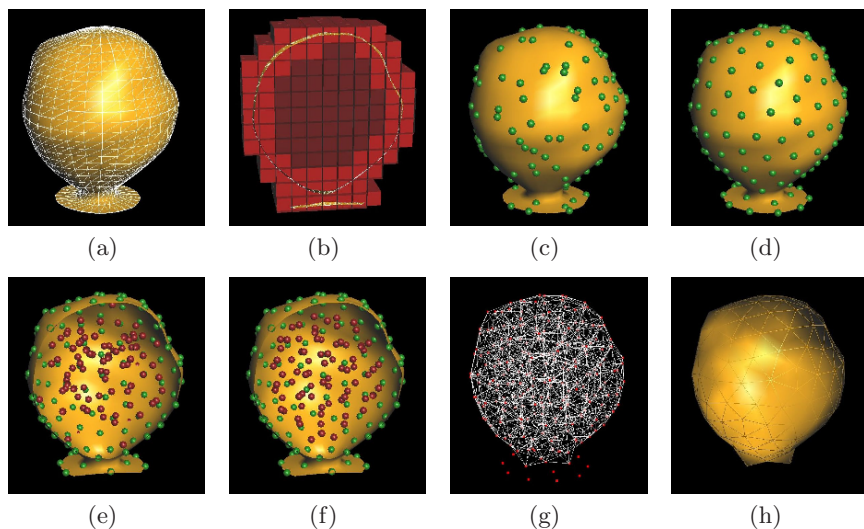
Figure 2.24. Tetrahedrization of myoma, (a) Triangular surface mesh of myoma, (b) Partitioning into cells ($\mathcal{C}_{surface}$ in light, $\mathcal{C}_{inner}$ in dark red), (c) Randomly initialized surface particles, (d) Surface particles after optimization, (e) Randomly initialized inner particles, (f) Inner particles after optimization, (g) Resulting tetrahedra in wireframe, (h) Surface of resulting tetrahedra

Based on the Delaunay criterion, several meshing techniques have been proposed, which differ mainly by the node selection or creation strategy (see e.g., [53, 52]). Automatic methods for 3D have for instance been discussed in [98, 289]; further related work can also be found in [74, 288]. However, Delaunay approaches usually require a special treatment of the boundary to maintain the existing geometry.

A different mesh generation strategy uses Quad- or Octree structures to obtain 2D or 3D meshes. This approach has initially been discussed in [302], and later on extended in [237]. The underlying idea is to progressively subdivide the space encompassing an object into cells until a desired resolution has been achieved. Thereafter, mesh elements are created based on the intersections of the surface with the cells. Unfortunately, the original boundary can be recovered only with limited precision. Moreover, problems can arise due to excessive element count.

Approaches in the third category employ an advancing front paradigm, progressively creating tetrahedra inward from the initial surface triangulation [168, 169]. The algorithm either selects existing nodes or creates new ones to form a tetrahedron. Moreover, intersection tests are required to detect collisions of different sections of the advancing front. An advantage of these methods is the retention of the original surface mesh.

(a)   (b)   (c)
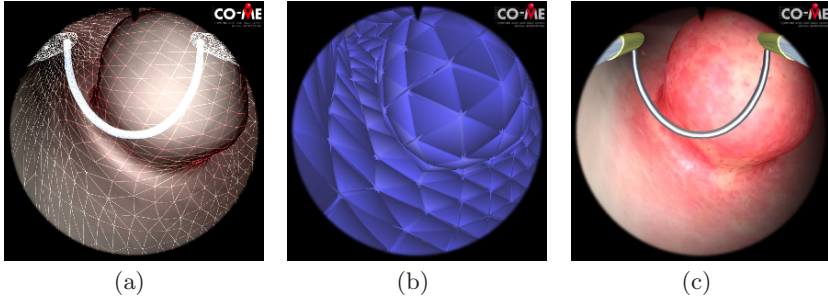
Figure 2.25. Mesh representations of training scene with myoma (with kind permission of Springer Science+Business Media, LLC). (a) Surface mesh of pathology integrated into cavity, (b) Tetrahedral representation of objects, (c) Final fully textured training scene

For generating the volumetric representations of the scene geometries, a hybrid technique can be followed. In a first step a regular triangulation of the existing surface mesh is obtained. This is necessary since the pathology generation does not ensure reasonable mesh quality. On the contrary, highly irregular meshes in terms of element size and connectivity result during scene generation. Moreover, for the tetrahedral mesh a lower resolution is required than for the surface representation.

As with Octree techniques, the space is initially subdivided into a set of regular hexahedral cells $\mathcal{C}$ at a predefined resolution. This set is then subdivided into inner $\mathcal{C}_{inner}$, outer $\mathcal{C}_{outer}$, and surface cells $\mathcal{C}_{surface}$. The latter are obtained in a two-stage process. First, cells enclosed by the bounding box of a mesh triangle are determined and linked to that specific triangle. Thereafter, this rough approximation is refined by keeping only those cells that actually intersect or touch their associated triangle. After obtaining the set of surface cells, the outer cells are determined by region growing, using the $\mathcal{N}_{26}$ neighborhood. This process is limited by the bounding box of the surface mesh. Finally, the remaining cells are considered to be inside the object $\mathcal{C}_{inner} = \mathcal{C} - \mathcal{C}_{surface} - \mathcal{C}_{outer}$.

After this, a preselected number of particles is initialized in each cell of $\mathcal{C}_{surface}$ at random positions and then projected onto the triangular surface. Thereafter, their locations are regularized based on a particle system, with the movement constrained to the surface mesh. A similar step is then carried out for the inner cells $\mathcal{C}_{inner}$, with the only difference being that particle movement is unconstrained. To improve convergence, the regularization is amended with a temperature term similar to that of simulated annealing. It reduces particle mobility in every iteration, thus causing the particle system to reach equilibrium in about 30 iterations. In the final step, all particles are connected into a tetrahedral mesh with the Delaunay approach. The individual steps of this process for a myoma surface mesh are depicted in Figure 2.24.

(a)   (b)

(c)   (d)
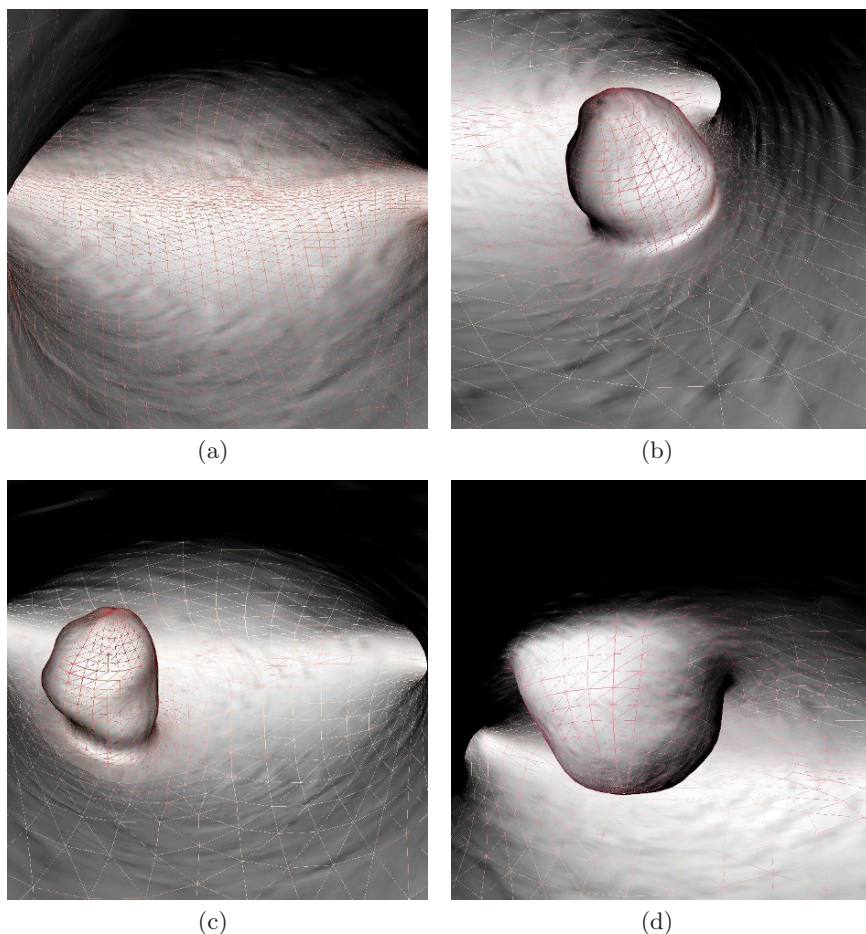
Figure 2.26. Surface meshes of different hysteroscopy scenes

Due to the selected approach the higher resolution surface mesh can only roughly be approximated by the tetrahedra. In Figure 2.25 the two object representations as well as the final rendering of a training scene are shown.

2.7 Modeling Examples

The proposed methods have been used to generate the geometrical representations of training scenes for hysteroscopy simulation. Figure 2.26 shows a few exemplary surface meshes. An open question remains the validity of the generated structures. While the model of healthy anatomy could be shown to be valid in terms of generality, compactness, and validity; a similar proof for

the pathology generation is not viable. A major problem is the limited resolution of imaging technology, which is usually not sufficient to clearly outline tumors or myomas in the uterine cavity. As an alternative, images acquired during hysteroscopy could be analyzed to determine typical shapes and sizes of tumors. Unfortunately, the actual growth process is not observable since detected tumors will usually be removed. Therefore, qualitative assessment of generated pathologies by medical experts seems to be the only viable approach at the current stage.

Another question is the appropriateness of these tools for training scene definition by a surgeon. The discussed pathology generation schemes exhibit a clear trade-off between interactivity and biophysiological validity. In terms of imitating real biological processes, the particle-based growth is currently the most accurate approach. However, due to its computation time and parameter setting problems, it currently is not appropriate for training scene generation. In contrast to this, the skeleton-based design does not incorporate any biological knowledge; however, it allows the highest degree of user control. It is still an open question which of the presented methods serves the actual goal of surgical simulation best.

3

Appearance

3.1 Introduction

Modelling of organ appearance in the virtual scene is the next element of the scenario generation process. As stated in [116], realism of rendered scenes demands complexity, or at least the appearance of complexity. Therefore, we strive to enhance the visual richness of the surface triangle models generated in the previous step by adding further detail. The application of surface textures to geometric models – also referred to as *texturing* – is currently one of the key techniques in interactive computer graphics, to add complexity and thus realism to visual rendering. With this approach, the representation of minute object detail with geometric information can be avoided. Moreover, texture mapping requires only little additional computational effort in the rendering process.

3.1.1 Definitions

The term *texture* does not have a specific general or mathematical definition. Depending on the context, it can have different meanings. For instance, from the point of view of the sense of touch, it denotes the tactile properties of the external surface of an object (e.g., roughness, stickiness). For the gustatory sense, the texture of food relates to specific properties, for instance, crispness or crunchiness. In contrast to this, in material sciences, the term characterizes material properties resulting from the orientation of crystallites.

In the visual domain, one usually thinks of texture as an area which contains a uniform spatial characteristic, which is often attributed to a specific material of the object or surface. In a general sense, the texture appearance is due to the varying light reflectance characteristics of the surface structure. The appearance can be due to different physical effects, ranging from variation of light absorption and reflection (denoted as *flat* textures) caused by submicroscopic structures, via macroscopic effects such as self-shadowing and

| (a) | (b) | (c) |
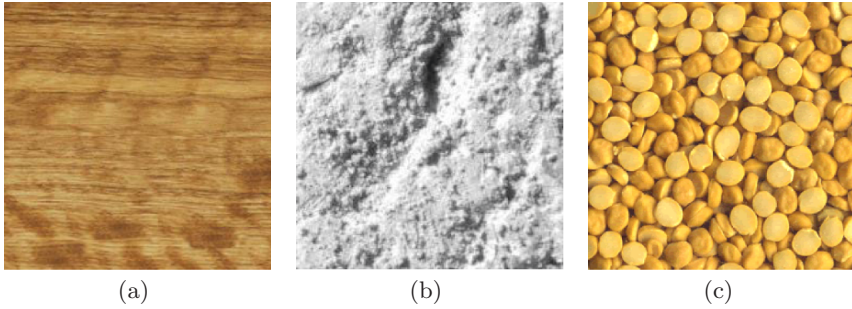
Figure 3.1. Different classes of textures. (a) Submircoscopic variation causing spatially varying color perception, (b) Surface relief on macroscopic scale, (c) Collection of objects from distance

-occlusion, to mere collections of objects seen from a distance (Figure 3.1). Texture patterns are usually a combination of all of these classes, and a clear separation is not possible. It should be noted that the appearance of textures is strongly influenced by viewing direction and illumination conditions. These factors are also critical in endoscopic imaging – our source of sample textures – and therefore have to be taken into account.

In addition, several perceptual properties are often attributed to textures, which shall be briefly addressed in the following.

Regularity. Textures can be subdivided into *deterministic* and *stochastic* variants. The former are usually built by repetition of basic patterns in various directions – a typical example being a brick wall. The latter obey a nontrivial probability distribution, and show no dominant frequency or pattern – a typical example being clouds or pasture. Most natural textures fall into the second category.

Isotropy. While isotropic textures are completely independent of direction, and can for instance be freely rotated, anisotropic ones show dominant directions in their appearance – an example of the latter being wood grain.

Resolution. In perceptual terms, texture resolution is related to homogeneity and scale. Usually, textures are invariant under translation, and to a certain degree also to scale. The smallest patch of a texture which exhibits the same perceptual characteristics can be regarded as the texture resolution.

Composition. Textures are often built up in a hierarchical manner containing smaller subtextures. This allows us to treat the different levels of the texture separately. An example of this could be the characteristic substructures of organ surfaces, such as specks on livers.

Many contributions to understanding texture perception in the human visual system, especially discrimination issues, have been made by Bela Julesz

(a)   (b)   (c)   (d)
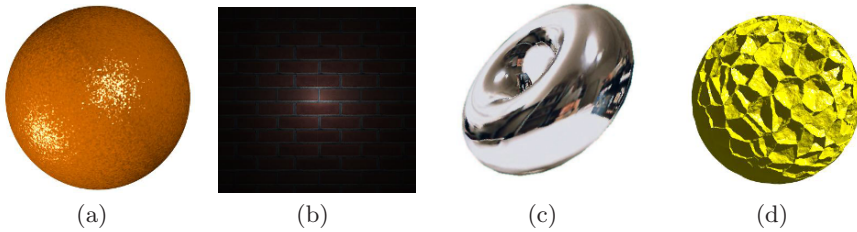
Figure 3.2. Application of texture mapping in computer graphics. (a) Bump mapping, (b) Light map, (c) Environment mapping, (d) Cellular texture

(e.g., [132]). He conjectured that textures with the same power spectrum or with identical second-order correlations cannot easily be discriminated. This topic will be discussed in more detail in Section 3.6.1.

Finally, it should be noted that semantics are often attributed to a specific texture, depending on the viewing context or preknowledge of the beholder. This is for instance true for medical experts, who gain information from an organ's appearance which might not be obvious to laymen.

3.1.2 Texturing in Computer Graphics

The discussion so far has examined textures from a general point of view. However, the application of textures in computer graphics is the main focus of our endeavors, therefore, this area is reviewed with more detail.

The notion of texture mapping was initially introduced in 1974 by Edwin Catmull as a new technique to perform high quality image synthesis [49]. While the major application of texturing is the representation of surface color, several other visual effects based on textures have been developed. In the following some typical techniques are outlined.

Surface color. The standard texturing approach uses one-, two-, or three-dimensional patterns, which are mapped to the surface of an object to determine its color. Initially introduced in [49], this notion is nowadays the most common application of textures. It will also be the major focus of the investigations related to surgical scene generation. Further generalization of the method by including transparency, as described in [94], allows the rendering of a number of natural phenomena, for instance, clouds or smoke.

Illumination enhancement. Real-time rendering algorithms require surface normals to compute light reflection. By using textures to define local perturbations of the normals, an otherwise flat surface appears to have bumps and wrinkles (Fig. 3.2(a)). This method, known as *bump mapping*, was introduced in [28]. A different effect related to lighting can be

achieved with *light maps.* Precomputed illumination data are used to locally modulate light intensity (Fig. 3.2(b)). The increased use of per pixel shading, however, makes this approach more and more obsolete. A different texture-based illumination technique was introduced to represent highly reflective surfaces, which mirror the object surroundings. In order to approximate results obtained from time consuming ray tracing, *environment mapping* can be applied (see e.g., [106]). In this method, a texture built up from images of the surrounding environment is indexed by the directions of rays reflected on the object surface (Fig. 3.2(c)).

Geometry adaptation. An alternative application of texture maps is to define the actual geometric variation of object surfaces. In contrast to bump mapping, this technique creates real bumps in the model. Therefore, high mesh resolutions are required, and interactivity is only possible in limited cases. *Hypertexture* or *cellular texture* methods fall into this category. The former was presented in [207], where object shape was defined as the procedural modification of density in three-dimensional space. The latter was introduced in [84], and denotes the generation of geometric patterns by biologically motivated cellular development. With this technique, surface elements such as scales, feathers, or thorns can be represented (Fig. 3.2(d)).

3.1.3 Relevance to Surgical Education

Apart from the underlying technical aspects, the relevance of appropriate texturing of surgical scenes should also be examined from the medical point of view. Especially in the limited views of minimally-invasive settings, where the surgical site is accessed through natural orifices or small incisions in the skin, texture contains several cues facilitating spatial navigation. As discussed in [32, 196], changes in texture characteristics according to perspective distortion provide knowledge about depth and orientation. Since endoscopic interventions are usually performed with monoscopic cameras, these monocular depth cues contain significant additional semantic information. Apart from this, different kinds of texture of various organs provide further cues for orientation. This is especially true for laparoscopic procedures in the human abdomen, where a large variety of structures is present. Typical organ surface textures allow a surgeon to infer his position from the limited endoscopic view. For instance, fat, intestines, or liver all exhibit characteristic surface patterns and color. Moreover, the variation of individual organ textures hints at possible pathological changes. The distinct appearance of the uterine endometrium could for instance provide knowledge about the presence of endometritis or hyperplasia. Thus, in an advanced education system, the organ textures should also reflect the pathologies present in the training scenario. In terms of the level of realism, appropriately varying textures represent a significant element of a surgical scene. They affect the training of low-level skills, e.g., orientation in the surgical site; as well as high-level skills, e.g., decision making according to semantic information.
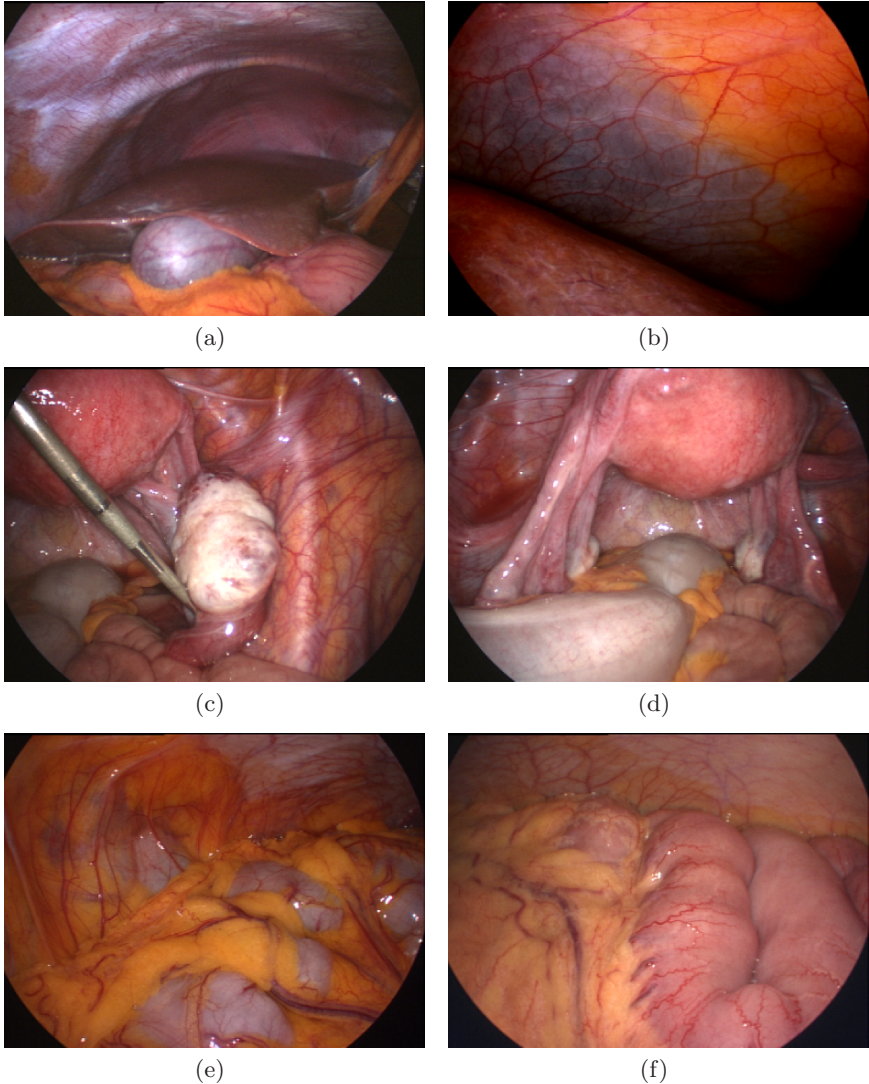
Figure 3.3. Pictures taken during laparoscopic interventions. (a) Cranial section of abdomen showing liver, gallbladder, peritoneum, and diaphragm, (b) Close-up of diaphragm with typical vascular structures, (c) Kaudal section of abdomen depicting uterus, ovary, and peritoneum, (d) Uterus structure, including corpus, fallopian tubes, and other adnexa, (e) Patches of fat covering intestinal structures, (f) Small intestine with fat and typical vessel trees

As discussed earlier, the methods presented in the following sections of this chapter were developed for two settings – gynecological laparoscopy and hysteroscopy. Because of different access methods to the uterus, varying

(a)   (b)

(c)   (d)
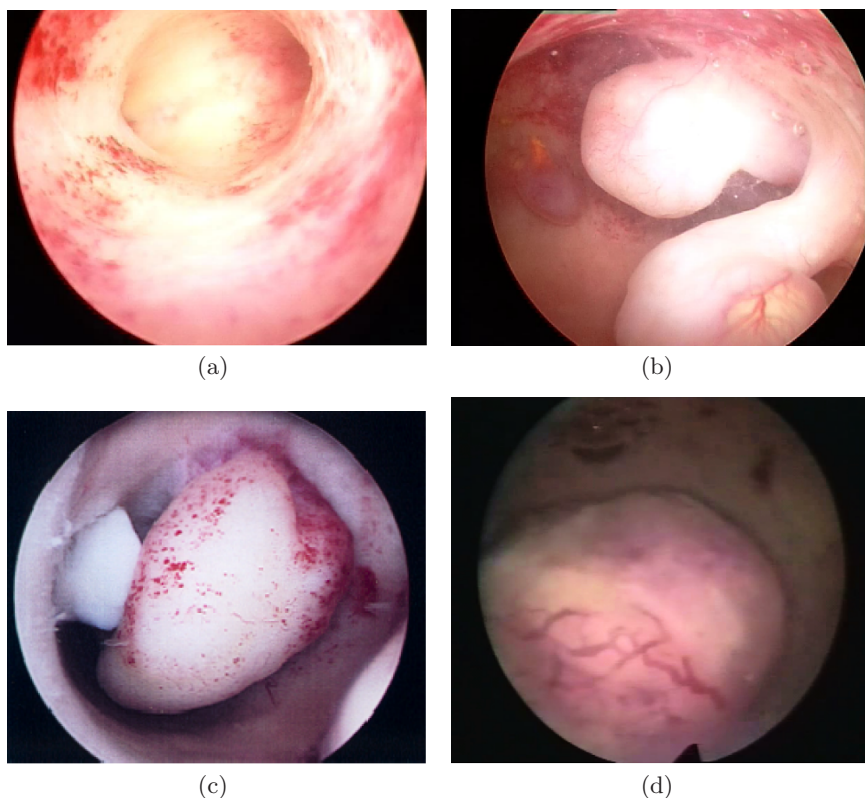
Figure 3.4. Pictures taken during hysteroscopies. (a) View inside the fluid-filled uterine cavity with no abnormal findings, (b) Small polyps growing from the endometrial wall, (c) Larger polyp inside uterine cavity, (d) Uterine leiomyoma with typical vessel structures

requirementsfor organ texturing have to be met. In Figures 3.3 and 3.4, snapshots of real interventions show characteristic views of both approaches. Typical surface textures can be seen in both collections. It should be noted that the imaging conditions vary greatly for the two methods. While in the gas-insuflated abdomen high quality pictures of all structures can easily be taken, the intrauterine image quality is usually quite low. The depicted selection represents samples taken under nearly optimal conditions.

3.1.4 Process Elements

In the next sections, all the steps necessary for generation of organ textures will be discussed in detail. The first element is the intraoperative image acquisition to form a ground truth, which is necessary for most of the texture synthesis methods. Then image enhancements are carried out to deal with
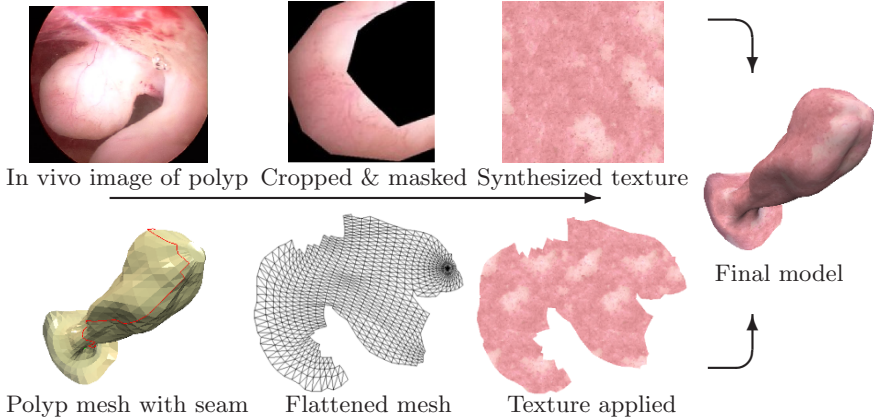
Figure 3.5. Texturing process for hysteroscopy simulation (from [203])

the often suboptimal imaging conditions. Subsequently, the image data can be used for textures synthesis. Thereafter, additional procedural texturing methods can be applied to include further detail in the textures. The final step comprises the mapping of the texture to the geometric organ surface.

It should be noted that tissue is often covered by vascular structures. These vessel patterns contain vital information for the surgeon, and need to be reproduced appropriately. However, the structures are usually too complex to be synthesized with texture generation approaches. Moreover, vessels also have to be included in the simulation as geometric objects, since they will be sources of bleeding when cut. Therefore, vascular trees have to be treated separately from the underlying *base texture*. The discussion in this chapter is constrained to the generation of varying, organ-specific base textures. The integration of vessel systems will be briefly reviewed in Chapter 5. Nevertheless, it should be noted that some surface structures exist – for instance, specks on livers, or follicles on ovaries – which will be addressed in this chapter.

In order to provide an overview of all necessary steps, the texturing approach taken for the hysteroscopy setting is depicted in Figure 3.5. The method uses tileable, variable textures created from in vivo images for the texture synthesis. Textures are mapped to 3D mesh geometry by mesh parameterization, taking into account the visibility of seams and distortion reduction. More details of these process elements, as well as those for laparoscopic textures, are explained after a review of previous work.

3.2 Previous Approaches in Surgery Simulation

This section focuses on a survey of the methods used for generation of textures in the context of surgical simulation or for other related natural phenomena.

(a)   (b)   (c)   (d)
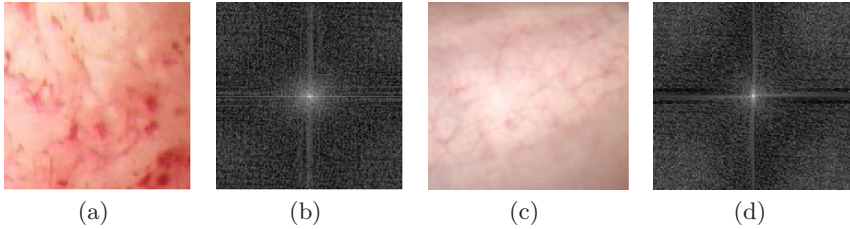
Figure 3.6. Fourier spectra of different organ textures. (a) Endometrium texture t_1, (b) Magnitude of Fourier transform $log(|T_1|)$, (c) Bowel texture t_2, (d) Magnitude of Fourier transform $log(|T_2|)$

More general work in texture generation and mapping will be addressed below. More precisely, procedural texturing will be detailed in Section 3.5, human texture perception in Section 3.6.1, and a review of pixel- and patch-based texture synthesis will be provided in Section 3.6.2. Nevertheless, here previous attempts related to surgical simulation are examined, in order to put the presented texturing methods into an application-oriented context.

The most basic approach for creating organ textures suggested in the past is direct texture painting. This usually requires a medical illustrator to manually draw the texture with the help of appropriate tools. In [233] a two-dimensional painting method was used to obtain textures for a virtual environment targeting biomechanical analysis and education. The illustrator has to ensure in a trial and error process that the 2D drawings map to the 3D geometry without distortion. More advanced approaches proposed to draw textures directly in 3D. Initial steps in this direction were reported in [113]; however, the lack of a true 3D user interface makes difficult an intuitive usage of the system. Improved tools, for instance using haptic interfaces, were presented later, e.g., in [2, 130, 20]. Nevertheless, the direct painting of textures does not meet the requirements of variable training scene generation, since a new texture would have to be created or adapted for every new organ. While high texture detail might be achievable with such an approach, its costly and time consuming nature prohibit its usage in our setting.

An interesting alternative to direct painting in the spatial domain is presented in [160], where the author suggests drawing a texture in the frequency domain. It is possible to create patterns found in wood or canvas with this method. Nevertheless, it is admitted that the correct spectrum for natural textures might be as complicated as the texture itself. In Figure 3.6, logarithmic transformations of the Fourier spectra of two representative samples are depicted. Unfortunately, no patterns can be identified which would encourage closer examination of texture drawing in the frequency domain.

In [154] an approach using polyhedron decomposition is described which allows the treatment of each surface triangle as an independent entity with its color information stored in a unique texture space. The triangles are

arranged in the parameter space to optimize area coverage. Textures representing anatomical details are manually placed by medical experts with a 3D tool, and interactively mapped to the surface. Blending between different texture patches reduces discontinuities. The method has been applied in a simulator for minimally invasive neurosurgery. Real images acquired during endoscopy are used to manually generate the texture patches. Again, these methods require too much manual input from the user.

A different strategy using texture samples and avoiding distortion, discontinuity, and repetitiveness is presented in [194]. It is based on the triangular tiling of a surface at a user-specified scale. The tiling defines a local parameterization, which is used to map triangular texture patches to the surface triangles. If the tiling is relatively homogeneous, distortions are expected to remain favorable. Due to the selected tiling, texture patches have to match their color and local derivative at the borders. The patches are either created manually or semiautomatically. In the former case, the patches are hand drawn samples, or manually edited real images. In the latter case, the textures are synthesized based on nonperiodic cellular patterns [297] or fractal noise [208]. This method was used to apply texture samples to a 3D mesh of a liver. The patches were synthesized with Worley's approach, and four sets of equilateral triangles with isotropic homogeneous texture were applied. This work was later extended in [193] by adding dynamic effects directly to the surface texture, for instance, cauterization marks or blood drops. These effects were added with direct painting paradigms, and distortions were avoided by considering the local Jacobians. The texture variation achieved with these techniques is rather limited, due to the small number of triangular patches.

A different approach for obtaining textures for medical scenes is the mapping of real volumetric data to surfaces. In [224], the Visible Human data set is utilized to obtain texture for organs. The authors segment individual structures from the data. Thereafter, polygon models are placed along the segmented boundaries and textured directly from the volumetric data. New textured surfaces, created for instance during cutting procedures, can also be obtained on the fly. Unfortunately, the segmentation of the anatomical structures in the dataset is time-intensive and tedious, and can be carried out only with limited precision. Moreover, variable scenes cannot be generated with this approach. Similarly, volumetric texture mapping is also proposed in [162]. A tetrahedral mesh is mapped to the Visible Human data, which can be cut and deformed. An interesting alternative has been suggested in [144]; namely to generate organ-specific CT color lookup tables based on the Visible Human CT and cryosectional data. By the mapping of Hounsfield units to RGB values, different CT datasets can be colorized. Nevertheless, this method provides only an approximate colorization of organs, and does not include detailed surface information. Moreover, the appearance of specific pathologies cannot be obtained with this approach.

Another recent trend in surgical simulation is the use of methods inspired by image-based rendering [136] for realistic visualization. For instance, in [82],

real images are combined with artificial specular reflections, modulated by a set of reflectance maps, to simulate different views by the endoscopic camera. The maps are generated with Perlin fractal noise. As an application example, this method was used in bronchioscopy simulation. The model geometries are obtained from CT images, while the textures are acquired from live endoscopic images, and later registered to the model. Unfortunately, the authors do not discuss the texturing process in more detail. It is not clear how correct correspondences between texture and geometry are obtained. Moreover, it is not explained how problems like missing regions or highlights visible in the actual endoscopic images are handled. Finally, the approach would have to be repeated for every new model. A similar approach is reported in [218], where specular highlights are created with environment mapping, utilizing current graphics hardware extensions. However, the authors do not report on how the cube map of the texture is generated in order to avoid distortions. Recently, a combination of conventional texture mapping, image mosaicking, view-dependent texture mapping, and light source shading has been proposed in [129] for realistic rendering of surgical scenes. The authors assume that the background in endoscopic simulation does not change significantly, thus enabling the rendering of novel views with image-based rendering principles. First, mosaicking is applied to composite background images from an endoscopic video sequence. Since different viewing angles cause varying specular highlights, a number of these background images are created for different angles. During runtime, view-dependent blending of these images is performed. However, the scene foreground, i.e., the actual interactive liver model, is rendered with standard methods. Therefore, a large difference in rendering quality between foreground and background can be noticed in the scene.

For completeness, reaction-diffusion processes, which have been suggested for the synthesis of natural textures [295, 270], should also be mentioned. With this approach, chemical processes describing pattern formation in biological morphogenesis are simulated. As an example, typical animal fur patterns can be reproduced with this method. However, there is no straightforward extension of reaction-diffusion texture generation to the formation of specific organ patterns.
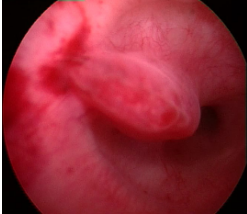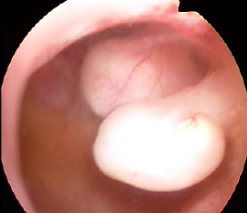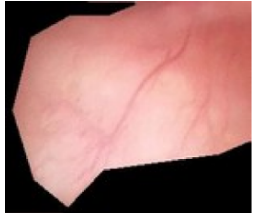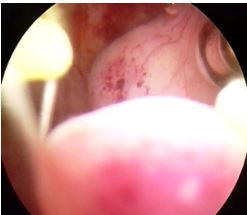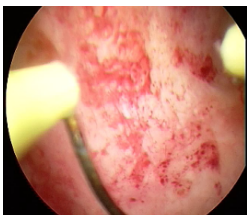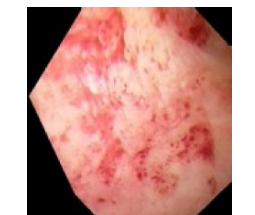
3.3 Data Acquisition and Enhancement

3.3.1 In Vivo Image Acquisition

The first step in the texture generation chain is the acquisition of image data from real interventions. This is necessary, since for the majority of organs an analysis/synthesis paradigm is followed. Only in limited simple cases can a texture be defined with a purely functional approach. Therefore, a ground truth covering the visual appearance of organs in healthy as well as pathologic conditions has to be obtained. Due to the varying boundary conditions

of laparoscopic and hysteroscopic interventions, different image acquisition techniques have to be used.

Table 3.1. Excerpt of in vivo image database

Original image	Cropped and masked image	Properties
		• ID: 105 • Object: polyp • Age: postmenopausal • Resolution: medium • Size: 374x312 • Focus: excellent
		• ID: 142 • Object: polyp • Age: unknown • Resolution: medium • Size: 198x140 • Focus: good
		• ID: 149 • Object: polyp • Age: unknown • Resolution: medium • Size: 253x111 • Focus: good
		• ID: 151 • Object: endometrium • Age: unknown • Resolution: high • Size: 329x292 • Focus: good

The largest limitations are encountered in hysteroscopy, mainly due to time constraints and poor visibility inside the uterus. The former are due to the problem of fluid overload caused by absorption or intravasation. Operation

time and the consumed quantity of the distension media have to be closely monitored. Generally, an operation time of 30 mins should not be exceeded. Due to this, only a limited amount of time can be spent obtaining high quality images of structures inside the uterus. Moreover, the view usually is obscured by floating tissue fragments, blood, and endometrial fibers. Therefore, instead of acquiring specific high-quality images, pictures were obtained from 10 hours of in vivo video taken during various hysteroscopies. Images representing different structures in the uterine cavity were selected, labeled, and input into a texture database. Usually, high resolution views of characteristic tissues were chosen, which optimally were perpendicular to the surface and without strong camera spotlight effects. 69 different samples were obtained that contained usable textures of various types of tissue. In Table 3.1 a few categorized textures of the searchable database are visualized.

In contrast to this, the acquisition process during laparoscopy is less complex. Since time constraints are less severe, and individual organs can be selectively brought into view, higher quality samples can be taken. In order to minimize the disturbance of the normal surgical procedure, snapshots can be taken by pressing a foot pedal. With this approach, about 400 high resolution true color images were obtained from 16 different laparoscopies.

3.3.2 Image Enhancement

Before the acquired images can be used for texture synthesis, a few additional preprocessing steps have to be carried out. The first step is to crop from the image the region which contains the appropriate sample for a specific organ texture. It should be noted that the quality of the image can be diminished by a number of effects. Texture samples can show a trend due to orientation, scale, or viewing angle. This is especially the case for highly curved organ surfaces. In order to reduce this trend, images should be taken from flat areas of an organ with a perpendicular viewing direction. However, perspective distortion present in endoscopic cameras cannot be avoided. A previous calibration of the optical system before image acquisition might reduce these effects; however, such a process has to be performed before every operation, and cannot easily be integrated into the normal workflow. Moreover, image quality might be further deteriorated due to blur – a particular obstacle in hysteroscopy. Finally, problems can be encountered due to the interlaced imaging mode, in which half images are acquired alternatingly for even and odd lines. This sometimes adds high frequency noise to the data.

The cropped samples obtained in the first step should contain relatively homogeneous regions of the desired base texture. However, additional structures might be present in the sample, which have to be removed before further analysis. This includes highlights coming from the camera system. Since a perpendicular viewing direction is sought, these interfering highlights are usually part of the image. In laparoscopy, specular reflections are present; while in hysteroscopy, diffuse intensity shifts are encountered. Apart from this, other

(a)   (b)   (c)   (d)
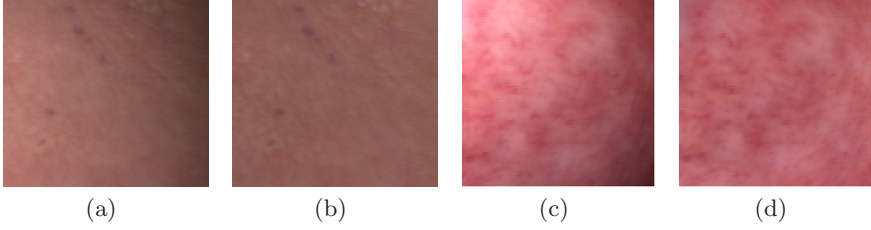
Figure 3.7. Reduction of intensity drift for texture samples. (a) Texture sample of liver surface, (b) Drift reduction with quadratic fit, (c) Texture sample of uterus surface, (d) Drift reduction with quadratic fit

surface structures which should not be part of the base texture also have to be removed, e.g., vessels, or structures like follicles on ovaries. Therefore, regions containing these types of structures are masked in the images. It should be noted that this results in arbitrarily shaped texture samples with masked pixels which have to be taken into account in the later processing steps.

Illumination trends due to drifts in intensity and saturation caused by unhomogeneous distribution of light energy and diffuse reflection on curved organ surfaces are another difficulty. While these effects might not be directly noticeable by a human observer, they have to be rectified before the synthesis process. One approach to alleviate this situation is to perform a filtering in HSV space based on least squares fitting. To this end, the RGB color image is converted to HSV space.

$$
H = \begin{cases}
60 \cdot \dfrac{G - B}{Max - Min}, & \text{if} \quad Max = R \\
60 \cdot \dfrac{B - R}{Max - Min} + 120, & \text{if} \quad Max = G \\
60 \cdot \dfrac{R - G}{Max - Min} + 240, & \text{if} \quad Max = B
\end{cases} \tag{3.1}
$$

$$
S = 1 - \frac{Min}{Max}
$$

$$
V = Max \,,
$$

where Max is the maximum of the color channel triplet $(R, G, B) \in [0.0, 1.0]^3$, and Min the minimum of these values (with $Max - Min \neq 0$ and $Max \neq 0$). In HSV space, drift will be corrected by separately processing the Hue (H) and Value (V) channel. A least squares approximation is performed with a quadratic or cubic fit function.

$$
f(x, y) = \sum_{n=0}^{k} \sum_{m=0}^{k} c_{n,m} \, x^n y^m, \quad \text{with} \quad k \in \{2, 3\} \,. \tag{3.2}
$$

In order to determine the unknown coefficients $c_{n,m}$, the least squares solution to an overdetermined system of linear equations has to be obtained. In the case of a quadratic fit, the equation system takes the form

$$
\begin{bmatrix}
1 & x_0 & y_0 & x_0^2 & x_0\,y_0 & y_0^2 \\
1 & x_1 & y_1 & x_1^2 & x_1\,y_1 & y_1^2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 & x_i & y_i & x_i^2 & x_i\,y_i & y_i^2
\end{bmatrix}
\cdot
\begin{pmatrix}
c_{0,0} \\ c_{1,0} \\ c_{0,1} \\ c_{2,0} \\ c_{1,1} \\ c_{0,2}
\end{pmatrix}
=
\begin{pmatrix}
v_0 \\ v_1 \\ \vdots \\ v_i
\end{pmatrix},
\tag{3.3}
$$

where v_i are the respective H or V values at positions (x_i, y_i) of the i unmasked texels in the sample texture. Equation 3.3 can be solved with standard solvers, e.g., Householder reflections or Givens rotations. After the coefficients of the fit function have been determined, the drift of the initial sample can be adjusted by subtracting the quadratic or cubic fit, and subsequently adding the mean of the sample texels. Finally, the new values are converted back to RGB color space (examples of the overall process are shown in Figure 3.7).

$$
H_i = \lfloor \tfrac{H}{60} \rfloor \bmod 6, \; f = \tfrac{H}{60} - H_i,
\tag{3.4}
$$

$$
p = V \cdot (1 - S), \; q = V \cdot (1 - (S \cdot f)), \; t = V \cdot \left(1 - \left(S \cdot (1 - f)\right)\right).
$$

$$
\begin{array}{llll}
R = V, & G = t, & B = p, & \text{if} \quad H_i = 0 \\
R = q, & G = V, & B = p, & \text{if} \quad H_i = 1 \\
R = p, & G = V, & B = t, & \text{if} \quad H_i = 2 \\
R = p, & G = q, & B = V, & \text{if} \quad H_i = 3 \\
R = t, & G = p, & B = V, & \text{if} \quad H_i = 4 \\
R = V, & G = p, & B = q & \text{if} \quad H_i = 5
\end{array}
$$

The final step in the image preprocessing is the reduction of colors by quantization. The acquired true color images can in principle contain more than 16 million different color values. As we will see later, computation time is often directly related to the number of colors. Therefore, it is advisable to limit the amount of the latter. The sample images are well suited for this step, since only a limited number of tones and intensities is present. Moreover, errors created during the process are obscured by the stochastic nature of the samples. Instead of uniform quantization of the color channels, more sophisticated algorithms such as the ones discussed in [99] or [298] should be used. Typically, 32 colors are sufficient for the majority of the samples. Figure 3.8 shows examples of the color reduction.

(a)   (b)   (c)   (d)
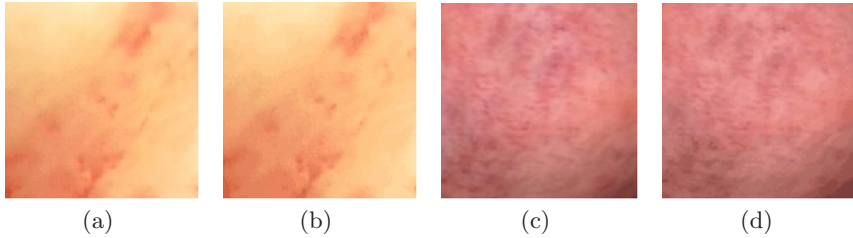
Figure 3.8. Color reduction by quantization. (a) Original texture sample of endometrial wall, (b) Quantized sample with 32 colors, (c) Original texture sample of uterus, (d) Quantized sample with 32 colors

3.4 Base Texture Generation

The next step in the texturing chain is the generation of the base textures. Generally, two different strategies could be followed for this: empirical procedural texture generation based on closed mathematical formulations, or analytical approaches applying example-based synthesis. Mainly the latter are used in the scenario definition process, since procedural methods suffer from several limitations. The main drawback is the nonintuitive control of the algorithms. First, an appropriate mathematical representation has to be found that describes a specific pattern. Moreover, a number of parameters have to be manually supplied to control the results. Nevertheless, a few simpler textures can be generated with the former strategy. Therefore we first discuss these methods, before moving on to the other category.

   Furthermore, it should be noted that the dimensionality of the generation has to be considered. Directly obtaining 3D textures avoids several texturing problems encountered in 2D approaches. The most significant advantage is the circumvention of the surface mapping step. Therefore, inherent obstacles in 2D, such as distortion or continuity problems, can be avoided. Instead, the textured object is "carved" from the block of texture. This works especially well for patterns that emerge from real 3D phenomena, e.g., wood or marble. However, 3D strategies perform well only with relatively homogeneous and isotropic textures. Moreover, it should be noted that not all characteristics of some specific 2D patterns can easily be reproduced with a 3D texture. Consider as a typical example a texture with uniform, filled circles. A volumetric version should contain uniformly distributed spheres; however, an arbitrary surface cut from this block would not contain circles of uniform size. In this sense, 2D approaches appear to be more flexible. Nevertheless, in this case an appropriate solution for mapping the texture to a 3D surface has to be found. Both 2D and 3D textures have been used in texture generation, and thus both will be discussed in the following.

3.5 Procedural Textures

3.5.1 Review

The notion of procedural texturing was initially coined in [208]. Instead of explicitly specifying and storing all details of a texture, they are abstracted into a function or algorithm (i.e., the procedure) which can then be evaluated on the fly. With this approach, problems with texture memory or bandwidth limitations can be avoided. Moreover, due to the procedural design, texture resolution is infinite, since magnifications can be interactively generated. Another concept which has to be mentioned in this context is solid texturing. Introduced by Ken Perlin at the same time as [94] and [206], this paradigm decouples texture from geometry and defines it in 3D space. This obviates the mapping of a 2D texture to a 3D surface, thus also avoiding any distortions or discontinuities.

The first usage of a mathematical function to define a solid texture can be found in [94]. Inspired by [229], the author uses Fourier expansions to represent real-world detail at a statistical level, for instance to represent clouds or trees.

$$T(x_s, y_s, z_s) = \sum_{i=1}^{n} c_i \frac{\big(\sin(\omega_i x_s) + P_{x_i} + 1\big)}{2} \cdot \sum_{i=1}^{n} c_i \frac{\big(\sin(\omega_i y_s) + P_{y_i} + 1\big)}{2}$$

$$(3.5)$$

where P_{x_i} and P_{y_i} represent phase shift functions which are added to avoid regularities in the pattern. c_i and ω_i provide some control over the amplitude and period of the signal.

Also in [206], basic solid texturing was applied. It is suggested that projection functions be defined to extrude 2D texture into 3D space – in the simplest case this can be an orthogonal projection, for instance, to generate 3D wood textures from concentric circles. Moreover, the usage of combination functions to merge different solid textures is also described. A typical granite texture could for example be generated by combining three basic solid textures.

The most sophisticated procedural approach has been described in [208]. It is based on the application of pseudo-noise, which is used as a numerical representation of the randomness found in natural structures. It should be noted that pseudo-noise only gives the appearance of randomness; nevertheless, it is sufficient for generating convincing textures. The core element of the algorithm is the definition of a random noise function $\nu : \mathbb{R}^n \longrightarrow \mathbb{R}$, which should be band-limited. Concentrating the energy in a small section of the frequency spectrum results in similar size and anisotropy of the apparently random variations. Moreover, the outcome should be statistically invariant under translation and rotation. Since we aim at generating solid textures, the following discussion is limited to defining ν for $n = 3$.

The first step is to define a grid of pseudo-random gradient vectors **g** positioned at 3D lattice points **q**. These random vectors are generated by a
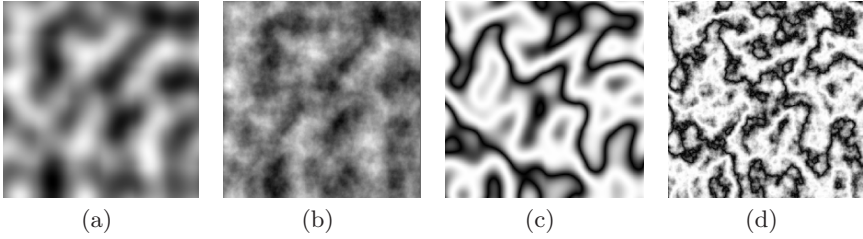
Figure 3.9. Examples of Perlin procedural textures. (a) Fractal turbulence τ with $m = 0$, (b) Fractal turbulence τ with $m = 3$, (c) Marble texture μ with $m = 0$, (d) Marble texture μ with $m = 4$

Monte Carlo simulation, which computes N points uniformly distributed on the surface of a unit sphere. The vectors are stored in an array G, which is usually periodic to limit the number of vectors. This is acceptable as long as the repetition occurs over a long distance – in practice $N = 256$ yields good results. In the next step, vectors $\mathbf{g}$ are randomly assigned to lattice points. This can be done with a heuristic indexing function based on lattice coordinates i, j, k.

$$\mathbf{g}(i,j,k) = G\Big[\tau\big[(i + \tau[(j + \tau[k \mod N]) \mod N]) \mod N\big]\Big], \quad (3.6)$$

where τ represents an array containing a precomputed random permutation of the numbers 0 to $N - 1$. After assigning the random gradient vectors using Equation 3.6, the noise function can be evaluated at an arbitrary 3D position $\mathbf{p} = (x, y, z)$ by calculating a weighted sum of inner products at the eight closest lattice points $\mathbf{q}_{i,j,k}$.

$$\nu(x, y, z) = \sum_{i=\lfloor x \rfloor}^{\lfloor x \rfloor + 1} \sum_{j=\lfloor y \rfloor}^{\lfloor y \rfloor + 1} \sum_{k=\lfloor z \rfloor}^{\lfloor z \rfloor + 1} \omega(\mathbf{p} - \mathbf{q}_{i,j,k}) \big[\mathbf{g}(i,j,k) \cdot (\mathbf{p} - \mathbf{q}_{i,j,k})\big] . \quad (3.7)$$

Here, ω represents an S-shaped, cubic drop-off filter to weigh the interpolants in each dimension.

The resulting noise function can now be used to generate textures with random appearance, either by direct mapping to colors, or as an additional parameter in a composite mathematical description. One typical approach is to generate fractal turbulence by iterating over octaves.

$$\tau(\mathbf{p}) = \sum_{i=0}^{m} \frac{\nu(2^i \cdot \mathbf{p})}{2^i} . \quad (3.8)$$

It should be noted that usually after a few harmonics the changes are less than the resolution of the texture, both in terms of spatial and grey value resolution.

Textures resembling marble can for instance be generated with

$$\mu(\mathbf{p}) = \left| cos(c_1 \mathbf{p}_1 + c_2 \mathbf{p}_2 + c_3\,\tau(\mathbf{p})) \right|, \tag{3.9}$$

where c_1 and c_2 allow control of the period of the typical marble patterns in two main directions, and c_3 denotes the strength of the turbulence. $\mathbf{p}_i$ represents the i-th component of the vector. Examples of procedural texture according to the discussed algorithm are depicted in Figure 3.9.

While Perlin's noise function provides reasonable results, there are a few shortcomings. One problem is the definition of the function, which causes it to evaluate to zero at the lattice points. This might be noticeable as a disturbing regular appearance. Since in general any noise function which fulfills the requirements stated above could be used, other approaches for procedural noise generation were also scrutinized.

In [159] a sparse convolution algorithm for three-dimensional noise is introduced. It is synthesized by the convolution of a kernel $h(\mathbf{p})$ with a Poisson noise process of impulses, which are uncorrelated in intensity and distributed at uncorrelated locations in space. Due to the impulse nature of the noise, the convolution reduces to a summation over the impulses.

$$\tilde{\nu}(\mathbf{p}) = \sum_{i=0}^{m} \alpha_i \cdot h(\mathbf{p} - \mathbf{p}_i), \tag{3.10}$$

where $\mathbf{p}_i$ is the location of the i-th impulse and α_i a scaling parameter. Using this in the more general form of Equation 3.8, a new turbulence function is obtained:

$$\tilde{\tau}(\mathbf{p}) = \sum_{i=0}^{m} \beta_i(\nu(\gamma_i \mathbf{p})). \tag{3.11}$$

The parameters β_i and γ_i can be used to individually control the amplitude and scale of all noise basis functions, respectively. As an example for the kernel h, a smooth cosine function was used to obtain a wood-like appearance.

3.5.2 Generation of Simple Textures

While the schemes discussed above are not sufficient to generate the more complex surface patterns found on most organs, they can be used to generate some of the simpler base textures, for instance those of fat or intestine. This can be accomplished by using the noise scalars to linearly interpolate between two typical colors in RGB space. These colors can be selected to be the minimal and maximal color value of a specific source image. In Figure 3.10, two generated example textures applied to artificially generated surfaces are depicted. To assess the quality of this result, the same objects are also shown integrated into a real in vivo scene. The noise was generated with Equation 3.11, using $m = 1, \beta_0 = 5.0, \beta_1 = 10.0, \gamma_0 = 1.0, \gamma_1 = 0.707$.

(a)   (b)   (c)
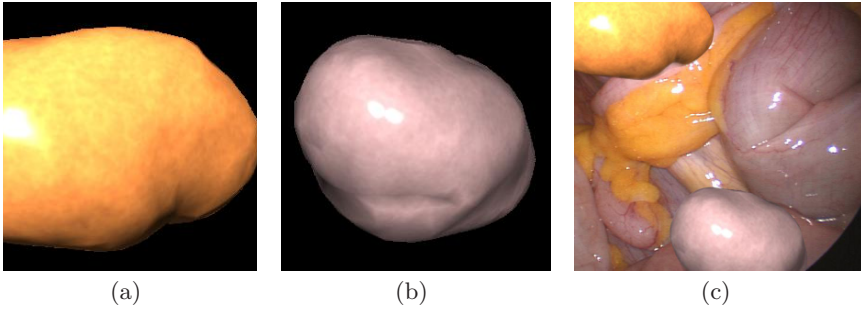
Figure 3.10. Simple textures obtained from procedural approaches. (a) Example texture resembling fat, (b) Example texture resembling bowel, (c) Sample objects integrated into in vivo image

It can be concluded that the presented approach enables the procedural generation of acceptable simple base textures. However, a number of problems have to be solved, which make the approach infeasible in most cases. Manual tuning is required to set the parameters of the noise functions, as well as specifying the correct RGB values to use for interpolation. In addition, color interpolation in RGB space can only provide meaningful output if the sample colors are relatively close to each other. Carrying out the interpolation in HSV space might be more appropriate for this. Moreover, a higher order interpolation with more sampling points might also provide better color variation.

In the context of the scene generation process, procedural texturing also does not fulfill the posed requirements. Synthesis times can be long at higher resolutions, and due to the empirical nature of the approach, several attempts might be necessary to obtain the desired results. Since the scene definition should be carried out by medical experts with the provided tools, this algorithm behavior is not acceptable. Due to the fact that also textures with higher variability should be generated, we focus mainly on texture synthesis by way of example.

3.6 Texture Synthesis

The underlying idea of approaches falling into the synthesis category is texture creation from example. A sample is supplied to an algorithm and analyzed, and then a new texture is automatically created resembling the original one. Different strategies can be used to ensure the similarity between the input and output texture. This paradigm fits quite well into the training scene generation process, since the methods can easily be controlled by novice users, i.e., the medical personnel in our case. The overall process should require as little user intervention as possible. However, if interaction should become necessary or

is desired by the user, control of the method should be intuitive. Interactive control over the placement of texture by the user should also be allowed. Nevertheless, overall synthesis quality should not be limited by providing this option. These boundary conditions will have some influence on the selection of the synthesis algorithm.

3.6.1 Human Texture Perception

A number of texture synthesis approaches are inspired by research examining the human perception of texture. This area of investigation was pioneered by the work described in [132]. Two random visual patterns, either side-by-side, or one contained in the other, were presented to test subjects. The patterns were identical in their probability distribution of the n-th order, but differed in their $(n + 1)$-th-order distribution. The test subjects had to perform a spontaneous visual discrimination task with these patterns. It was observed that textures differing in their first-order statistics can easily be discriminated, while patterns with the same first- and second-order statistics could not be distinguished. This led to the well known conjecture that textures appear indistinguishable to humans if their first- and second order statistics are identical [133].

This assumption, however, was later disproved by the same group [45, 134]. They were able to create textures with iso-second-order patterns which were distinguishable in pre-attentive human visual perception experiments. Nevertheless, inspired by the earlier work, pixel pair statistics are still very common in texture synthesis approaches. Especially for the less structured textures found on human organs, methods based on second-order distributions using pixel pairs yield good results.

More recent work based on psychophysical and neurophysiological examinations indicates that a multichannel spatial frequency and orientation analysis is performed within the visual cortex (see, e.g., [122, 179, 273, 64, 23, 178]). It was suggested that the response of cells in the primary visual cortex of mammals can be modeled with Gabor filter kernels [91]. Thus, similar textures should produce the same responses to a bank of oriented filters.

3.6.2 Review

Texture synthesis methods proposed in the past originate from diverse underlying ideas and follow different strategies. A distinction of the approaches is not straightforward, since often a combination of methods is used. One possible categorization discriminates between pixel- and patch-oriented synthesis. The former synthesizes a pixel at a time, while the latter pastes complete patches into the new texture.

Pixel-Based Synthesis

Synthesis algorithms focusing on single pixels can be further subdivided according to their sampling strategy. The first class determines new textures by matching global statistics in feature space. The second class generates pixels based on local information in the sample and result texture.

A number of methods following a *global sampling* approach have been developed. One of the first – inspired by the work of Julesz – has been reported in [92]. A pixel-wise optimization strategy is followed, minimizing a goal function with respect to selected statistical features. Given a sample texture T_{src} and a target texture T_{tgt}, pixels in the latter are changed so as to minimize the error E between the respective feature vectors $\mathbf{p}_{src}$ and $\mathbf{p}_{tgt}$. One key element of the approach is the appropriate definition of the feature vectors. One option is to model the texture by pixel pair co-occurrences at certain distances, thus capturing second-order statistical properties. Unfortunately, in such a straightforward implementation, the computation time depends quadratically on the number of texture colors. As an alternative, it was suggested that the full co-occurrence matrices be replaced with simpler models based on moments of various order, for instance, second-order spatial moments combined with first-order spatial averages (i.e., autocovariance with intensity histograms). However, this resulted in reduced synthesis quality.

A different paradigm was followed in [117], inspired by psychophysical studies which assume that a bank of space localized filters are underlying human texture perception to discriminate texture fields. The method consists of matching histograms at different levels of a Laplacian or steerable texture pyramid. The first phase is the analysis of a sample texture, which consists of building an image pyramid where the levels are obtained by convolution and subsampling. The same is done for a white noise image, which is modified iteratively in the second phase to match the appearance of the sample texture. This is achieved by multiscale matching of marginal histograms. While the Laplacian pyramid is suited for isotropic textures, the steerable version enables the synthesis of textures with some oriented structures. Although there is no proof of convergence, reasonable results are obtained after a few iterations. Nevertheless, the technique performs only well on basic stochastic homogeneous textures. Moreover, performing too many matching iterations can introduce artifacts in the synthesized image.

A method related to the previous work was introduced in [31]. Again a multiresolution filter-based approach is applied, following a coarse-to-fine manner. The algorithm randomizes an input texture sample while preserving the cross-scale dependencies. First an analysis pyramid is constructed based on responses to a bank of oriented first- and second-order Gaussian derivatives. Then the synthesis is carried out on subsequent levels by building a set of candidate pixels for a specific texture location. The candidates are selected if the distances of features between the analysis pixel feature vector and the synthesized feature vector is below a set of selected thresholds. Difficulties in

this method are due to the nonintuitive tuning of the threshold parameters; especially, since the outcome is very sensitive to this choice. Additionally, boundary artifacts are produced when the textures are not tileable.

Another approach following a multiresolution paradigm has been suggested in [244, 217]. This method is based on matching first- and second-order properties of wavelet coefficients and coefficient pairs of steerable pyramids. Pixels in the image pyramid are updated according to iterative orthogonal projections from the filter response of the synthetic texture onto that of the sample.

A combination of multiscale feature matching with Markov Random Field (MRF) methods – the FRAME model – has been presented in [309, 308]. It builds on comparing marginal histograms of Gabor filter bank responses as features. The synthetic texture is updated according to MRF probability functions. While a solid mathematical framework is presented, the method is computationally quite expensive.

A difficulty, which hampers the application of all these multiscale approaches in our specific texture generation process is the masking of pixels. During the convolution steps between the different subbands, unknown texture values have to be propagated, which reduces the quality of the overall synthesis process.

For completeness, the work described in [100, 101] should also be mentioned. Instead of matching global features, the authors suggest performing a global spectral analysis of a texture sample in order to obtain basis and perturbation functions for procedural texture generation. A 2D texture sample T_s is approximated by using a summation of cosines.

$$r(x, y) = A_0 + \sum_{i=1}^{N} A_i \cos\left(2\pi(f_i x + g_i y) + \theta_i\right) . \tag{3.12}$$

Here, A_0 is the average value of the texture, A_i the amplitude of the i-th significant term, f_i, g_i its frequencies, and θ_i its phase, respectively. The cosines are determined by spatial partitioning of the spectrum of T_s and selecting the dominant signal in each bin. A new texture T_n, which can be 2D or 3D, can then be synthesized by using a pseudorandom signal. A problem is the control of the variation along the third dimension, since the texture remains a 2D pattern, which is perturbed along the third axis. This situation can, however, be partly alleviated by using additional samples representing different orthogonal slices of a structure.

In contrast to the methods discussed above, several approaches follow *local sampling strategies*. The majority of these algorithms model texture as a Markov Random Field and generate new images by stochastic sampling.

A core assumption of MRF is *locality*, that is, the value of a pixel is predictable from a finite set of neighboring pixels, and independent of the rest. This property of spatial Markovianity is described by the local conditional probability density function (LCPDF) (see, e.g., [24, 97]).

$$P(X_s = x_s | X_r = x_r, r \neq s) = P(x_s | x_r, r \in \mathcal{N}_s), \qquad s \in S, x_s \in \mathcal{L} \quad (3.13)$$

where $x_s \in \mathcal{L}$ is the state of the variable X_s at site s on texture lattice S, and $\mathcal{N}_s \subset S$ is a neighborhood of s. The second property of MRF is *stationarity*. This usually refers to the fact that the marginal probability of any texture window remains invariant under translation.

In order to calculate 3.13, the equation is reformulated according to the Hammersly-Clifford theorem [24]. Under positivity condition on a finite lattice, an MRF can be rewritten as a Gibbs random field, which assigns a probability to each realization $X = x$ in the MRF.

$$P(X = x) = \frac{1}{Z} e^{\left(-\sum_{C \subset S} V_C(x)\right)}. \quad (3.14)$$

Here, V_C assigns a potential based on site values to the clique C, which is a subgraph of S where all sites are neighbors, and Z is a normalizing constant. The conditional probability can then be obtained using the defined potential function.

$$P(X_s = x_s | X_r = x_r, r \neq s) = \frac{e^{\left(-\sum_{s \in C} V_C(x)\right)}}{\sum_{\lambda \in \mathcal{L}} e^{\left(-\sum_{s \in C} V_C(\lambda, x_{\neq i})\right)}}. \quad (3.15)$$

New texture can thus be synthesized by drawing samples from the distribution P. Two distinct strategies can be identified to carry out this step. *Parametric* approaches try to learn the parameters of the assumed underlying Gibbs distribution from samples. Unfortunately, this is hampered by the high dimensionality of the approach. The main problem is the fitting of all parameters for larger neighborhood sizes, which is computationally intractable. Therefore, *nonparametric* methods refrain from explicitly determining the MRF model.

The former approach was applied in [62]. The authors assumed autobinomial conditional probabilities for the clique potentials as a texture model. Second-order statistics of particular textures were compared to those predicted by the MRF.

A cluster-based, semiparametric model was introduced in [216]. The probability function was compressed with discretized Gaussian kernels.

$$P(x) = \sum_{i=1}^{N} w_i \left(\prod_{j=1}^{M} G_{i,j}(x) \right), \qquad w_i > 0, \sum w_i = 1 \quad (3.16)$$

where w_i are weighting parameters. According to the restricted probability function, the value of a new pixel is then determined in a fixed order within a causal neighborhood. Moreover, to reduce complexity, a hierarchical approach is applied.

In contrast to the previously discussed methods, nonparametric texture synthesis approaches do not require the direct estimation of parameters in statistical models.

Initial work in this direction has been described in [204]. The authors developed a noncausal, nonparametric, multiscale MRF model capable of synthesizing a large range of textures. The local conditional probability density function was determined to obtain the probability of a pixel value in a noncausal neighborhood. A nonparametric estimate of the LCPDF of a sample image was obtained by building its multiscale histogram. A Parzen-window density estimator is applied to the multiscale histogram in order to determine the LCPDF. Pixelwise stochastic relaxation is then carried out to synthesize a new texture. Moreover, they augmented their coarse-to-fine approach with local annealing and parallelization of the relaxation.

In [79] a nonparametric sampling approach was described. Assuming an MRF model, a probability distribution for a target pixel is obtained based on its already synthesized neighbors, and on windows of the sample texture with similar neighborhoods. An exhaustive nearest neighborhood search has to be performed for each target pixel. A new image is grown in a spiral, one pixel at a time, starting from an arbitrarily placed 3x3 seed. In general, the neighborhood window has to be large enough to capture local structures completely. Moreover, problems can occur when no good matches can be found. The algorithm can get lost in the search space and start sampling randomly which produces incorrect results.

A similar technique was suggested in [290]. Again, no explicit probability distribution was constructed. The authors synthesized a new texture pixel by pixel in scanline order, while preserving local similarity. This was done by considering an L-shaped neighborhood of fixed size in the current output image, and searching the input sample for the candidate with the most similar neighborhood, which was then copied into the output image. Again, this method is quite slow, due to the extensive searching process. Therefore, the authors suggest accelerating the neighbor search with tree-structured vector quantization (TSVQ). A further improvement in computation time could be achieved by incorporating a multiresolution pyramid. These enhancements make the algorithm two orders of magnitude faster then the previous one. Unfortunately, the method tends to introduce some blurring into the results.

A further improvement of the nearest neighbor search is described in [8]. It is based on the observation that neighbors to a pixel in the output image often come from locations in the sample image which are near to the sources of pixels in its current neighborhood. Thus instead of an exhaustive search, a limited number of candidates is generated, based on the original locations of the current neighborhood pixels. To this end, a coherence map is maintained, which stores source locations of synthesized pixels.

Patch-Based Synthesis

In recent years a different paradigm for texture synthesis has emerged. No explicit mathematical texture model is developed, but instead a heuristic approach is followed, in which new textures are obtained by copying complete patches from example images. Usually, this is performed in an iterative fashion, where new patches are selected according to some error metric. The latter ensures an optimal fitting of overlap regions at the patch edges. The overall process can be influenced by selecting the patch size or the extent of the overlap regions. The main focus of most methods is the treatment of the patch boundaries, as well as employing optimal search strategies to select the patches.

The first work following a patch-based strategy – termed *chaos mosaics* – was reported in [300]. Instead of copying individual pixels, the authors randomly placed source texture patches in the output image. Their biggest problem was feature discontinuities across seams. In later work they overcame some of these problems with simple cross-edge filtering [299]. A key feature of their method was the fast synthesis of output images with reasonable quality.

Related to the previous method, in [219] *lapped textures* are introduced, which can be used to cover arbitrary triangle meshes with example textures. Seams across patches are avoided with alpha blending. A robust flattening approach is used to minimize distortions and match local orientation.

A synthesis process termed *image quilting* is outlined in [80]. New textures are synthesized one patch at a time in scanline order by placing overlapping square samples from a source image. Candidate patches are selected randomly from the source so as to minimize the L2 norm for pixels in the overlapping area. Once a square patch has been found, an optimal cutting patch between adjacent patches is determined. The authors apply dynamic programming to perform this minimum error boundary cut.

Similarly, in [161] another patch-based sampling strategy is suggested. Instead of determining an optimal cut between patches, boundary errors are alleviated by simple blending. In addition, a number of algorithmic improvements are made to enable real-time user interaction. An approximate nearest neighbor search scheme is followed for patch selection. Principal component analysis is applied to reduce the dimensionality in this step. Moreover, further enhancements could be achieved by using hierarchical data structures, e.g., kd-trees to find the nearest neighbor or quad-trees to select initial candidates.

Improved handling of patch overlap regions used in a hybrid synthesis approach is described in [192]. A multiscale paradigm is followed to minimize the border error. After a candidate patch is selected, the error of the overlap region is examined. If it exceeds a predefined threshold, a smaller patch is resynthesized and tested again. If necessary, this process is iterated down to individual pixels. In order to speed up the candidate search, it is carried out in the Fourier domain. This allows a quick calculation of the error with a fast Fourier transform (FFT).

In [150] the treatment of the patch boundaries is formulated as a graph cut problem. By representing the overlap region as a graph, the task of finding the best seam turns into the search for a minimum-cost graph cut. Again, an FFT-based acceleration is applied by carrying out patch searching in the Fourier domain.

Patch-based texture synthesis provides good results at reasonable processing times; however, a number of limitations exist which make their usage complicated in our application area. One problem is due to the image masks present in our texture samples, which limit the candidate search. Another drawback is the recalculation of texture patches necessary if affine transformations are to be applied to the texture, which reduces interactivity during the scene generation process. Finally, due to the patch oriented nature of these methods, stochastic variation can be diminished compared to pixel-wise strategies. For these reasons, pixel-wise synthesis approaches are mainly employed. In the following two sections, the methods applied to generate textures for laparoscopic as well as hysteroscopic scenes are discussed in more detail.

3.6.3 Texture Generation for Laparoscopic Simulation

In the context of simulation of surgical interventions in laparoscopy, a two-stage texture generation process can be followed [182]. The main step consists of pixel-based texture synthesis. It provides base textures of moderate stochastic variation, which is sufficient for the majority of the abdominal organs. If required, these textures can then be enhanced by adding low-frequency detail in a second step, following a procedural approach.

The synthesis method applied is the one detailed in [92]. Inspired by Julesz' conjecture, it models texture by pixel-pair statistics. An advantage of this approach is its capability of dealing with masking, which is often necessary due to the low quality of our source images. Moreover, the method can easily be extended to 3D synthesis, which simplifies the required texture mapping to object meshes. Furthermore, the technique allows the integration of directional control of the synthesis process, which permits the generation of anisotropic textures. Finally, varying levels of synthesis quality can be achieved by applying different statistical models for determining the texture feature vectors. This allows for the selection of an appropriate model depending on the character of the sample image, while the texture generation procedure stays the same.

The main idea of this approach is to minimize the error between feature vectors containing a statistical description of a source and a target texture. First, for a given sample image T_{src}, the feature vector $\mathbf{p}_{src}$ is determined. Similarly, the vector $\mathbf{p}_{tgt}$ of an arbitrary sized target texture T_{tgt} is obtained. Following an optimization in a least squares sense, pixels in the output image are changed, so as to minimize the error between the feature vectors.

$$E = \|\mathbf{p}_{tgt} - \mathbf{p}_{src}\|^2 \overset{!}{=} min . \tag{3.17}$$

Texture lattice
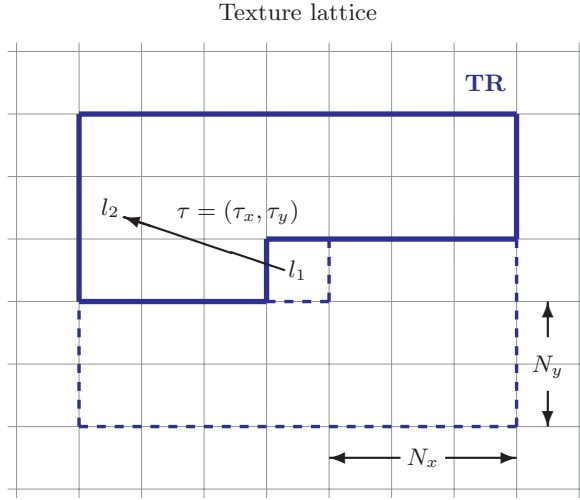
Figure 3.11. Translation window TR for determining co-occurrences

To initialize this procedure, the target texture is filled with noise, which should match the histogram of that of the source image. This can be done by randomly copying pixels of the sample texture to the target lattice. While this is only an approximation of the source histogram, the resulting deviation does not influence the synthesis approach. Moreover, since first-order statistics are part of the applied texture models, this will automatically be adapted during the optimization. Finally, the random initialization ensures variation of the output samples.

After this initial setup step, the algorithm iteratively visits locations in the output lattice in a random order. The current texel value l is replaced with the value l', which maximally decreases the squared Euclidean feature vector difference. Recomputing the whole vector $\mathbf{p}_{tgt}$ to determine the error E would tremendously decrease performance. However, this is not necessary. Instead, it is sufficient to just determine the change of error Δ_E resulting from replacing l with l' in the target texture.

$$\Delta_E = E(l) - E(l') . \tag{3.18}$$

Thus for each location the algorithm visits, we have to determine the value l', which maximizes Δ_E. If such a value can be found, l is replaced by l', and the feature vector $\mathbf{p}_{tgt}$ as well as the error E have to be updated.

Following the outline of the algorithm, we now have to define the statistical models which should be used to determine the feature vectors. While a number of approaches would be possible, this discussion focuses on the co-occurrence and the autocorrelation models.

Co-occurrence Model

This model is based on the co-occurrence of pixel color pairs in the texture lattice with a relative distance of τ. This translation vector is usually encoded with the tuple (τ_x, τ_y) denoting the L_1-distance in the x- and y-direction, respectively. The probability of co-occurrence of colors $l_1, l_2 \in \mathcal{L} = \{1, \dots, L\}$ at distance τ is given by:

$$r_\tau(l_1, l_2) = \frac{1}{N} \sum_{i=1}^{N} \delta(t_i - l_1) \cdot \delta(t_{i+\tau} - l_2). \qquad (3.19)$$

Here, N is the total number of texture pixels t in the image lattice. They are indexed in 1D fashion, with t_i being at location (x, y) and $t_{i+\tau}$ at location $(x + \tau_x, y + \tau_y)$. By determining all probabilities for all colors $l_1, l_2 \in \mathcal{L}$ for a specific translation τ_k, we can setup a co-occurrence matrix.

$$\mathbf{M}_2(\tau_k) = \begin{pmatrix} r_{\tau_k}(l_1, l_1) & \cdots & r_{\tau_k}(l_1, l_{|\mathcal{L}|}) \\ \vdots & \ddots & \vdots \\ r_{\tau_k}(l_1, l_{|\mathcal{L}|}) & \cdots & r_{\tau_k}(l_{|\mathcal{L}|}, l_{|\mathcal{L}|}) \end{pmatrix}. \qquad (3.20)$$

By combining all matrices $\mathbf{M}_2(\tau_k)$ for all translation vectors τ_k in the texture lattice T, we could build the feature vector $\mathbf{p}$. However, its size linearly depends on the number of translations and quadratically on the number of colors. Therefore, some reductions in size have to be made to keep the problem manageable. The translations can be restricted to a certain distance. This is done by considering just a limited window of translations. Moreover, by exploiting the fact that $r_\tau(l_1, l_2) = r_{-\tau}(l_2, l_1)$, we can further reduce the number of translations. Hence, the translation window (see Figure 3.11) is defined according to

$$TR = \left\{ \tau = (\tau_x, \tau_y) \left| \begin{array}{l} (0 \le |\tau_x| \le N_x \wedge 1 \le \tau_y \le N_y) \quad \vee \\ (-N_x \le \tau_x \le -1 \wedge \tau_y = 0) \\ \tau_x, \tau_y, N_x, N_y \in I\!N \end{array} \right. \right\}. \qquad (3.21)$$

Therefore, the number of parameters in the vector needed to describe textures by using second-order spatial averages is finally given by $|\mathcal{L}|^2 \cdot (2N_x N_y + N_x + N_y)$. After setting up the initial feature vectors, the algorithm randomly selects positions t_i in the lattice of the target texture. For these positions the texture value which maximizes Δ_E has to be found. Since we limited the translations to the window TR, we only have to examine the positions $t_{i+\tau}$ and $t_{i-\tau}$ for a specific translation τ. Instead of recomputing the whole error, we can just update the error change. If l^+ is the color at $t_{i+\tau}$, and l^- the one at $t_{i-\tau}$, then a replacement of l with l' at t_i necessitates the replacement of

co-occurrence (l, l^+) with (l', l^+), and (l^-, l) with (l^-, l') in the feature vector $\mathbf{p}_{tgt}$. Thus, the vector update for the translation τ can be carried out following the scheme

$$r_\tau^{tgt}(l, l^+) := r_\tau^{tgt}(l, l^+) - \frac{1}{N_\tau^{tgt}} \tag{3.22}$$

$$r_\tau^{tgt}(l', l^+) := r_\tau^{tgt}(l', l^+) + \frac{1}{N_\tau^{tgt}}$$

$$r_\tau^{tgt}(l^-, l) := r_\tau^{tgt}(l^-, l) - \frac{1}{N_\tau^{tgt}}$$

$$r_\tau^{tgt}(l^-, l') := r_\tau^{tgt}(l^-, l') + \frac{1}{N_\tau^{tgt}} \ .$$

Here, N_τ^{tgt} denotes the total number of pixel-pairs lying inside the output image lattice for the specific translation τ. This update has to be applied for all $\tau \in TR$. Based on the vector updates of Equation 3.22, the resulting contribution to the error change can be determined. When replacing l with l' the error change is given by

$$\Delta_E(l') = \sum_{\tau \in TR} \left[\frac{4}{(N_\tau^{tgt})^2} + \frac{2}{N_\tau^{tgt}} \cdot \right.$$
$$\left(r_\tau^{src}(l, l^+) - r_\tau^{tgt}(l, l^+) + r_\tau^{tgt}(l', l^+) - r_\tau^{src}(l', l^+) + \right. \tag{3.23}$$
$$\left. \left. r_\tau^{src}(l^-, l) - r_\tau^{tgt}(l^-, l) + r_\tau^{tgt}(l^-, l') - r_\tau^{src}(l^-, l') \right) \right] \ .$$

Using this equation, we can now determine the value $l^* \in \mathcal{L} - \{l\}$, which maximizes Δ_E. If the error change is less than or equal to zero, then we already have found the optimal value and can continue with a new location in the lattice. Otherwise, we replace l with l^*, and update $\mathbf{p}_{tgt}$ according to Equation 3.22, as well as the error $E := E + \Delta_E(l^*)$. It should be noted that in order to deal with the boundary, the texture is regarded topologically as a torus by joining opposite edges. This also produces output images, which tile seamlessly. Example textures synthesized with this algorithm can be seen in Figure 3.12.

While the convergence of the algorithm has not been proven theoretically, in practice the error monotonically decreases with each scan. The lower bound of the error is zero; however, one can not guarantee a perfect fit of the co-occurrences (i.e., zero error). Typically, the optimization can be stopped after 6 iterations, where only a small percentage of pixels is changed. This change rate can actually be used as a termination criterion. Moreover, it should be noted that a fundamental element of the method is the random selection of pixel locations. If a scanline order visitation scheme were followed, the error decrease would not be homogeneous over the whole domain, which would become visible as artifacts in the output texture.

(a)   (b)   (c)   (d)
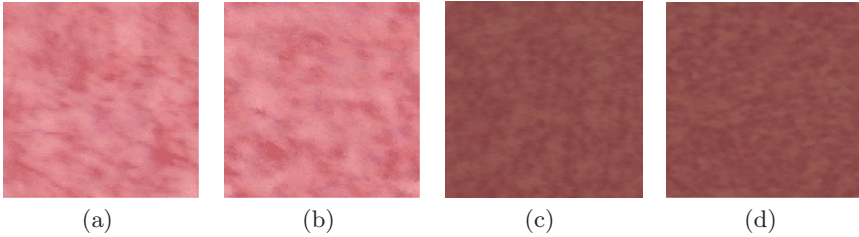
Figure 3.12. Texture synthesis with co-occurrence model. (a) Drift-corrected, quantized uterus texture sample, (b) Synthesized uterus texture, (c) Drift-corrected, quantized liver texture sample, (d) Synthesized liver texture

A major drawback of the texture model is the large number of parameters in the optimization. In order to synthesize textures of sufficient quality, the translation window has to be large enough. This leads to a considerable increase in computation time, which renders the method unfit for interactive scene definition purposes. A reasonable alternative is the autocorrelation model, which replaces the full co-occurrence matrices with second-order spatial moments.

Autocorrelation Model

According to the Wiener-Khintchine theorem, the Fourier transformation of the autocorrelation function describes the power spectrum of a signal.

$$J(\omega) = \int_{-\infty}^{\infty} K(\tau)e^{-i\omega\tau}d\tau \qquad (3.24)$$

where the autocorrelation is defined as

$$K(\tau) = \frac{1}{N\sigma^2} \sum_{i=1}^{N}(t_i - \mu)(t_{i+\tau} - \mu) \qquad (3.25)$$

with σ^2 denoting variance and μ mean of the texture signal. The theorem indicates that the autocorrelation function contains some information about frequency and orientation in a texture. Therefore, it seems appropriate to use it as a reduced statistical model for describing texture. However, since it only provides the statistical moments, the histogram of the texture also has to be taken into account by determining

$$H(l) = \frac{1}{M} \sum_{i=1}^{M} \delta(t_i - l) \qquad \forall l \in \mathcal{L} \qquad (3.26)$$

with M being the number of pixels in the texture lattice. Combining the histogram and the autocorrelation values for translations $\tau \in TR$, we form

(a)   (b)   (c)   (d)
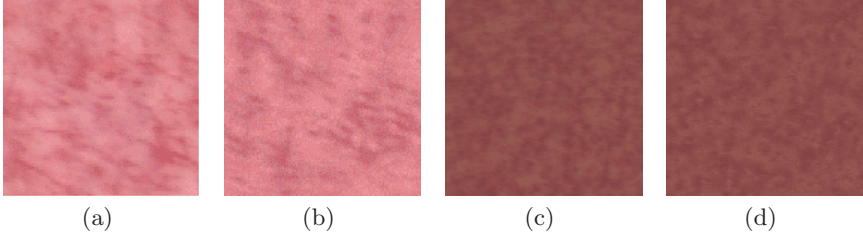
Figure 3.13. Texture synthesis with autocorrelation model. (a) Drift-corrected, quantized uterus texture sample, (b) Synthesized uterus texture, (c) Drift-corrected, quantized liver texture sample, (d) Synthesized liver texture

feature vectors $\mathbf{p}_{src}$ and $\mathbf{p}_{tgt}$. For this model, the length of the vectors is reduced to $|\mathcal{L}| + (2N_x N_y + N_x + N_y)$.

The different definition of the feature vector also necessitates a new error metric. Optimally, the histograms H_{src} and H_{tgt} should be identical; however, due to the size of the optimization space, constraint optimization is not practicable. Therefore, we minimize the error according to:

$$E = \|\mathbf{p}_{src}^{K} - \mathbf{p}_{tgt}^{K}\|^2 + \alpha\|\mathbf{p}_{src}^{H} - \mathbf{p}_{tgt}^{H}\| \stackrel{!}{=} min \qquad (3.27)$$

where $\mathbf{p}_{<>}^{K}$ and $\mathbf{p}_{<>}^{H}$ denote the subvectors containing the autocorrelation and histogram values, respectively. The parameter α has to be selected manually to control histogram equality. As in the previous model, optimization is carried out by maximizing the error change. When replacing pixel l with l' at position t_i, first the histogram has to be updated:

$$H_{tgt}(l) := H_{tgt}(l) - \frac{1}{M_{tgt}} \qquad (3.28)$$

$$H_{tgt}(l') := H_{tgt}(l') + \frac{1}{M_{tgt}} \ .$$

Thereafter, the autocorrelation has to be recomputed. Again, for a specific translation vector τ, values l^- and l^+ are affected.

$$K_{tgt}(\tau) := K_{tgt}(\tau) + \frac{1}{N_\tau \sigma^2}(l' - l)(l^- + l^+ - 2\mu) \qquad (3.29)$$

where σ is the standard deviation, and μ the mean of the texture lattice T. As in the previous model, we also have to determine the resulting contribution to the error change, when replacing l with l'. Since in general α should be set high enough to ensure histogram equality, we can assume that only minute changes of the mean value μ will happen. By considering the value as being constant, we can avoid the complete recomputation of the moments, and again perform local error updates.

Red channel   Green channel   Blue channel

Original image
(uterus sample)

Red-Green   Green-Blue   Blue-Red
correlation   correlation   correlation
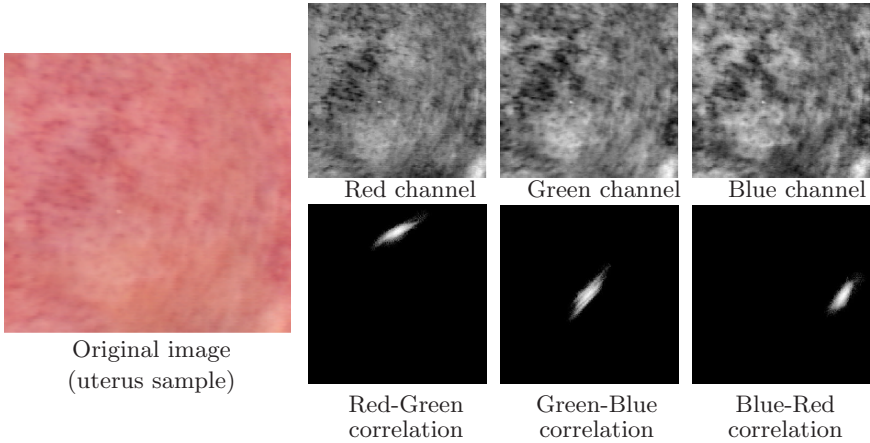
Figure 3.14. Correlation of color channels

$$
\begin{aligned}
\Delta_E(l') = {} & \frac{2\alpha}{M}\left(H_{src}(l) - H_{tgt}(l) - H_{src}(l') + H_{tgt}(l') + \frac{1}{M}\right) \\
& + \frac{l' - l}{\sigma^2}\sum_{\tau \in TR}\left[\frac{1}{N_\tau^2\sigma^2}(l' - l)(l^- + l^+ - 2\mu)^2 \right. \\
& \left. - \frac{2}{N_\tau}(l^- + l^+ - 2\mu)(K_{src}(\tau) - K_{tgt}(\tau))\right].
\end{aligned}
\tag{3.30}
$$

After determining the optimal value $l^* \in \mathcal{L} - \{l\}$, the same update strategy is followed as before. In general, the quality of the synthesis process is reduced due to the model simplification. However, reasonable results can still be achieved due to the rather stochastic nature of the organ textures. Examples synthesized using the same sample images as with the previous method are depicted in Figure 3.13.

Algorithm Enhancements

Several improvements or extensions are possible based on the texture synthesis approaches discussed above. Some of these will be outlined in this section.

One important aspect is the *treatment of color* in the synthesis methods. So far, the discussion has been limited to samples using small numbers of indexed colors resulting from a quantization step. While reasonable results can be achieved with these reduced color models, the usage of full RGB images as source textures for the synthesis process should also be examined. The straightforward approach would be to reformulate the synthesis algorithm for three-dimensional vectors containing the red, green, and blue channel of the image. Unfortunately, due to the high correlation between the channels, a

(a)   (b)   (c)   (d)
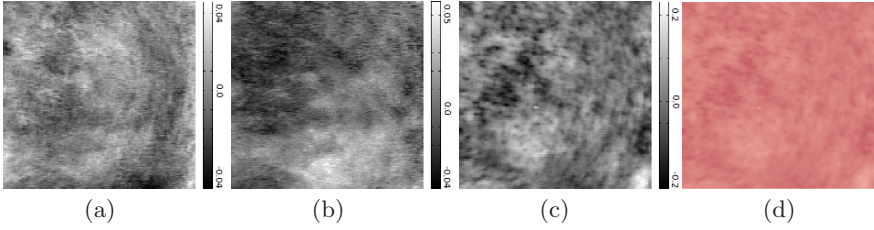
Figure 3.15. Decorrelation of color channels (normalized view). (a) New color channel 1, (b) New color channel 2, (c) New color channel 3 (primary channel), (d) Color image based on primary channel

separate treatment of individual channels is not possible. An example of the color correlation for a typical texture can be seen in Figure 3.14. The top row shows the red, green, and blue components of the source image, while the bottom row depicts the color distribution projected on the Red-Green, Green-Blue, or Blue-Red surface of the RGB cube, respectively. A clear linear correlation between the color channels is visible.

One option to amend this situation is the linear decorrelation of the color channels via the Principal Component Analysis, thus defining a new, texture-specific color space. This can be done by determining the eigenvalue decomposition

$$\mathbf{C} = \mathbf{P}\tilde{\mathbf{C}}\mathbf{P}^T \tag{3.31}$$

of the texture covariance matrix

$$\mathbf{C} = \frac{1}{N}\sum_i (\mathbf{T}_i - \boldsymbol{\mu})(\mathbf{T}_i - \boldsymbol{\mu}^T) \tag{3.32}$$

where $\mathbf{T}_i$ is the RGB value of texel i, $\boldsymbol{\mu}$ the vector of channel means of texture $\mathbf{T}$ containing N texels. Matrix $\mathbf{C}$ is square, real, and symmetric, with orthogonal eigenvectors. Matrix $\mathbf{P}$ defines a linear transformation from color space $\mathcal{C}$ into color space $\tilde{\mathcal{C}}$. Since $\tilde{\mathbf{C}}$ is diagonal, the three color channels are linearly decorrelated. By subtracting the mean $\tilde{\boldsymbol{\mu}}$ from the decorrelated color channels, we obtain a new color space with mean values zero. However, it should be noted that the PCA captures only linear correlation to reduce dimension, but fails to detect nonlinear components. Since some nonlinear dependency is also present in organ textures, the new color channels are not fully independent. Nevertheless, the nonlinear component is in general negligible. Figure 3.15 depicts the obtained linearly decorrelated color channels for the previously used example texture. Moreover, for the highly correlated organ textures, it can usually be noticed that a primary channel exists, which contains most of the texture color information. This allows one to focus the synthesis process only on this channel. The final texture can then be obtained by setting the other

<center>(a)   (b)   (c)   (d)</center>
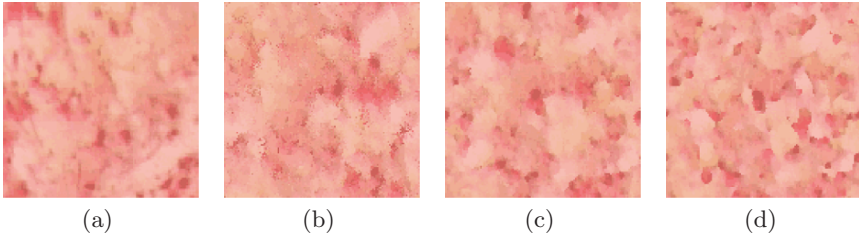
Figure 3.16. Synthesis examples using distance weighting with varying β. (a) Original texture of endometrium, (b) Synthesized with $\beta = 0.0$ (unmodified version), (c) Synthesized with $\beta = 2.0$, (d) Synthesized with $\beta = 3.0$

two channels to zero, and transforming the primary channel back into color space $\mathcal{C}$. An example of the backtransformation of the primary channel is also visualized in Figure 3.15. The resulting texture can not easily be distinguished from the original one.

An additional option to control the texture synthesis process is the integration of a *distance weighting* function. A problem of the general formulation is the unbalanced contribution of different translations τ_k to the optimization. Since a larger number of texels is included in the computation for longer translations, mainly coarse structures are formed in the initial optimization steps. This results in inaccuracies of fine detail, which becomes apparent as slightly noticeable high-frequency noise in the image. To avoid this problem, in Equation 3.23 the contribution to the error update of different translations can be scaled with a weighting function $w(\tau_k)$. One possibility for the weighting could be a Gaussian filter. A simpler function, also providing good results, applies weighting according to

$$w(\tau_k) = \frac{1}{(\|\tau_k\|_2)^\beta} \ . \tag{3.33}$$

The synthesis process can be influenced by selecting parameter β. For the examined organ textures, values of $\beta \approx 2$ lead to improved synthesis output. An example of the influence of the parameter on the process is shown in Figure 3.16.

While the introduction of a weighting function is a rather heuristic step, a more formal approach can be taken using information theory principles to compute *transinformation*. This measure (sometimes also referred to as mutual information) denotes how much information can be obtained about one random variable by observing another. Generally, the transinformation of X relative to Y is given by

$$I(X,Y) = \sum_{x,y} p(x,y) \log \left(\frac{p(x|y)}{p(x)} \right) \ . \tag{3.34}$$

(a)   (b)   (c)   (d)
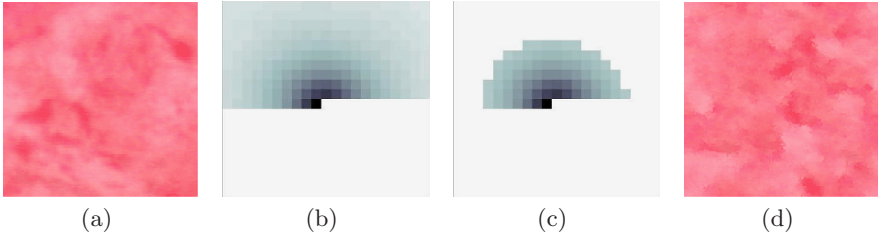
Figure 3.17. Transinformation-based reduction of translation window. (a) Original uterus texture, (b) Transinformation for window TR with $N_x = N_y = 10$, (c) Reduced set of translations TR', (d) Synthesized texture with TR'

For the synthesis process, this model can be applied to determine how much information about a texture is contributed by a specific translation τ. Using the previously defined co-occurrence $r_\tau(l_1, l_2)$, as well as $r_\tau(l_1|l_2)$, which denotes the conditional probability of texel i having color l_1, given that texel $i + \tau$ has color l_2, we can compute the transinformation $I(i, i + \tau)$ for a specific τ by examining all pairs of colors $l_1, l_2 \in \mathcal{L}$. Determining this for all translations $\tau \in TR$, we can select only those which contribute more information than a user-defined threshold. This approach limits the number of translations, and thus also the overall computation time, while the synthesis quality is more or less maintained. An example is depicted in Figure 3.17. By removing all translations which contribute less than 20% of the maximal transinformation, the number of translations in a window $TR(N_x = N_y = 10)$ can be reduced from 220 to 77. It is interesting to note that the profile of the transinformation for the type of texture we examine is similar to the weighting applied in the previous method. Thus, due to the radially decreasing weights, the previous method also increases the influence of nearby texels, while more distant locations are neglected. By building a thresholded transinformation window, however, the number of computations can be significantly reduced. Unfortunately, for both methods some iterative tuning of the controlling parameters – weighting function or transinformation threshold – is required to obtain the desired results.

The synthesis techniques discussed so far have focused only on obtaining isotropic output images. However, for some specific organ surface patterns it would be useful to also allow generation of *anisotropic textures*. This is for instance true for the uterus, where texture is oriented along muscle fibers. In order to include anisotropy into the synthesis process, we have to extend the approach by considering the local coordinate systems in the texture, which define the orientation at that position. When considering a specific translation $\tau = (\tau_x, \tau_y)$ in window TR for the current texture location, we can apply a mapping to the local coordinate system via a rotation θ and optionally also a scaling (u_x, u_y). This gives a local set of translations $\tilde{\tau}$.

Rotated and scaled translation
window at current location

Original texture

Synthesized aniso-
tropic texture
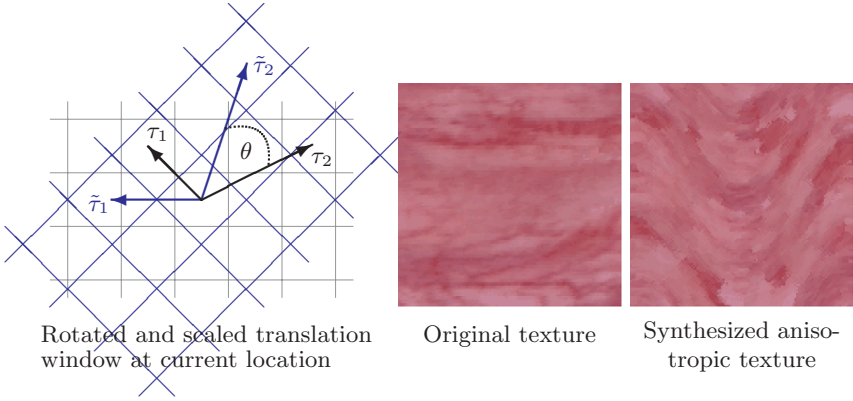
Figure 3.18. Rotation of translation window to include local orientation in the synthesis process. The example shows a sinusoidal wave overlaid on the texture lattice

$$\tilde{\tau}_x = u_x \tau_x \cos(\theta) - u_y \tau_y \sin(\theta) \qquad (3.35)$$
$$\tilde{\tau}_y = u_x \tau_x \sin(\theta) + u_y \tau_y \cos(\theta) \, .$$

Since the new translations usually do not coincide with the texture lattice, the color has to be bilinearly interpolated. If a quantization has been performed for the sample texture, then the new color might not be part of the histogram, and thus it has to be mapped to the closest color value. It should also be noted that for indexed color images, a bilinear interpolation cannot be done. Instead the color of the nearest neighbor has to be selected. An example of including local orientation into the synthesis is visualized in Figure 3.18. A sinusoidal wave is used to obtain local orientation θ at each lattice point, while no scaling is applied. A texture with prominent horizontal patterns is used as the source image to better visualize the effect. It has to be mentioned that for more isotropic textures, the visual effect is reduced and sometimes almost not visible.

While this process allows the inclusion of local orientation into the synthesis process, the question of how to obtain these local patterns still has to be addressed. The anisotropies have to match the actual geometry of the object to be textured. Thus an orientation field has to be defined on an arbitrary triangular mesh. In practice, a user would specify a few major orientations on the mesh, while the remaining ones are obtained from an optimization process, ensuring smooth transitions. An orientation at a specific vertex i can be described by a local coordinate system $(\mathbf{u}_i, \mathbf{v}_i, \mathbf{n}_i)$. Since $\mathbf{n}_i$ is the normal at the vertex, the remaining degree of freedom is rotation θ_i around it. Thus in an optimization process, the difference between the rotations of the local coordinate systems at vertices i and j, which are connected by edge (i, j), has to be minimized. A global energy term can be defined according to

(a)   (b)   (c)
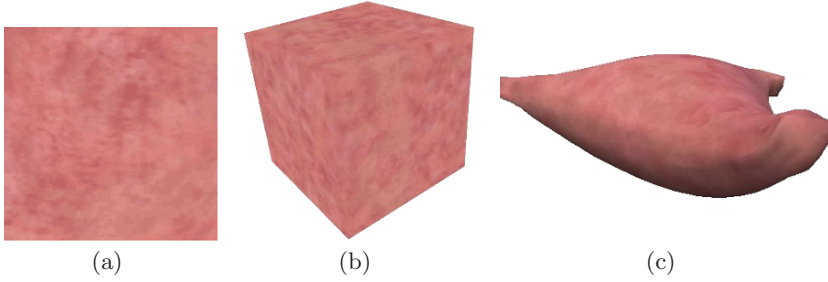
Figure 3.19. Synthesis of isotropic 3D texture, (a) Original uterus texture, (b) Synthesized texture block, (c) Uterus surfaces carved out of 3D texture

$$E_\theta = \sum_{(i,j)} \frac{1}{|(i,j)|} \cdot \frac{1 - \cos\left(h(\theta_i, \theta_j)\right)}{2} \tag{3.36}$$

where $h(\theta_i, \theta_j)$ is a scalar valued function measuring the difference between the orientation of the local coordinate systems connected by an edge. After optimization by minimizing the energy term, a vector field with smooth transitions is obtained.

In the previous sections mainly 2D textures were considered; however, this approach can also be extended to generate *3D textures*. Assuming that the examined texture is isotropic, a statistical description can be built without taking the lattice orientation into account. Instead of using specific translation vectors τ, only the length of the vectors affects the model, thus making it independent of the dimension. Therefore, the statistical model for a specific 3D translation vector can be obtained from a 2D vector with the same length. It should be noted that vectors in space might occur for which a specific translation in 2D with corresponding length cannot be found. In these cases, the nearest neighbor can again be taken into consideration. With this simple extension, it is possible to synthesize isotropic 3D texture volumes. As mentioned, volumetric textures have the advantage that a mapping step to an object surface is not necessary. Figure 3.19 depicts a synthesized volume texture, which is mapped to the geometry of a uterine cavity. Nevertheless, computation times become quite high, even for smaller textures, thus making an interactive processing infeasible. Moreover, to obtain sufficient surface detail, high volume resolution is required. Due to texture memory limitations this cannot be achieved if several different textures are needed.

The described methods allow synthesis of organ textures of high quality. Nevertheless, a number of limitations were encountered. Using only highly homogeneous source images results in output images acceptable for highly realistic simulation. However, several organs show low frequency patterns on their surface, which cannot easily be replicated by the algorithm. Figure 3.20

(a)   (b)   (c)
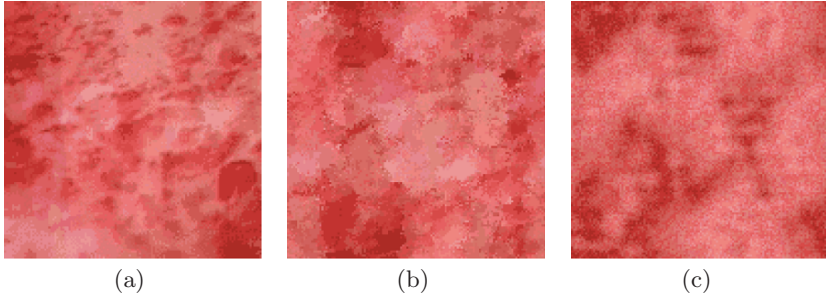
Figure 3.20. Limitations of the synthesis models. (a) Drift-corrected, quantized uterus endometrium sample, (b) Synthesized with co-occurrence model, (c) Synthesized with autocorrelation model

depicts an example with synthesis outcome of insufficient quality for a sample with unhomogeneous patterns. Moreover, even with the optimized approaches, the computation time is still unfavorable. Finally, some iterative, and unfortunately also nonintuitive, parameter tuning is necessary to improve the outcome of the synthesis. With medical experts as the end users of the scene generation tool, this is not acceptable.

3.6.4 Texture Generation for Hysteroscopic Simulation

In order to be able to synthesize a wider range of textures, including inhomogeneities and anisotropies at low and high frequencies, a different approach has been integrated into the texture generation framework based on Markov random fields. A fast, multiscale nonparametric synthesis approach is followed, as outlined in [204], which captures sufficient higher-order statistical characteristics.

General MRF Model

In order to define the MRF, the pixels of an image $\mathbf{x}$ with dimension $M_x \times M_x$ are considered to be at sites s on a lattice $S = \{(i,j) \,|\, 0 \le i,j < M_x\}$. Each site represents a variable X_s which is equal to a value x_s within the state space Λ. For grey value or color indexed images, the state space is defined as $\Lambda = \{0, 1, 2, \ldots, L-1\}$, where L is the number of grey levels or colors. The configuration space for the sets of variables $\mathbf{X} = \{X_s \,|\, s \in S\}$ is the set of all possible images $\Omega = \Lambda^{M_x \times M_x}$. For a texture to be modeled as an MRF, it is assumed that the value of the pixels depends only on a limited number of local neighbors. In [24] it was proven that the joint probability measure on Ω is uniquely determined by its LCPDF with respect to a specific neighborhood system

$$P(x_s \,|\, x_r, r \neq s) = P(x_s \,|\, x_r, r \in \mathcal{N}_s) \qquad (3.37)$$

where $\mathcal{N}_s$ represents a predefined neighborhood of site s. For homogeneous MRF a symmetric neighborhood of order c for a site s can be defined as

$$\mathcal{N}_s^c = \{(k,l) \in S \,|\, 0 < (k-i)^2 + (l-j)^2 \leq c\}. \qquad (3.38)$$

For a parametric approach the LCPDF has to be estimated from the multidimensional histogram of a particular texture using Gibbs distributions. This allows us to define the probability density via appropriate parametric potential functions. The parameters have to be optimized to match the LCPDF to the histogram. Unfortunately, the sample data are sparsely dispersed over the corresponding multidimensional histogram and the true distribution is not known, which leads to results of lower accuracy when using parametric estimation. Therefore, a nonparametric approach is followed.

Nonparametric MRF Model

Given a texture sample $\mathbf{y} \in \Omega$ and a neighborhood system $\mathcal{N} = \{N_s \,|\, s \in S_y\}$, the first step of the nonparametric MRF is to determine the multidimensional histogram of the image. To this end, the frequency of occurance of a specific combination of pixel values L for all sites p using neighborhood $\mathcal{N}_p$ is obtained.

$$F(L_0, \ldots, L_{|\mathcal{N}_p|}) = \sum_{p \in S_y, \mathcal{N}_p \subset S_y} \delta(y_p - L_0) \prod_{r \in \mathcal{N}_p} \delta(y_r - L_{n_r}) \qquad (3.39)$$

where n_r are indices for the sites r in neighborhood $\mathcal{N}_p$. By determining F for all combinations of $L_0, \ldots, L_{|\mathcal{N}_p|} \in \Lambda$, the multidimensional histogram can be built. The total number of dimensions is the statistical order of the model, which is equal to the neighborhood size $N = |\mathcal{N}_p|$. The LCPDF can now be calculated according to

$$P(y_s \,|\, y_r, r \in \mathcal{N}_s) = \frac{F(y_s, y_r, r \in \mathcal{N}_s)}{\sum_{\tilde{L} \in \Lambda} F(\tilde{L}, x_r, r \in \mathcal{N}_s)}. \qquad (3.40)$$

It should be noted that the multidimensional histogram is usually only sparsely filled. If a 3×3 neighborhood were used with 32 grey levels, then the histogram would contain $32^9 \approx 3.51 \times 10^{13}$ bins. Even a large sample image with a resolution of 1024×1024 pixels would only fill a minute fraction of the histogram space. This situation can be alleviated by smoothing the distribution over the histogram. To this end a nonparametric density estimator can be applied. The Parzen-window estimator spreads each sample datum over a larger area in the histogram. Denoting $\mathbf{Z}_p = [y_p, y_q, q \in \mathcal{N}_p]$ and $\mathbf{z} = [x_s, x_r, r \in \mathcal{N}_s]$, the true density function can be approximated with

(a)   (b)   (c)
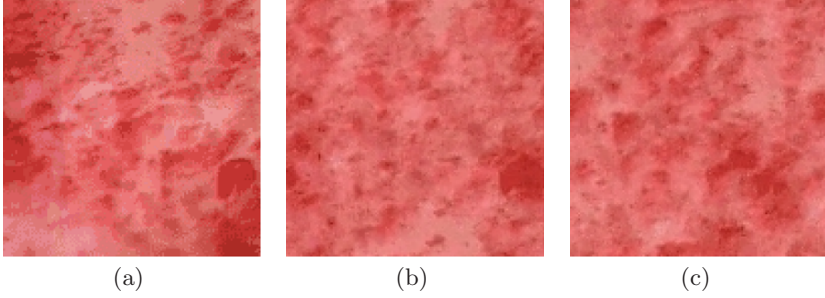
Figure 3.21. Synthesis examples with the nonparametric MRF model (3x3 neighborhood). (a) Drift-corrected, quantized uterus endometrium sample, (b) Synthesized texture example 1, (c) Synthesized texture example 2

$$\hat{F} = \frac{1}{nh^d} \sum_{p \in S_y, N_p \subset S_y} K\left\{\frac{1}{h}(\mathbf{z} - \mathbf{Z}_p)\right\} \tag{3.41}$$

where n is the number of sites $p \in S_y$, and $d = |\mathcal{N}_p| + 1$ the dimension of the histogram (i.e., the number of elements in $\mathbf{Z}_p$). The histogram is convolved with a radially symmetric Gaussian kernel.

$$K(\mathbf{z}) = \frac{1}{(4\pi^{d/2})} e^{\left(-\frac{1}{2}\mathbf{z}^T \mathbf{z}\right)} \tag{3.42}$$

The area of influence of the density estimator is controlled by the window parameter h. If it is too small, then noise will be introduced into the estimation function $\hat{f}$. If it is too large, then local detail will be blurred. According to [243], an optimal selection of the parameter can be done according to

$$h_{opt} = \sigma\left\{\frac{4}{n(2d+1)}\right\}^{1/(d+4)} \tag{3.43}$$

where σ^2 is the variance of the histogram of training image $\mathbf{y}$. The LCPDF can thus be approximated using the Parzen-window density estimator.

$$\hat{P}(x_s | x_r, r \in \mathcal{N}_s) = \frac{\sum_{p \in S_y, \mathcal{N}_p \subset S_y} e^{\left((-1/2h^2)(\mathbf{z}-\mathbf{Z}_p)^T(\mathbf{z}-\mathbf{Z}_p)\right)}}{\sum_{x_s \in \Lambda} \sum_{p \in S_y, \mathcal{N}_p \subset S_y} e^{\left((-1/2h^2)(\mathbf{z}-\mathbf{Z}_p)^T(\mathbf{z}-\mathbf{Z}_p)\right)}} \tag{3.44}$$

In order to synthesize a texture, stochastic relaxation can be applied using the estimation of the LCPDF. A popular approach for the relaxation is the *Iterative Conditional Modes* (ICM) method described in [25]. Starting with a randomly generated texture, the image sites are randomly visited and updated maximizing the marginal posterior distribution at each pixel. For small neighborhood sizes, superior results can be obtained using the nonparametric MRF model. In Figure 3.21, results from applying the method to the
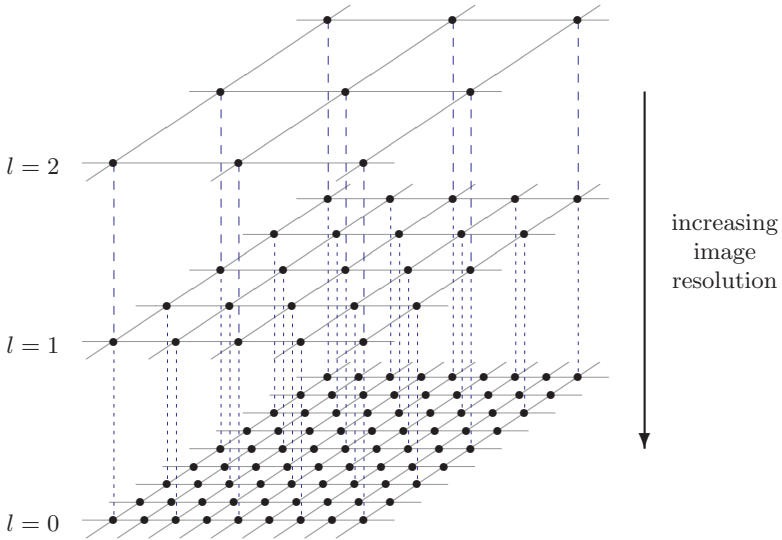
Figure 3.22. Decimation between grid levels in multiscale approach

previously shown unhomogeneous texture are depicted. One problem already encountered in the previous approach is the propagation of low-frequency, global image characteristics. These are typically propagated across the image lattice only by local interactions, and the relaxation can get stuck in local minima. To tackle this difficulty, a multiresolution technique can be followed.

Multiscale Texture Synthesis

The underlying idea is to synthesize coarse structures at lower resolution, and successively add more detail as the resolution increases. In this scheme, the outcome of the relaxation at one level constrains the relaxation at the following one. In addition to giving higher synthesis quality, this also reduces the number of iterations necessary to reach equilibrium. The multiscale representation is defined by images $\mathbf{X}^l$ for each level l on lattices given by:

$$S^l = \{(2^l i, 2^l j) \,|\, 0 \leq i, j < N/2^l\} \tag{3.45}$$

where S^0 is the highest resolution image. The lattice S^{l+1} is obtained from S^l by decimation. The multigrid representation is depicted in Figure 3.22. Moreover, the neighborhood relationship for each level also has to be redefined:

$$\mathcal{N}_s^c(l) = \left\{(2^l m, 2^l n) \in S^l \,|\, 0 < (i-m)^2 + (j-n)^2 \leq c\right\}. \tag{3.46}$$

Starting from the lowest resolution, at each grid level stochastic relaxation is carried out until equilibrium is reached. This requires that decimations $\mathbf{Y}^l$

(a)   (b)   (c)

(d)   (e)   (f)
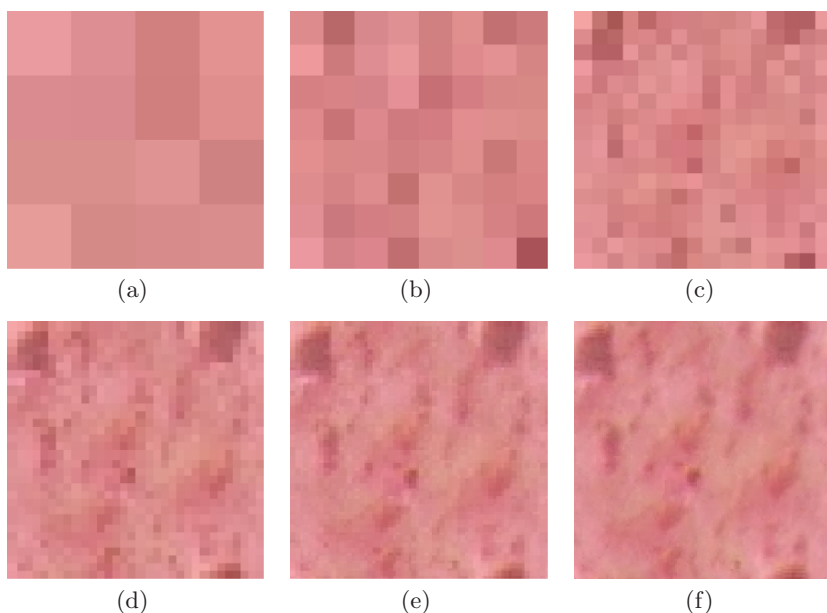
Figure 3.23. Multiscale texture synthesis. (a) Level 5 (4×4), (b) Level 4 (8×8), (c) Level 3 (16×16), (d) Level 2 (32×32), (e) Level 1 (64×64), (f) Level 0 (128×128)

Level $n + 1$

○ No confidence

● Full confidence
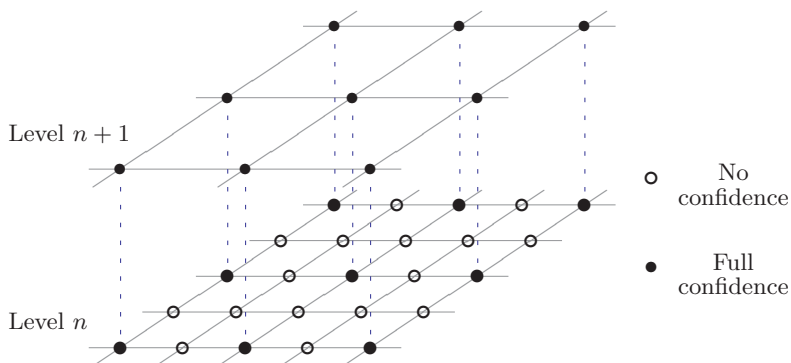
Level n

Figure 3.24. Propagation of pixel temperature between levels

of the training image are also obtained for all levels to construct the LCPDF. The individual equilibria of the multiscale approach during a synthesis process are shown in Figure 3.23.

Algorithm Enhancements

An extension of the synthesis approach can be made by including a *pixel temperature function* into the process. As with stochastic annealing, a local temperature function is integrated into the relaxation process. Each pixel is assigned a temperature with $0 \leq t_s \leq 1$, which represents a degree of confidence in the current value, where 0 denotes no, and 1 full confidence. The temperature is integrated into the Parzen-window estimation of Equation 3.41 by weighting the difference between the sample and output neighborhood values:

$$(\mathbf{z} - \mathbf{Z}_p) = \left[x_s - y_p, (x_r - y_r)(1 - t_r), r \in \mathcal{N}_s \right] . \tag{3.47}$$

After the relaxation process at level $l + 1$, all pixels propagated to level l receive complete confidence, while the rest have no confidence assigned (see Figure 3.24). Every time a pixel value is changed in the following relaxation on the new level, its temperature is adjusted according to

$$t_s = max \left\{ 0, \frac{1 + \sum_{r \in \mathcal{N}_s} t_r}{|\mathcal{N}_s|} \right\} . \tag{3.48}$$

At the start of the relaxation, only the values propagated from the previous level are used in the LCPDF. However, as the process advances, other sites gain more confidence, and affect the computation. Additionally, the confidence level can be used as a relaxation termination criteria, i.e., when all sites have reached full confidence, the relaxation can move on to the next level.

A further modification of the algorithm can be done by using a specialized *neighborhood searching scheme*. As previously discussed, an estimate of the LCPDF is obtained based on a sum of Gaussian kernels. Since the data are sparsely distributed, it can be assumed that the site giving the smallest distance between vectors $\mathbf{Z}_p$ and $\mathbf{z}$ will have the largest influence. Thus, the LCPDF can be approximated by just searching for the site with minimal distance. The sample subset is selected as all pixels with a neighbor of the same color as the respective neighbor of the output pixel being iterated. This is similar to the method proposed in [8]; however the subset is larger, resulting in a higher synthesis quality. Examples of the complete synthesis process can be seen in Figure 3.25.

3.7 Additional Texture Detail

With the nonparametric MRF synthesis strategy it is possible to synthesize textures showing relatively irregular patterns. However, some organ surfaces exist which need a specialized treatment, since not all their texture characteristics can be easily replicated. This includes, for instance, follicle patterns

(a)	(b)	(c)	(d)
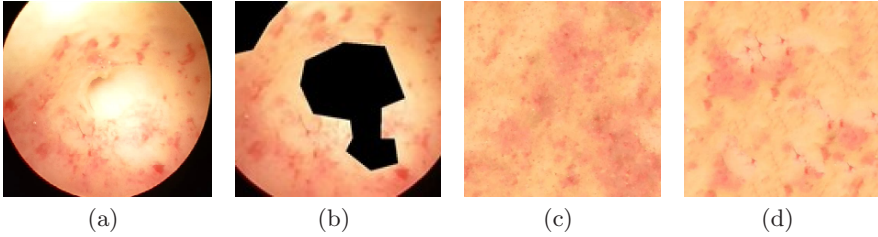
Figure 3.25. Fast MRF-based synthesis examples of in vivo texture (from [203]). (a) In vivo image, (b) Cropped and masked sample, (c) Synthesized texture 1, (d) Synthesized texture 2

on ovaries or vessel trees on some organ surfaces. Thus, an approach is required to further augment generated textures with macroscopic structural patterns. This can be done by procedural postprocessing of synthesized textures. The main idea is to optimally combine separate texture patterns, which necessitates the definition of texture transfer functions. As two examples, the enhancement of liver textures with low-frequency specks and ovary textures with follicle patterns will be discussed.

3.7.1 Embedding of Specks into Liver Textures

After generating a homogeneous base texture for the liver, it can be further enhanced by adding characteristic specks. Since these specks are usually too small to use a texture synthesis strategy, their appearance is generated using the basic procedural approach. To make the pattern more granular, the approach used for generating fat texture can be modified by steepening the edges of the noise signal with a transfer function defined as

$$\hat{T}_i = \frac{1}{2}\left(1 + \frac{\arctan\left((2s+1)^4(2T_i - 1)\right)}{\arctan(2s+1)^4}\right) \tag{3.49}$$

where T_i are the unmodified texture values, and s is a scaling factor ($0 \leq s \leq 1$). In order to merge the textures, a third, *mask texture* has to be created, which defines the smooth transfer between the base and pattern texture. For the specks, this mask texture can be automatically generated. To this end, spheres with varying radii are randomly distributed in the texture space. Moreover, to approximate the actual shapes of the specks, jitter, controlled by a vectorial noise function, is added. Finally, the textures are blended according to the mask texture. An example of this process is visualized in Figure 3.26. It should be noted that adding smaller detail like the specks is necessary only when dealing with relatively homogeneous base textures. More sophisticated algorithms, like the described MRF approach, are usually capable of automatically synthesizing these patterns.

(a)	(b)	(c)	(d)
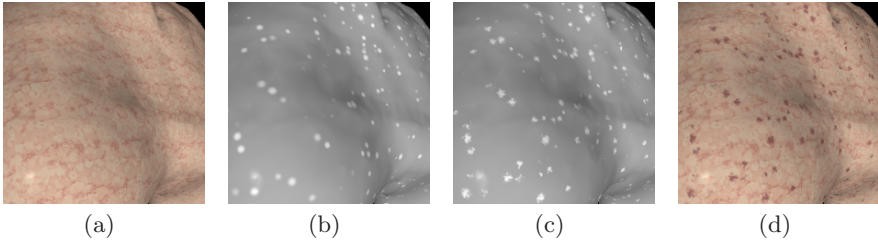
Figure 3.26. Texture enhancement using blending. (a) Original textured surface, (b) Automatically generated, randomized spheres, (c) Jittered spheres according to noise function, (d) Blended 3D textures

3.7.2 Overlay of Real Follicle Textures

In some cases the embedding of additional texture is more complicated. This is for instance true if the texture has to coincide with the actual geometrical variation of an organ surface. An example of this is the follicles which can be found on the surface of ovaries. A straightforward approach would be to let the user mark those areas where different texture components should be blended. For the blob-shaped follicles, this can be easily done. Figure 3.27 depicts the addition of follicle textures onto an organ surface covered with a synthesized homogeneous texture. In this special case, the follicle texture was not synthesized, but cropped from the original image. While this basic approach leads to acceptable results, it is not appropriate for the scene generation process, since it involves too much manual tuning and image processing.

3.8 Texture Mapping

In the previous sections the generation of textures for surgical scenes has been discussed. What remains to be done is the mapping of those textures to object geometries. If the texture is stored in 3D-space as a trivariate scalar or vector-valued function of indexed or RGB colors respectively, then the surface mapping is trivial, since the texture values can be directly read at the vertex coordinates. No seams or distortions are introduced with this approach. However, 3D texturing is only viable for homogeneous, isotropic patterns. Moreover, the resolution is rather small due to the currently still limited texture memory. Also, the synthesis of 3D texture blocks takes considerably longer than that of their 2D counterparts, thus hampering interactivity of scene generation. If the texture is built as a bivariate field, then a bijective mapping function has to be determined, relating 3D points on a mesh surface to the parametric 2D texture space. This step can usually not be carried out without introducing errors.

A mapping M between two surfaces S_1 and S_2 is called isometric, if the geodesic distance d_{geo} between arbitrary points is maintained:

(a)   (b)   (c)   (d)
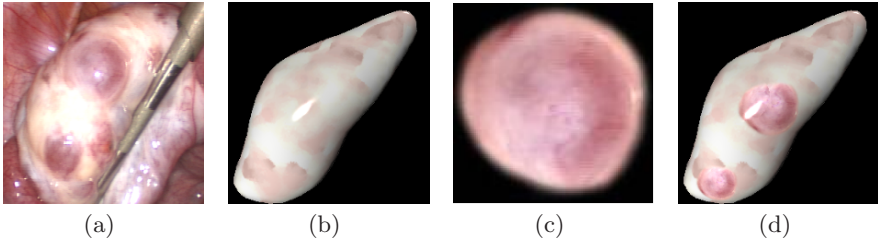
Figure 3.27. Addition of special texture patterns. (a) Follicles on real ovary, (b) Virtual ovary covered with base texture, (c) Cropped follicle pattern, (d) Follicle pattern embedded into base texture

$$\forall \mathbf{p}_i, \mathbf{p}_j \in S_1 : d_{geo}(\mathbf{p}_i, \mathbf{p}_j) = d_{geo}\big(M(\mathbf{p}_i), M(\mathbf{p}_j)\big) . \tag{3.50}$$

In this case, the surface S_1 is called developable. Gaussian curvature at a point $\mathbf{p}$ on surface S is defined as

$$K = \frac{1}{R_1 R_2} \tag{3.51}$$

where R_1 and R_2 are the largest and the smallest principle radii of curvature of the respective osculating circles. A sphere has, for instance, a constant positive Gaussian curvature, while that of a cylinder is everywhere zero. Carl Friedrich Gauss stated with his *Theorema Egregium* that an isometric mapping of surfaces with unequal Gaussian curvatures K is not possible [96]. Thus distortions are inevitable when mapping from 2D texture patches with zero curvature to 3D surfaces with non-zero curvature. So a method has to be found to minimize these distortions.

It should be noted that to avoid this problem, one could also directly synthesize textures on the 3D surfaces. Methods to extend the approach discussed in Section 3.6.3 have been presented in [173]. The authors propose techniques to locally project the 3D surface to a tangent plane, which is only successful when the local curvature is not too high. Also, MRF-based synthesis directly on surfaces has been suggested for homogeneous patterns [271, 291, 303], as well as textures with local variations [306]. These approaches require the warping of the spatial neighborhood function over the surface to approximate the mesh curvature. Therefore, generally only relatively small texture structures can be reliably applied to a 3D surface with these methods. Moreover, user interaction is needed to define a vector field over the surface in order to provide a consistent orientation for the texture pattern. Finally, these types of methods do not allow interactive transformations of texture over the 3D surface.

For completeness, it also has to be mentioned that the previously discussed patch-based texture generation strategies can also be directly applied to surfaces. The concept of directly pasting patches onto meshes was initially

introduced in [219]. Refined versions targeting real-time synthesis were later reported in [248, 176]. As in the 2D process, texture discontinuities along seams can appear, which have to be minimized with boundary matching or blending strategies. A drawback also of these approaches is the limited interactivity, due to the necessary recalculations of texture patches.

While direct texture generation on surfaces is a promising alternative, we have decided to separate the synthesis and surface mapping steps. One reason is the reuse of generated textures, which are stored in a database. A direct mapping strategy would require a resynthesis of textures for each new mesh. Nevertheless, some geometric information of the object meshes still has to be taken into account in the texturing process, as previously exemplified with the mapping of follicle patterns. In the following, the approach followed in the scene generation process will be discussed. It contains two major steps aiming at distortion minimization; namely, mesh cutting along seams and optimal parameterization.

3.8.1 Mesh Cutting

For texture mapping, the object meshes created in the previous chapter have to be parameterized (an extensive overview of current techniques can be found in [86]). Our meshes are represented as piecewise linear triangular surfaces $S_{\mathcal{T}}$, defined by a set of triangles $\mathcal{T} = \{T_1, \ldots, T_N\}$. For the parameterization, a piecewise linear mapping $f : S_{\mathcal{T}} \rightarrow S^*$ of the surface $S_{\mathcal{T}} \in I\!\!R^3$ into the planar domain $S^* \in I\!\!R^2$ has to be found. As discussed above, distortions are usually introduced in this step. A mapping can either be conformal, i.e., angle-preserving; or equiareal, i.e., area-preserving. An isometric mapping is conformal and equiareal; however, since for our meshes such a mapping cannot be found, we minimize a combination of angle and area distortion.

A surface can be mapped to a plane only if it is homomorphic to a disk. Thus a number of cases can be encountered where a surface has to be cut before the mapping step. The genus G of a closed surface is defined by the Euler-Poincare formula:

$$G = \frac{1}{2}E - V - F - B + 2 \tag{3.52}$$

where V, E, and F are the number of mesh vertices, edges, and faces, respectively, and B the number of boundary loops. The genus of a surface can be regarded as the number of holes or handles in a 2-manifold mesh. Surfaces with non-zero genus are not homomorphic to a disk, and thus have to be cut to reduce the genus. The same is true for a manifold mesh with a boundary. If interior holes are present, then these surfaces also have to be separated. Apart from this, the introduction of additional seams by the subdivision of an already disk-like mesh can be applied to further reduce distortion. For the scene generation process, the mesh cutting can be performed according to [235]. Surfaces are separated along existing edges according to two quality
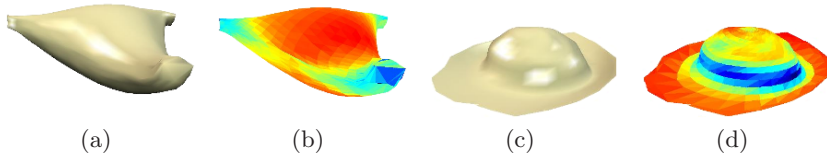
Figure 3.28. Relative visibility of triangle mesh faces (low=blue, high=red). (a) Uterine cavity surface mesh, (b) Inside visibility measure, (c) Myoma surface mesh, (d) Outside visibility measure

metrics. Firstly, a visibility measure for mesh edges is calculated by rendering the scene from different viewpoints and marking the visible elements. Secondly, the distortion of mesh nodes is determined by estimating the Gaussian curvature. The 3D surface is then separated along its nodes and seams with high distortion, but minimal visibility.

Visibility Measure

In order to obtain edge visibility $V(e)$, a graphics hardware-based approach can be followed. The mesh is rendered orthographically either from multiple viewing positions uniformly distributed on a bounding half-sphere, or into multiple viewing directions from inside an object. This differentiation has to be made, since the pathology meshes will be visible from the outside, whereas the cavity surface will be seen from the inside. By assigning a unique color to each polygon, the visibility of a face can be determined by the presence of these colors in the rendered image. The edge visibility can then be obtained by averaging the visibility of the adjacent faces. The relative visibility of the faces of two example meshes are shown in Figure 3.28. It can be seen that the tubal orifices in the uterine cavity have a low visibility from the inside. Similarly, the visibility of the myoma stem region is reduced, since it is partly covered by the surrounding cavity submesh.

Distortion Measure

In the next step, the distortion at the mesh vertices has to be determined. This can be computed according to an approximation of the local Gaussian curvature. To this end a spherical region around a specific vertex $\mathbf{v}_i$ is considered. The radius of the former is defined according to:

$$R_i = k \max_{\mathbf{v}_j \in \mathcal{N}_i} \|\mathbf{v}_i - \mathbf{v}_j\| \tag{3.53}$$

where $\mathcal{N}_i$ is the set of nodes adjacent to $\mathbf{v}_i$, and k is a scaling factor. Based on this, we define a local subset of triangles:

(a)   (b)   (c)   (d)
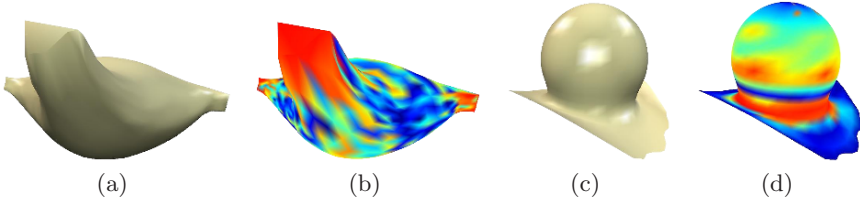
Figure 3.29. Relative distortion at triangle mesh vertices (low=blue, high=red).
(a) Uterine cavity surface mesh, (b) Distortion measure, (c) Smooth myoma mesh,
(d) Distortion measure

$$\tilde{M}_k(\mathbf{v}_i) = \begin{cases} T_j & | \quad \forall \mathbf{v}_j \in T_j : \|\mathbf{v}_i - \mathbf{v}_j\| < R_i, & \text{if} \quad k > 0 \\ T_j & | \quad \mathbf{v}_i \in T_j, & \text{if} \quad k = 0 \,. \end{cases} \qquad (3.54)$$

Thus the subset is composed of all triangles falling completely into the spherical region. The boundary Γ_k of the local subset $M_k(\mathbf{v}_i)$ is piecewise linear. By considering a new set $\tilde{\mathcal{N}}_l$ of triangles $\hat{T}_j = \{\mathbf{v}_i, \mathbf{v}^i_{\Gamma_k}, \mathbf{v}^{i\oplus 1}_{\Gamma_k}\}$, which are formed by the center vertex and two consecutive vertices on the boundary, we can define the distortion measure:

$$\Psi'_k(\mathbf{v}_i) = 1 - \frac{\sum_{|\tilde{\mathcal{N}}_l|} \alpha_l}{2\pi} \qquad (3.55)$$

where α_l are the angles at vertex $\mathbf{v}_i$ in the new triangles. In order to accommodate for different mesh resolutions, k has to be varied in certain bounds. Unfortunately, the distortion metric is quite sensitive to these variations. Therefore, it is more appropriate to determine the distortions according to

$$\Psi_k(\mathbf{v}_i) = \max_{0 \leq q \leq k} \Psi'_q(\mathbf{v}_i) \,. \qquad (3.56)$$

It should be noted that at saddle points the metric can become negative. Moreover, if the sum of the angles at a mesh boundary vertex is less than 2π, then it is locally developable and the distortion is zero. Figure 3.29 depicts two examples of the measure. The models are colored according to metric values at the vertices. High distortion is present due to the local mesh curvature at the orifices of the uterine cavity, or at the stem of the myoma.

Optimal Seam Selection

Based on the metrics defined above, seams can now be laid in order to reduce overall mesh distortion. An optimal cut path along the mesh edges has to be found, which minimizes distortion and seam visibility. To this end, a solution to a *Prize-collecting Steiner tree* has to be found. With $G = (V, E)$ being an undirected graph with associated vertex- and edge-weights, the Steiner

(a)   (b)   (c)
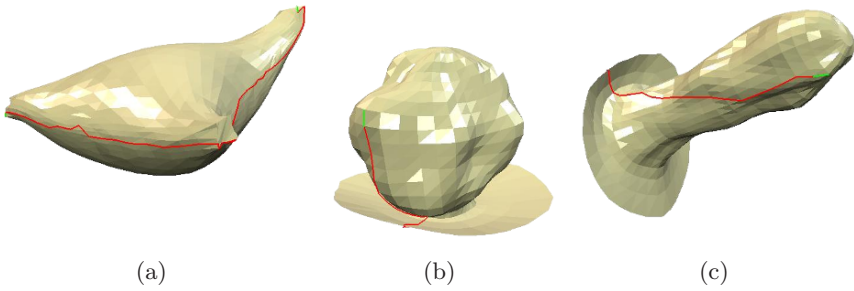
Figure 3.30. Seams shown in red on different meshes, connecting high distortion vertices. (a) Seam on uterine cavity mesh, (b) Seam on myoma mesh, (c) Seam on polyp mesh

problem consists in finding a connected subgraph $T = (V_T, E_T) \subset G, V_T \subseteq V, E_T \subseteq E$, that minimizes the objective function

$$c(T) = \sum_{v \notin V_T} \Psi(v) + \sum_{e \in E_T} \left(V(e) \cdot |e| \right). \tag{3.57}$$

This problem is known to be NP-complete. An approximation of the solution is obtained following a two-stage process. First, vertices V_T with distortion above a defined threshold – the *terminals* – are selected. The threshold can for instance be set to a percentage of the overall mesh distortion or be user-specified. In order to force the seams to go through less visible areas, the selection can be influenced by considering distortion scaled with vertex visibility $\Psi(v)/V(v)$. In the next step, a Steiner tree connecting the terminals is determined.

A heuristic algorithm using front propagation is used to approximate the result within polynomial time. Starting at each terminal of V_T, a front is grown along the edge with minimal cost. If two fronts meet, their paths are combined, and the resulting subtree is added to the approximate Steiner tree. This is continued until all fronts have merged. The edge cost defined in Equation 3.57 can also be extended by introducing an enhanced edge distortion measure:

$$\sum_{e \in E_T} \left((V(e) \cdot |e|)(1 - \Psi(e)) \right). \tag{3.58}$$

This results in more cuts being placed along edges of high curvature, where texture artifacts are less visible. Figure 3.30 depicts seams obtained for different meshes. Only a small number of vertices with high distortion were selected as terminals.

3.8.2 Mesh Parameterization

After mesh distortion – as well as possibly mesh genus – have been reduced by introducing additional seams, the actual parameterization step can be carried out, i.e., the mesh is flattened into the 2D plane.

Numerous methods for this step have been suggested in the past. One of the first, described in [272], used graph theory, providing a barycentric mapping theorem for embedding a planar graph. Forces on the mesh vertices are minimized by placing each vertex at the barycenter of its neighbors. This approach was later extended in [85], where shape-preserving weights were introduced. This avoids triangle flips in the presence of obtuse angles, guaranteeing bijective mapping for fixed, convex boundaries. An approach using a discrete approximation of continuous harmonic maps is discussed in [78]. The authors generate parameterizations while minimizing angular distortion with the discrete maps. However, in some cases inverted elements can be generated. In [227] the parameterization problem is considered as an optimal mapping of a signal to a surface. A nonlinear stretch metric derived from Taylor expansions of the signal error is used to reduce distortion. Moreover, to speed up the process, a hierarchical approach is applied. All these methods require a predefined fixed, convex boundary in the parameterization domain. Thus, the boundary mapping has to be specified in advance, preferably to a convex polygon. This also denotes selecting boundary shape and node distribution.

Unfortunately, fixed boundary methods typically introduce additional distortions. This can be avoided with free boundary methods. An additional advantage of these techniques is the usually reduced number of seams required in the mesh. An approach not requiring fixed boundaries has been discussed in [121]. The authors use a *most isometric parameterizations* (MIPS) method, minimizing a nonlinear deformation functional of the first fundamental form of the mapping. However, their method is limited to rather simple cases. In [158], the *least squares conformal maps* (LSCM) technique is proposed, which also uses an adaptive boundary. An error metric based on the discretization of Cauchy-Riemann equations is calculated for constructing free-boundary maps. Unfortunately, bijective mapping is not guaranteed, since folded triangles may appear. Another technique performing free-boundary parameterizationis *angle-based flattening* (ABF) [236]. A functional is minimized according to differences between 2D and 3D mesh angles. This robust and efficient method generates provable conformal mappings with low stretch and no flipped triangles. It is well-suited for the mesh parameterization in the scene generation process and will be discussed in detail below.

Angle-Based Flattening

The angle-based parameterization approach is based on the observation that a planar triangular mesh is defined by the angles within each triangle, except for global scaling, rotation, and translation. An optimization problem is defined,

where a set of constraints guarantees the validity of the flattened mesh. The objective function to be minimized is given by

$$f(\alpha) = \sum_{t \in T} \sum_{k=1}^{3} \frac{1}{w_k^t} (\alpha_k^t - \hat{\alpha}_k^t)^2 \qquad (3.59)$$

where α_k^t are the three unknown planar angles of triangle t in counterclockwise order; and w_k^t are weights, which are usually set to $(\hat{\alpha}_k^t)^{-2}$ to consider relative distortion. The optimal angles $\hat{\alpha}_k^t$ are derived from the 3D mesh angles β_k^t according to

$$\hat{\alpha}_k^t = \begin{cases} 2\pi \cdot \beta_k^t \cdot (\sum_l \tilde{\beta}_l^t)^{-1} & | \quad \mathbf{v}_k^t \in V \backslash V_\Gamma \\ \beta_k^t & | \quad \mathbf{v}_k^t \in V_\Gamma \end{cases} \qquad (3.60)$$

where $\mathbf{v}_k^t$ is the vertex at angle k of triangle t, V_Γ is the subset of boundary vertices, and $\tilde{\beta}_l^t$ are all angles around vertex $\mathbf{v}_k^t$. The unknown angles have to fulfill a number of constraints. Each triangle $t \in T$ has to be valid, thus requiring

$$g_1(t) = \sum_{i=1}^{3} \alpha_i^t - \pi \overset{!}{=} 0 . \qquad (3.61)$$

For interior nodes $\mathbf{v} \in V \backslash V_\Gamma$ mesh planarity has to be ensured.

$$g_2(\mathbf{v}) = \sum_{S_\mathbf{v}} \alpha_l^t - 2\pi \overset{!}{=} 0 \qquad (3.62)$$

where $S_\mathbf{v}$ is the set of angles adjacent to vertex $\mathbf{v}$. Finally, edges shared by two triangles should be of equal length. Thus, for each internal vertex

$$g_3^*(\mathbf{v}) = \prod_{S_\mathbf{v}} \frac{\sin(\alpha_{k \oplus 1}^t)}{\sin(\alpha_{k \ominus 1}^t)} \overset{!}{=} 1 \qquad (3.63)$$

where $k \oplus 1$ and $k \ominus 1$ denote in the current triangle the index of the next and previous angle from the one at $\mathbf{v}$, respectively. As indicated in [305], this equation can be rewritten by applying the log function, in order to obtain a diagonal Hessian matrix in the Lagrangian optimization step.

$$g_3(\mathbf{v}) = \sum_{S_\mathbf{v}} \log\left(\sin(\alpha_{k \oplus 1}^t)\right) - \log\left(\sin(\alpha_{k \ominus 1}^t)\right) \overset{!}{=} 0 . \qquad (3.64)$$

The constrained minimization problem can be solved using Lagrangian multipliers, giving the modified objective function

$$\begin{aligned} L(x) &= L(\alpha, \lambda_1, \lambda_2, \lambda_3) \\ &= f(\alpha) + \sum_t \lambda_1^t g_1(t) + \sum_\mathbf{v} \lambda_2^\mathbf{v} g_2(\mathbf{v}) + \sum_\mathbf{v} \lambda_3^\mathbf{v} g_3(\mathbf{v}) . \end{aligned} \qquad (3.65)$$

(a) (b) (c) (d)
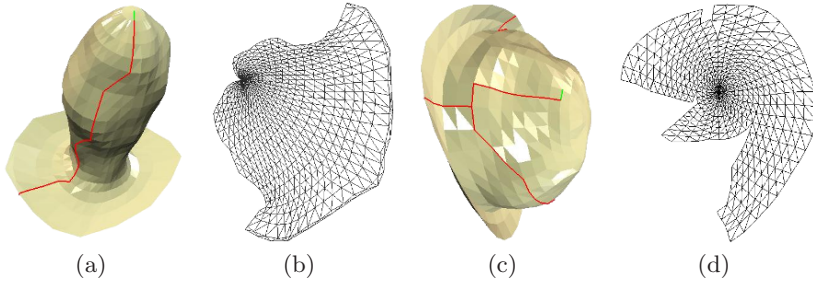
Figure 3.31. Parameterization by angle-based flattening. (a) Polyp mesh with cut seams, (b) Flattened mesh, (c) Myoma mesh with cut seams, (d) Flattened mesh

Inequality constraints can be reformulated as equality constraints using an active set strategy. Thus, a solution is obtained with a standard Newton iteration

$$x_{n+1} = x_n - \nabla^2 L^{-1}(x_n) \cdot \nabla L(x_n) \, . \tag{3.66}$$

For an initial guess the optimal angles $\hat{\alpha}_k^t$ can be selected. Due to the discussed matrix variation, the Hessian is diagonal, but can become ill-conditioned. Using a line search algorithm and a backtracking scheme alleviates this problem.

2D Mesh Retrieval

After solving for the 2D planar angles, the actual vertices need to be reconstructed in the plane. Using a front propagation method, visiting one vertex at a time can quickly become unstable from a lack of numerical precision. Instead, the conversion problem can be formulated as a global linear system, thus enabling the simultaneous computation of the 2D vertices. Given a triangle $t = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$, then the edge $\overline{\mathbf{p}_1\mathbf{p}_3}$ is given by:

$$\overline{\mathbf{p}_1\mathbf{p}_3} = M^t \cdot \overline{\mathbf{p}_1\mathbf{p}_2} = \frac{\sin(\alpha_2^t)}{\sin(\alpha_3^t)} \begin{pmatrix} \cos(\alpha_1^t) & \sin(\alpha_1^t) \\ -\sin(\alpha_1^t) & \cos(\alpha_1^t) \end{pmatrix} \overline{\mathbf{p}_1\mathbf{p}_2} \tag{3.67}$$

where α_i^t is the angle at vertex i of triangle t. Thus, the third vertex is uniquely defined with respect to the position of the two other vertices and the angles.So to obtain the positions, a solution in the least-squares sense has to be found for:

$$\min_{\mathbf{p}_i} \sum_{t=(\mathbf{p}_i, \mathbf{p}_{i+1}, \mathbf{p}_{i+2}) \in T} \| M^t(\mathbf{p}_i - \mathbf{p}_{i+1}) + \mathbf{p}_{i+2} - \mathbf{p}_{i+1} \|^2 \, . \tag{3.68}$$

| | | | |
| (a) | (b) | (c) | (d) |
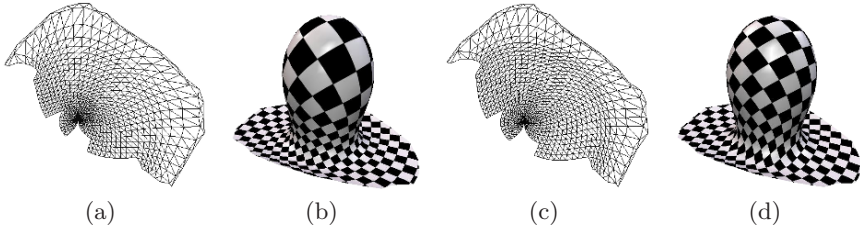
Figure 3.32. Adapted parameterization to reduce linear distortions. (a) Flattened mesh after standard ABF, (b) Distorted checkerboard texture, (c) Improved parameterization, (d) Enhanced length preservation

It should be noted that the mesh is defined except for translation, rotation, and scaling. To eliminate these degrees of freedom, two vertices sharing an edge can be fixed, e.g., by setting P_0 and P_1 to $(0,0)$ and $(1,0)$. This minimization problem is well defined and has a unique minimum. The system can for instance be solved with a SuperLU direct solver, thus finally providing the parameterization for the 3D mesh, which can be used for texturing. Examples of the outcome of the angle-based flattening procedure are visualized in Figure 3.31.

Reduction of Length Distortion

In the process carried out so far, a parameterization minimizing angular distortion has been obtained. However, due to the angle-based formulation, edge lengths are not preserved, thus introducing linear distortions. The mesh smoothing algorithm suggested in [234] can be applied to alleviate this situation.

The first step is to define a linear approximate distortion function ρ based on the erroneous edge lengths. With average distortion $\tilde{\rho}_i$ at mesh vertex i, the distortion at an arbitrary point $\mathbf{x}$ inside a triangle of the parameterization mesh with barycentric coordinates (u, v, w) can be defined as:

$$\rho(\mathbf{x}) = \tilde{\rho}_1 u + \tilde{\rho}_2 v + \tilde{\rho}_3 w \tag{3.69}$$

where the average distortion is the mean of the edge length change ratios $\rho^* = \|e_2 D\| \cdot \|e_3 D\|^{-1}$ of all adjacent edges. In order to cancel out the length distortion, the inverse mapping ρ^{-1} has to be determined. To this end, a uniform, regular grid can be adapted using Laplacian smoothing to conform to the distortion function. After obtaining the adapted mesh, the actual mapping back to the original, regular mesh approximates ρ^{-1}, and can thus be applied to the initial parameterization. The effect of this step is depicted in Figure 3.32. In order to better visualize the change, a regular pattern has been applied to the geometry. It should be noted, however, that the visual effect is less noticeable for more homogeneous, stochastic organ textures.
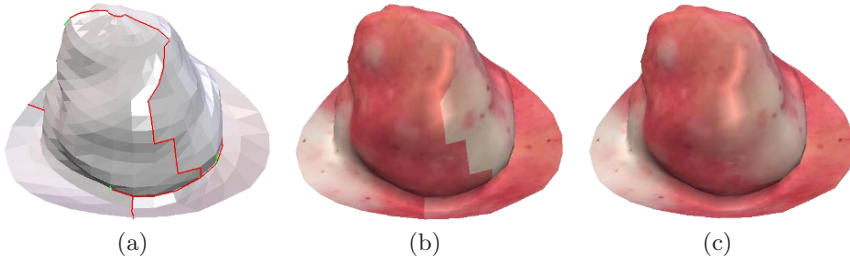
Figure 3.33. Blending strategy to avoid visual artifacts. (a) Myoma mesh with cut seams, (b) Noticeable texture discontinuities across seam, (c) Artifact removal via alpha blending

Seam Blending

Thereafter, the texture can be projected back to the 3D surface by inverse mapping. Due to the selected approach, texture discontinuities exist along the cut seams of the mesh. Although number, length, and visibility of the seams have been minimized, these discontinuities still create unwanted visual artifacts. To reduce these artifacts, hardware-based blending can be applied. Given a 2D planar mesh of a 3D surface, and a table of matching edges along the seams, duplicate triangles along the seams can be introduced. These contain the extrapolated texture coordinates from across the seam. Vertices directly on the seam are given an alpha value of 0.5, while the rest are incrementally decreased as the overlaid triangles progress away from the edge. Alpha blending can then easily be performed with standard graphics libraries. The result of this step is shown in Figure 3.33. It should be noted that this example shows an extreme case of seam discontinuity. Usually the seams are less noticeable due to more homogeneous textures and the visibility reduction.

3.9 Modelling Examples

The methods described above have been used either in the laparoscopic or the hysteroscopic simulator projects. MRF-based texture synthesis and ABF-based parameterization have been integrated into the scenario definition process for hysteroscopic training scenes. In Figure 3.34 several examples of texture mapping onto meshes relevant for laparoscopy as well as hysteroscopy are shown. In a hysteroscopy scene, a number of different tissue structures are present. Healthy anatomy as well as neoplasms exist, which usually have different visual appearances due to the varying genesis of these objects. Therefore, the texturing processes are carried out separately for the object submeshes. Since texture discontinuities similar to the ones discussed above will appear, an overlap region is included for each object mesh. By using several triangle strips around the mesh boundaries, the blending across different textures
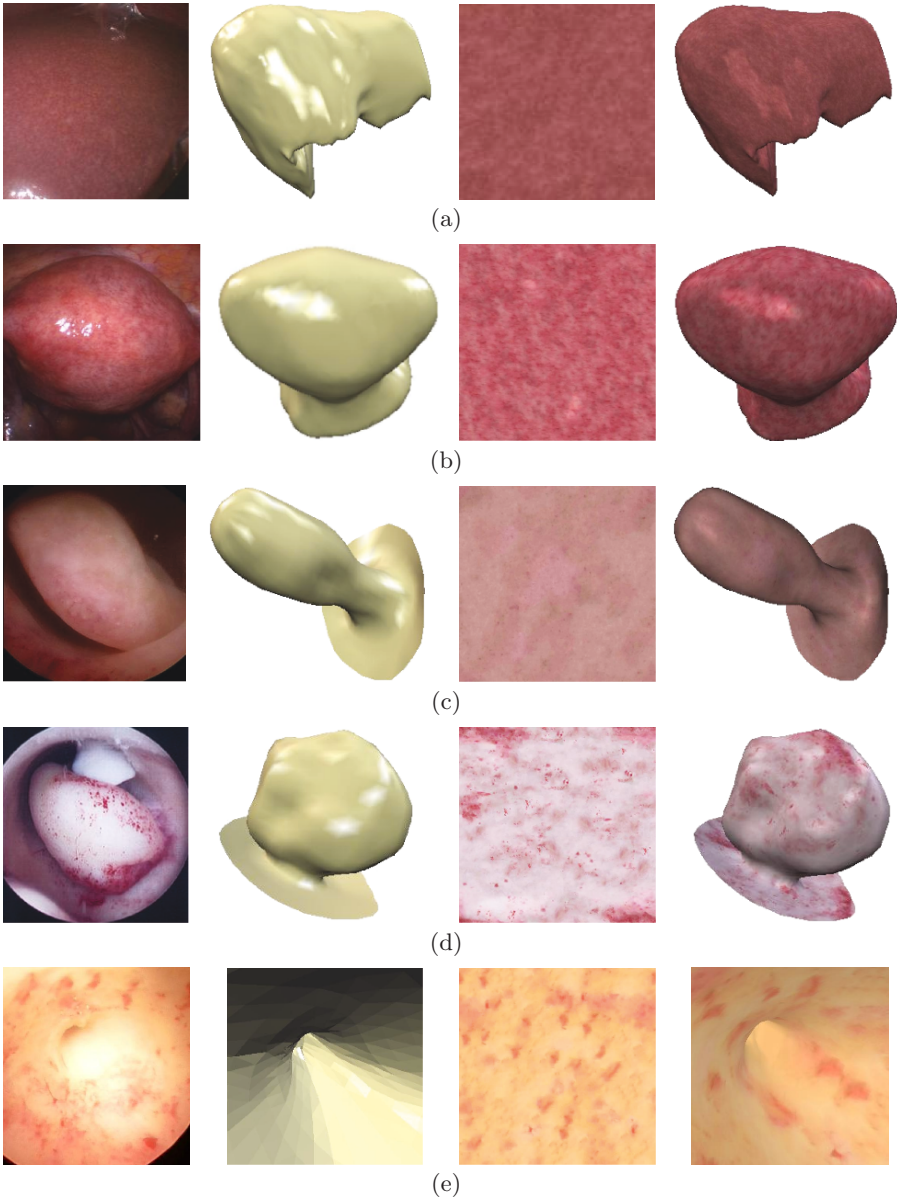
Figure 3.34. Examples of textured meshes showing source image, object mesh geometry, synthesized texture, and final result. (a) Liver, (b) Uterus, (c) Polyp, (d) Myoma, (e) Endometrium around tubal ostium

(a)

(b)

(c)

(d)
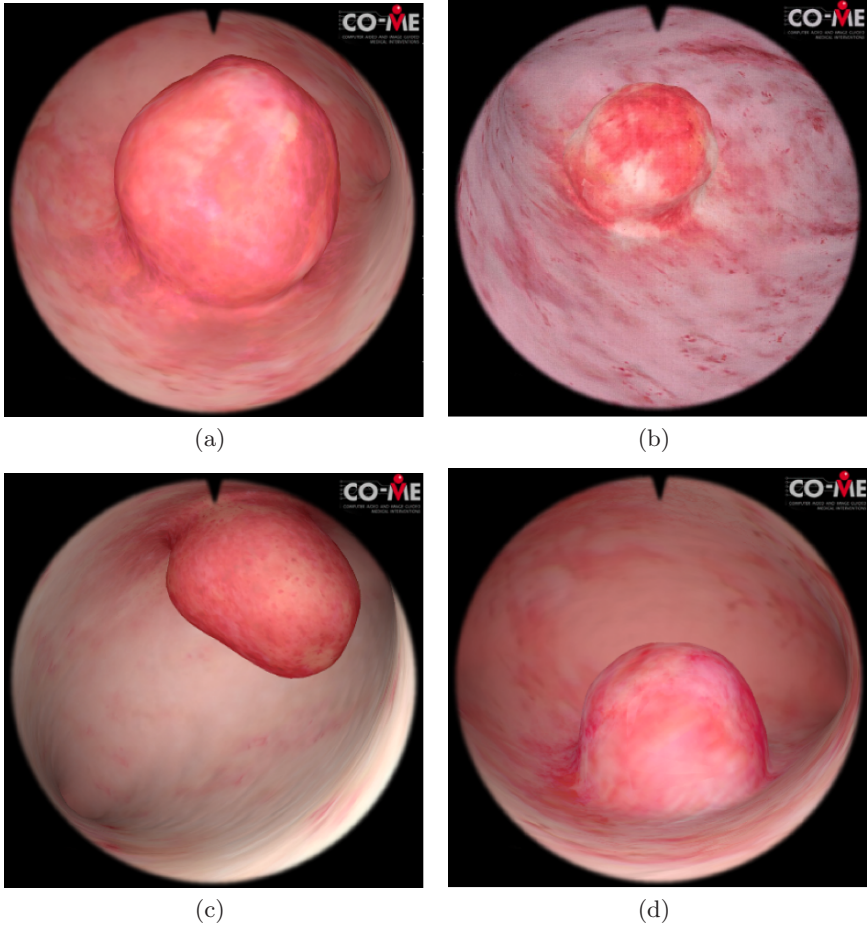
Figure 3.35. Complete hysteroscopy scene examples

can be improved. Several complete hysteroscopy scenes textured with the discussed method are presented in Figure 3.35.

4

Biomechanics

4.1 Introduction

The focus of the previous chapters has been on the definition of the static representation of objects, mainly for the visualization module of a surgical simulator. However, to model the dynamic biomechanical behavior of soft tissue in a surgical scene, additional components have to be defined. A part of this is the generation of the deformation mesh, which has already been addressed in Section 2.6. In addition to this, appropriate parameters describing the tissue behavior have to be determined. Since the parameter definition depends on the specific deformation model chosen for the simulator, an overview of the common approaches will be provided.

4.1.1 Deformation Models

Realistic behavior and real-time capability are two main features required for surgical training simulators. These contradictory requirements pose a major problem to soft tissue modeling. While high accuracy is needed to achieve realism, highly complex models usually lead to increased computation times. In addition to this, simulation stability as well as the complexity of interactive topology modifications need to be taken into account. Several approaches have been proposed to model soft tissue in the past, which will be briefly reviewed in the following.

Finite Element Method. The first approach is the Finite Element Method (FEM), which provides a rigorous representation of soft tissue physics with well-defined boundary conditions based on continuum mechanics [18]. In this approach, a body is subdivided into a number of finite elements (e.g., hexahedra or tetrahedra in 3D, quadrilaterals or triangles in 2D). Displacements and positions in an element are interpolated from discrete nodal values. For every element, the partial differential equations governing the motion of the material points of a continuum can be formulated, resulting in a discrete system of differential equations:

$$\mathbf{M\ddot{u} + C\dot{u} + K\delta u = f - r} \qquad (4.1)$$

where $\mathbf{u}$ is the vector of nodal displacements, $\mathbf{M}$ the mass matrix, $\mathbf{C}$ the damping matrix, $\mathbf{K}$ the stiffness matrix, and $\mathbf{f}$ the vector of external and $\mathbf{r}$ the vector of internal node forces, respectively. All these matrices may be time dependent. In order to simplify the calculation, a quasistatic solution is often attempted, where the dynamic part of the equations is neglected ($\mathbf{\ddot{u} = \dot{u} = 0}$). Different levels of accuracy for deformation simulation have been realized with this method, ranging from linear elastic [104] to nonlinear anisotropic systems [213]. An advantage of FEM is that only a few material parameters are required to describe the response of a physical system. These material parameters can be obtained from soft tissue measurements and directly integrated into the calculation. Nevertheless, high computation times still remain an obstacle for real-time applicability of the method. Moreover, if cutting procedures are to be allowed, which require a modification of mesh topology, element mass and stiffness matrices must be recalculated during the simulation, which is computationally intensive. Precomputation and condensation have been suggested as a remedy to some of these problems [35].

Boundary Element Method. Similar to FEM, this method is also based on continuum mechanics; however, the underlying partial differential equations are formulated as integral equations in boundary integral form. Boundary Element Methods (BEM) [33] have received increased attention since the 1980s. BEM reduces the computational complexity of the solution, since only a discretization of the object boundary is required. The degrees of freedom of the surface elements represent displacements and tractions, which are piecewise interpolated between the element vertices. The application of this method to the real-time simulation of linear elastic deformable objects has initially been discussed in [128, 186]. Using Green's functions as a fundamental solution to the elastic problem, a system of equations results:

$$\mathbf{Hu = Gt} \qquad (4.2)$$

where $\mathbf{H}$ and $\mathbf{G}$ are non-sparse $3N \times 3N$ matrices, $\mathbf{u}$ the vector of nodal displacements, and $\mathbf{t}$ the vector of nodal tractions at N nodes. A drawback of this method is that modeling is based on linear elasticity, thus not accommodating large displacements. Moreover, the objects are considered as homogeneous and isotropic. Nevertheless, the computation time is reduced to the order of $O(N^2)$. By using precomputation and superposition techniques, haptic update rates can be achieved [127].

Long Element Method. A three-dimensional object is modeled as a collection of two-dimensional elements filled with an incompressible fluid in the

Long Element Method (LEM). A static solution for elastic global deformations is obtained based on Hooke's law, Pascal's principle, and the principle of volume conservation [58]. An advantage of this approach is reduced computational complexity, since the meshes are one order of magnitude smaller than tetrahedral or cubic geometries. Also, unhomogeneities in the direction of the elements can be easily represented. Nevertheless, the LEM only yields accurate results for small deformations. Large strains lead to difficulties in volume conservation. A solution would be to update the object discretization at each time step, which is a bottleneck for real-time performance.

Tensor-Mass Model. A simplification of FEM techniques can be found in the Tensor-Mass Model (TMM), which pairs continuum mechanics' linear elasticity with mass lumping [59]. First the object is discretized into a tetrahedral mesh. Linear elastic forces are then determined following an energy-based continuum mechanical formulation. The movement of mesh vertices is computed according to Newton's law of motion. The force computation is independent of mesh topology, thus allowing an easy integration of interactive modifications such as cutting or tearing. Moreover, computational complexity is linear in the number of edges, making TMM faster then FEM approaches. The original model is based on linear elasticity and hence capable of simulating only small deformations. However, a modification of the model has been proposed in [212], which accommodates large displacements by using nonlinear strain tensors and anisotropic material laws.

Mass-Spring Model. In this method, deformable objects are represented by a network of point masses connected via springs. Mass-Spring Models (MSM) have been widely used in surgical simulation since the pioneering work by Terzopoulos [264]. The system solution is relatively easy to obtain because the equations of motion do not have to be constructed explicitly. However, an MSM is a discrete approach that only roughly approximates the true physics governing the deformation of an elastic object. Moreover, the integration of realistic tissue properties into these models is not straightforward, often necessitating manual parameter tuning. In addition, the resulting physical behavior depends on the connectivity of the point-masses. Nevertheless, topology modifications pose no problems to MSM.

Meshless Methods. Recent advances in the surgical simulation field focus on mesh-free deformation paradigms [21]. One example is the Point-Associated Finite Field (PAFF) approach, which is also known as the Method of Finite Spheres, introduced in [67]. An object is approximated by a set of points without explicit connectivity. As in classical FEM, a Galerkin formulation is applied to discretize the partial differential equations governing the deformation behavior. The displacement field is approximated using shape functions with local, spherical support. A major advantage is the point-based object representation, thus rendering
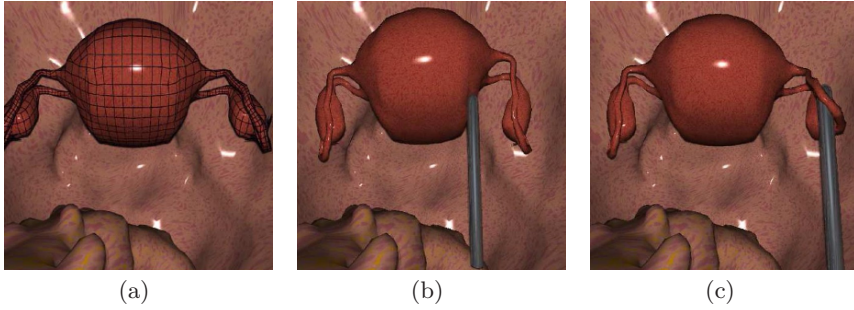
Figure 4.1. Deformation simulation with nonlinear FEM (courtesy of Roland Hutter). (a) Finite element mesh with about 2000 hexahedra, (b) Deformation of uterine corpus, (c) Deformation of Fallopian tube

an explicit meshing step unnecessary. This technique supports the simulation of large deformations as well as topology modifications like cutting. Unfortunately, the straightforward implementation of the approach is computationally intensive, thus requiring localized solutions [69]. However, such a step compromises computational accuracy.

Two different approaches to modeling soft tissue were taken in the laparoscopic and the hysteroscopic surgical simulation system, respectively.

For the former, a nonlinear, explicit Finite Element model has been developed [124], which uses reduced volume integration as well as hourglass control to improve stability. The most time consuming part in the explicit formulation is the computation of the internal element forces, including the calculation of stresses and strains. To obtain the internal forces of an element the integral

$$f_i^I = \int_{V^0} F_{il} B_k^I S_{kl} dV^0 \tag{4.3}$$

has to be determined. Here f_i^I are the internal forces in direction i of node I of the element, F_{il} are the components of the deformation gradient, B_k^I are the derivatives of the interpolation functions with respect to coordinate k, S_{kl} are the components of the 2nd Piola-Kirchhoff stress tensor, and V^0 is the initial volume of the element. Commonly, this integral is evaluated by numerical quadrature. Since the approach models objects with hexahedral meshes, an eight point quadrature is needed, i.e., the integrand in Equation 4.3 has to be computed with respect to eight different integration points. Since this computation is time consuming, reduced volume integration methods have been proposed [22]. Within these schemes the integrand is calculated just once for each element, thus reducing the computational burden. However, this technique leads to some instabilities due to spurious energy modes – so-called hourglass modes – which lead to rank deficiency. These modes have to be controlled with additional stabilization forces. A shortcoming of commonly used

formulations for obtaining these forces is their incremental character, which leads to accumulated integration errors over time. The solution to this problem is the formulation of the forces with respect to the initial configuration. This method leads to very stable results even in cases of large deformations [124]. Unfortunately, high computation times are still necessary for the full-scale nonlinear model. For a mesh of about 2000 hexahedral elements, a total of 20 GFLOPS sustained is needed for real-time interaction. Therefore, at the time of the project a high-performance parallel machine had to be built and special purpose data transfer protocols developed [226], which allowed computations in real time. Finally, while the usage of hexahedral elements helped to avoid some stability problems, interventional procedures such as cutting could not easily be integrated. An example of soft tissue deformation using the FEM approach is depicted in Figure 4.1. The selection of the material law and the determination of the appropriate parameters is outlined in Section 4.1.2.

For the latter, the MSM approach has been extended by incorporating efficient ways of volume and surface area preservation [265]. In this method, deformable solids are discretized into tetrahedra and mass points. The dynamic behavior of objects is determined according to forces derived from potential energies. These are used to preserve distances between the mesh nodes, and maintain surface areas as well as volumes of tetrahedra. Scalar constraints C are formulated to represent the deformations. In order to compute forces based on these constraints, a set of potential energies of the form

$$E(\mathbf{p}_0, \ldots, \mathbf{p}_{n-1}) = \frac{1}{2} k C^2 \tag{4.4}$$

is determined, with k denoting a stiffness coefficient. The force at a mass point $\mathbf{p}_i$ is then given by

$$\mathbf{f}_i = \left(-kC - k_d \sum_{0 \leq j < n} \frac{\partial C}{\partial \mathbf{p}_j} \mathbf{v}_j \right) \frac{\partial C}{\partial \mathbf{p}_i} \tag{4.5}$$

with $\mathbf{v}_i$ denoting the velocity of a mass point and k_d the damping coefficient. The direction of a force $\mathbf{f}$ based on a potential energy E corresponds to the negative gradient of E, i.e., a dynamic simulation resulting from these forces reduces the deformation energy of an object. A combination of several potential energy terms is used to model object behavior. The first potential energy E_D ensures distance preservation between connected vertices. The second term E_A denotes surface area preservation, based on the difference of the current and initial triangle areas. The third potential energy E_V guarantees volume preservation. It is based on the difference between the current volume of a tetrahedron and its initial volume V_0:

$$E_V(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k, \mathbf{p}_l) = \frac{k_V}{2} \frac{\left(\frac{1}{6} (\mathbf{p}_j - \mathbf{p}_i) \cdot ((\mathbf{p}_k - \mathbf{p}_i) \times (\mathbf{p}_l - \mathbf{p}_i)) - V_0 \right)^2}{V_0^2} . \tag{4.6}$$

The calculation of the signed volume with the mixed product helps to preserve the initial orientation of the vectors. Inverting a tetrahedron results in forces $\mathbf{f}_V$ that restore the original orientation of the tetrahedron. All forces resulting from the above energy terms are computed as stated in Equation 4.5. This modeling scheme allows for the integration of cutting procedures. Nevertheless, appropriate mesh parameters need to be determined, which will be discussed below.

4.1.2 Tissue Parameter Acquisition

An important element of any deformation simulation module is the determination of appropriate material parameters. Living tissue is a nonlinear, unhomogeneous, anisotropic material with viscoelastic, and in some cases also viscoplastic, properties. Moreover, soft organs are usually layered, with each layer exhibiting different mechanical properties. In addition, considerable variations between healthy and pathological tissue can be encountered. In the past, several experimental methodologies have been proposed for the mechanical testing of soft tissues to acquire appropriate tissue parameters. These can be categorized into ex vivo and in vivo approaches.

Ex Vivo Measurements

The first steps aiming at the measurement of mechanical properties of living tissue were undertaken in [301] and [89]. Their work mainly focuses on ex vivo quantification of soft tissue. Geometrically well-defined samples are excised from a body and examined using standard material testing methods, usually tension or compression tests with known boundary conditions [83]. Inflation and indentation experiments have also been carried out [65]. Moreover, recent work examines tissue response due to torsional loading [274]. Because of well-defined experimental conditions, stress and strain values can easily be obtained. However, several shortcomings limit the ex vivo approach. As shown in [139] and [201] the mechanical behavior of tissue after excision is much different from the in vivo case. This is due to loss of perfusion or muscle tone, dehydration, or temperature changes. To alleviate this problem an approach has recently been suggested in [141], where porcine liver is kept in quasi-in vivo physiologic conditions during ex vivo testing. First results show better approximations of the in vivo state; however, further research in this direction is needed.

In Vivo Measurements

Due to the mentioned limitations of ex vivo testing, several approaches focused instead on in vivo acquisition of soft tissue parameters. These follow either an indirect strategy via non-invasive imaging techniques or a direct one via

invasive in vivo testing. The latter can be further classified with regard to the selected access method to the organs – during either open surgery or minimally-invasive procedures.

Non-invasive acquisition. A holy grail of in vivo tissue property measurement is non-invasive acquisition via imaging technology. Methods have been developed using MRI [190, 87] or US imaging [198]. Strain images are acquired which allow one to map and quantify small displacements caused by propagating harmonic mechanical waves. Unfortunately, only elastic moduli can be obtained this way due to the restriction to small deformations. Thus the initial slope of the stress-strain curve can be determined; however, nonlinear behavior at large strains cannot be obtained. Nevertheless, recent studies indicate that an extension to the nonlinear case could be possible [246]. A problem that remains with these approaches is, however, the undetermined boundary conditions during the measurements.

Invasive acquisition. In invasive methods, measurements are acquired via direct contact with the organs. The easiest access to the latter is during open surgery, since fewer restrictions exist regarding the size of the testing devices. For instance, in [38], methods for measuring porcine tissue response to extension and indentation are described. Properties of in vivo human liver have been acquired during open surgery and are described in [48]. A testing probe using a haptic interface, which has been used to perform experiments on porcine organs, is discussed in [142]. A less invasive approach is acquisition during minimally invasive procedures. In [200], the TeMPeST 1-D device is introduced, which can be inserted through a trocar during an MIS intervention to perform tissue measurements. A motorized endoscopic grasper, which can apply forces and measure tool-tissue interaction, is described in [39].

Based on tissue measurements, the parameters of a deformation model can be estimated. This often involves an inverse solution, since the unknown parameters have to be determined from the measured system response. In most continuum mechanics approaches this includes the selection of an appropriate strain energy function describing the material.

Such an approach has also been taken for the laparoscopic surgery simulator LaSSo. To approximate the complex behavior of soft tissue, an exponential material law similar to the Veronda-Westmann material formulation [278] was selected; however, unhomogeneity and anisotropy were not included. In addition, due to the high water content of biological tissues, near-incompressibility was assumed. The tissue parameters could then be derived by carrying out in vivo aspiration experiments on human tissue during open surgeries [139]. This involves placing a tube against the target tissue and producing a weak vacuum in the tube. The vacuum fixes the organ to the tube, thus setting well-defined boundary conditions, and causes small deformation of the tissue, which is sucked into the tube. During this procedure, the applied pressure
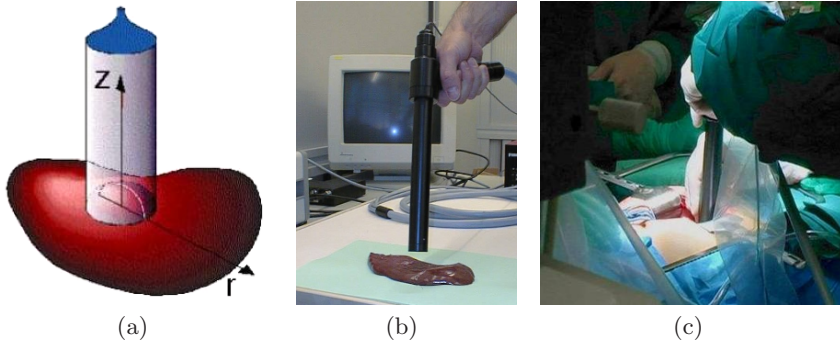
Figure 4.2. Overview of tissue aspiration device (from [138] and [259]). (a) Measurement principle based on aspiration, (b) Aspiration device, (c) Application of device during open surgery

as well as the resulting deformation is tracked. Assuming axisymmetry and homogeneous tissue in the portion covered by the aspiration tube, a complete description of the deformation is given by the profile of the aspirated tissue. An explicit axisymmetric Finite Element simulation of the aspiration experiment is used together with a Levenberg-Marquardt optimization to estimate the material model parameters in an inverse parameter determination process. Tissue properties of the uterus, kidney, and liver in healthy as well as pathological condition can be obtained with this technique [191]. Figure 4.2 visualizes the components of the aspiration strategy.

In contrast to the FEM approach, setting specific soft tissue properties for an MSM is more complex and has received less attention in the past. A major disadvantage of the MSM is that it contains possibly thousands of parameters (masses, spring constants, mesh topology), which makes it difficult to automatically determine optimal values. In many applications the parameters are set manually by iterative tuning until the visual appearance is pleasing. In order to avoid this, two approaches to determining parameters have been developed, which will be discussed below.

4.1.3 Relevance to Surgical Education

The appropriate visual as well as haptic simulation of soft tissue behavior is of major importance in a surgical training system. Force feedback rendered via a haptic device is based on the deformation model; thus, incorrect deformation modeling could be noticeable when interacting with the models. Nevertheless, it has to be noted that the degree of realism of the force computation required for a specific training effect is not yet known.

In this context the limitations of the human sensory system have to be taken into account. For instance, just noticeable differences (JND) of stimuli, as well as the usual visual dominance over other sensory channels, need to

be considered when building a system. Visual perception generally dominates haptic perception, thus allowing one to change the percept of the latter by introducing sensory conflicts (see, e.g., [252, 16]). Therefore, visual, auditory, and haptic design guidelines have to be determined (see, e.g., [112]).

Nevertheless, from a surgical point of view there are a number of cues that can be gained from the deformation behavior. For instance, as shown in [191], healthy and pathological tissue exhibit different deformation responses. Especially in hysteroscopy, the amount of force needs to be carefully controlled when moving the hysteroscope inside the uterus. Wall perforation is a common complication in hysteroscopy [183]. The most critical situation is a perforation with the resectoscope electrode during cutting procedures, since lesions of intraabdominal organs are likely. In these cases hysteroscopy usually has to be stopped and an emergency abdominal intervention performed. Therefore, small variations in the force feedback possibly alerting a surgeon during an intervention have to be integrated into the simulation. Moreover, the different compliance levels of myomas and polyps are also a significant element. Due to the higher stiffness of myomas a different resection procedure is needed than for polyps.

4.2 Previous Work

The last decade has seen a growing interest in research on the stiffness value identification problem with regard to mass-spring systems. Two main approaches have been followed so far – analytical derivation of parameters and optimization techniques.

The first strategy focuses on the determination of mathematical relationships of the MSM mesh properties based on known values. Initial work in this direction has been presented in [275]. A formula for the stiffness coefficient is proposed, which can be used to approximate the behavior of linear elastic material in the 2D plane stress case. The stiffness is determined proportional to the triangle area and Young's modulus for triangulated spring meshes. Moreover, the authors show that assigning the same stiffness values to all springs fails to simulate even a simple uniform elastic membrane. In recent work [284], energy terms for a given object using explicit continuum expressions are applied to obtain material properties. The parameters of the MSM are determined to approximate the energy term as closely as possible. They present formulas for linear springs in a plate bending model using equilateral triangle or rectangle meshes. With this method the authors were able to provide correct bending and stretching resistances. Later on they extended this approach by using angular springs to improve the lateral resistance [285]. In addition, they introduced preload, which allowed them to improve the accuracy. However, due to this they encountered some problems with boundary effects, which had to be minimized. In [222] a neuro-fuzzy network is designed to simulate the behavior of soft tissues. Linguistic terms defined

by *if-then* rules describe the tissue characteristics. The system parameters are used to initialize the neuro-fuzzy system, and are subsequently adjusted. Unfortunately, this concept implies initial manual tuning of the parameters by a user. Heuristic schemes for determining parameters for particle-based systems have been proposed in [174]. One approach obtains stiffness values in rectangular structures according to the angles between diagonal springs, while another method computes the spring constants according to the number of connections attached to a mass point.

The second strategy for parameter estimation is based on optimization processes, which try to adapt the behavior of an MSM according to a reference. A few optimization-based approaches have been proposed in the literature; for instance, in [73] the use of simulated annealing for spring constant identification is suggested. Moreover, a method is proposed to obtain a homogeneous point and mass distribution. The authors could optimize systems of a few hundred points; however, for large systems the optimization became computationally too expensive. In [171] an evolutionary strategy is suggested to identify spring parameters for cloth models. The basic idea is to optimize a cost function, which measures the difference between the behavior of a predicted and a reference model. The authors reduced the complexity of the optimization by using only five parameters for the complete model. These were specifically selected based on the application area of cloth animation. This allowed them to identify parameters for large meshes. Within the context of robot-environment interaction, a related technique has also been presented in [131]. Genetic algorithms are applied to determine several parameters of a dynamic system, including mass distribution, stiffness, and damping.

4.3 Genetic Optimization Approach

4.3.1 Outline

All previous approaches using optimization techniques to determine parameters of an MSM have a predefined topology, usually regular tetrahedral or hexahedral lattices. However, in contrast to FEM, the topology of an MSM mesh has a significant influence on the deformation behavior. Therefore, an approach has been developed which simultaneously focuses on the derivation of the spring connections as well as the elastic constants [26, 27].

The main idea of this method is the comparison of the deformation behavior of a training model with that of a known reference system, which can be based on any deformation approach. Given the external forces $\mathbf{f}^{ref}$ and the corresponding vertex positions $\mathbf{p}_i^{ref}$ of the reference model in the deformed state, the goal is to optimize the mesh parameters so that the MSM and the reference model behave similarly. The quality criterion is defined as a cost function based on the Euclidean distance between the nodes of the learning model and the reference model:

$$G(\theta) = \sum_i \|\mathbf{p}_i^{ref} - \mathbf{p}_i(\theta, \mathbf{f}^{ref})\|^2 \tag{4.7}$$

where $\mathbf{p}_i$ are the equilibrium positions of the MSM for a specific parameter vector θ and the given external forces. An optimization is then carried out to determine a set of optimal parameters $\hat{\theta}$:

$$\hat{\theta} = \arg\min_\theta G(\theta) \tag{4.8}$$

Since classical optimization methods usually get stuck in local minima, alternative methods have to be applied, one possibility being Genetic Algorithms (GA) [185]. These techniques attempt to mimic the natural evolution of species. They consist of populations of individuals, each of the latter representing a potential solution to a problem. The fitness of an individual is determined by the cost function. The optimization principle consists in evolving the population by means of genetic operators – *Selection*, *Crossover*, and *Mutation*.

The first operator selects within the population a pair of individuals (parents) with a predetermined probability. The second operator models the generation of a new pair of individuals, i.e., offspring, from the parents by concatenating the prefix of one parent with the suffix of the other one. The selected parents undergo a crossover with a predefined probability. The last operator randomly changes the value of one or more genes of the genome. A genetic algorithm can be quite sensitive to this parameter. For probabilities larger than 5%, the role of mutation becomes predominant and disturbs system convergence.

The evolution of the system by means of these genetic operators converges to a population in which all the individuals are identical. This unique species represents the best solution of the problem. Although GAs do not guarantee convergence to the global optimum, the local optimum can still be considered as a sufficiently close approximation of the exact solution.

4.3.2 2D Topology Optimization

The first focus of the optimization strategy is the identification of mesh topology in 2D. An MSM is used as the reference model, since in this case the exact solution is known. Constant mass distribution and spring stiffness is assumed, since the focus is on the mesh topology. The former will be included in the optimization later on.

The cost function is modified to include mesh connections. A binary variable α_{ij} is introduced, in order to represent the different topologies of the training mesh, which is defined by

$$\alpha_{ij} = \begin{cases} 1 & \text{if } \mathbf{p}_i \text{ and } \mathbf{p}_j \text{ are linked by spring } (i,j) \\ 0 & \text{otherwise} \end{cases} . \tag{4.9}$$

If $\mathbf{p}_i$ is the ith node of the mesh, then the internal force $\mathbf{f}_i$ applied to it is obtained by:

$$\mathbf{f}_i = -\sum_{j \in \mathcal{N}(i)} k_{ij} \alpha_{ij} (\|\mathbf{p}_i - \mathbf{p}_j\| - l_{ij}^0 \frac{\mathbf{p}_i - \mathbf{p}_j}{\|\mathbf{p}_i - \mathbf{p}_j\|}) \tag{4.10}$$

where k_{ij} is the stiffness of the spring between points $\mathbf{p}_i$ and $\mathbf{p}_j$ with rest length l_{ij}^0, and $\mathcal{N}(i)$ the set of points potentially connected to point $\mathbf{p}_i$. The total internal forces with m points are thus expressed as

$$\mathbf{f}_{int} = \sum_{i=0}^{m} \mathbf{f}_i . \tag{4.11}$$

Thus, the cost function is now defined as the difference between the deformation of a training MSM $\mathcal{M}$ with a topology defined by the values of α_{ij} and a reference model $\mathcal{R}$:

$$G(\alpha_{01}, \alpha_{02}, \alpha_{03}, ...) = \sum_{i=1}^{m} \|\mathbf{p}_i^{\mathcal{R}} - \mathbf{p}_i^{\mathcal{M}}\|^2 \tag{4.12}$$

where $\mathbf{p}_i^{\mathcal{R}}$ is the equilibrium position of the ith point of $\mathcal{R}$ and $\mathbf{p}_i^{\mathcal{M}}$ is the equilibrium position of the ith point of $\mathcal{M}$. The optimal topology of $\mathcal{M}$ can now be obtained by minimizing the cost function using the genetic algorithm approach. The genome (or individual) of the GA is a string of bits which represents a potential topology of the mass-spring model. The size of the string depends on the number of neighbors that are allowed for each node. This number defines the search area for optimizing the connections to the node under consideration. In 2D a maximum of eight neighbors is allowed for each node.

The population consists of 20 genomes, since larger sizes do not improve the results but instead increased computation times. The population is initialized with random binary values. The points $\mathbf{p}_i^{\mathcal{M}}$ of the training model are set to the positions of the reference model. Thus, the number of nodes of $\mathcal{M}$ is equal to that of $\mathcal{R}$ and each point of $\mathcal{M}$ has an exactly defined corresponding point in $\mathcal{R}$. No topology information of the reference model is used. The selection strategy of the GA is based on the roulette wheel approach. Moreover, the crossover value is set to 30% and the probability of mutation to 1%.

Different representative load cases are applied to the training model and the reference model. From this set of load cases, the genetic algorithm determines a topology of $\mathcal{M}$ which optimally approximates all the deformations of $\mathcal{R}$. Incorporating all load cases into the genetic algorithm from the start yields the best results. Figure 4.3 shows the optimization outcome for
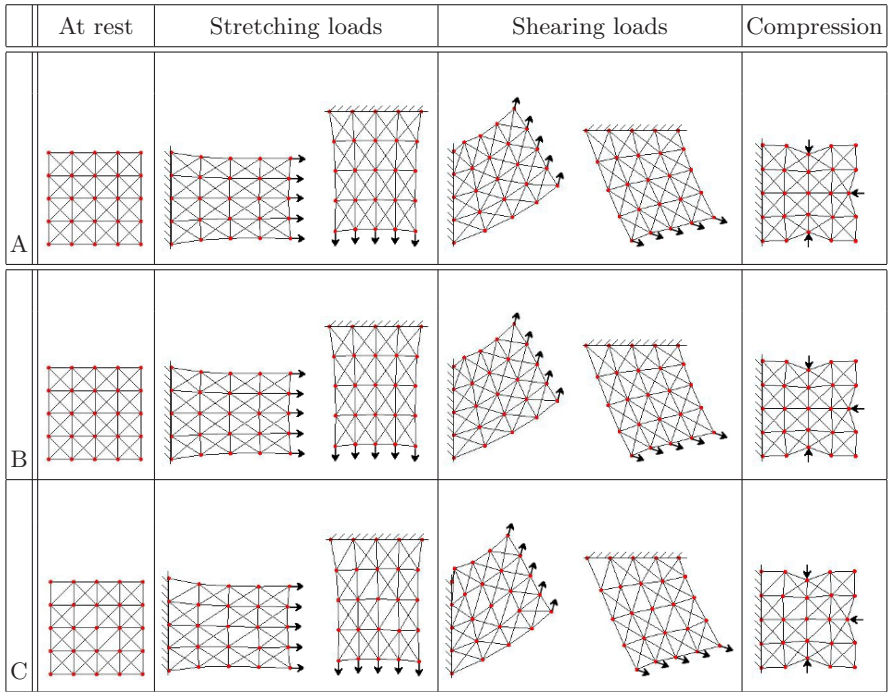
Figure 4.3. Genetic optimization of mesh topology (with kind permission of Springer Science+Business Media, LLC).

an isotropic mesh configuration. Row A depicts the equilibrium positions of the reference model while applying different loads. The first column contains the model without loads and the second shows deformations under stretching loads. Shearing forces are applied in the third column and compression forces in the fourth. Rows B and C are two different results obtained with the genetic algorithm. Several independent trials to obtain the regular isotropic mesh have been carried out. In most cases, the optimization fully recovered the topology of the reference mesh in about 400 iterations as shown in row B. In these cases the cost function reaches zero. However, in the case shown in row C the algorithm did not converge after the maximum number of 1000 iterations. However, even if only a suboptimal solution missing a few links has been found, the global behavior remains close to that of $\mathcal{R}$. This is also illustrated by the resulting low cost function value reached, $G = 0.01$. Similar results were also achieved for anisotropic 5×5 mesh configurations.

The proposed method has also been applied to irregular meshes. In Figure 4.4 the resulting configurations are depicted, using the same genetic parameters as previously defined. The first row contains the reference model of a cross-section of a kidney, while the second shows the optimized mesh. Not all

|	At rest	Load case 1	Load case 2
Reference			
Result			
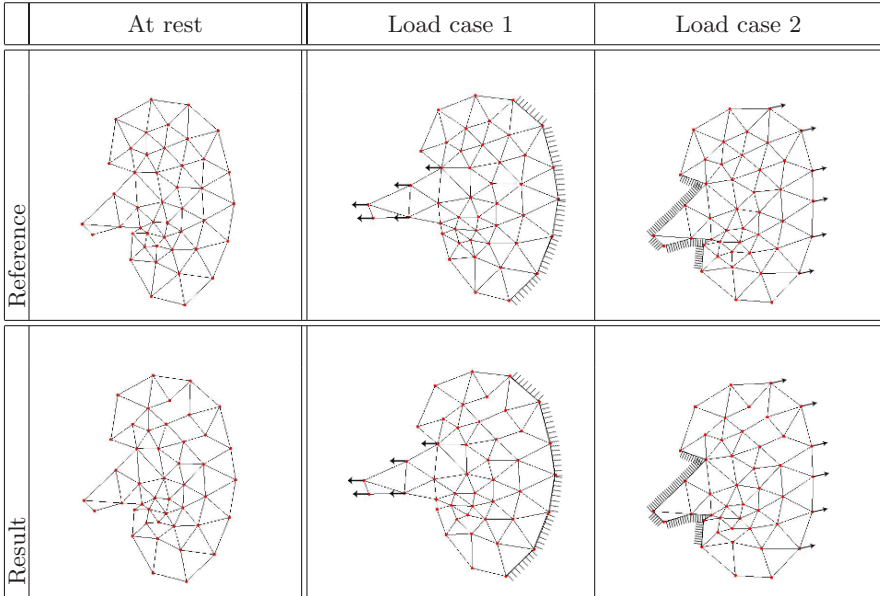
Figure 4.4. Topology identification for irregular mesh (with kind permission of Springer Science+Business Media, LLC).

connections could be recovered; however, the global behavior remains similar to the observations on the reference.

4.3.3 Extension to 3D Topology Identification

Due to the increasing number of springs in the 3D case, the genetic algorithm has to be optimized in order to ensure fast convergence towards the optimal solution. Furthermore, a reasonably limited neighborhood for possible spring connections has to be defined for 3D. The following experiments were performed with the $\mathcal{N}_{26}$ Moore neighborhood, assuming a maximum of 26 neighbors. As in the 2D case, an individual is described by a vector of binary values which represents a potential topology of the mass-spring model. Increasing the population size to large values did not have a significant influence on the convergence speed of the algorithm or on the outcome. Good results were achieved with a low value of 10 genomes, which also resulted in reasonable computation times. While the crossover operator improves convergence in the early stages of the process, its influence becomes of secondary importance later, when the population is close to the optimum. Therefore, the crossover probability is fixed to a high value of 80%. In contrast to this, the mutation parameter has a significant effect throughout the whole evolution. While a high value speeds up the convergence at the beginning of the
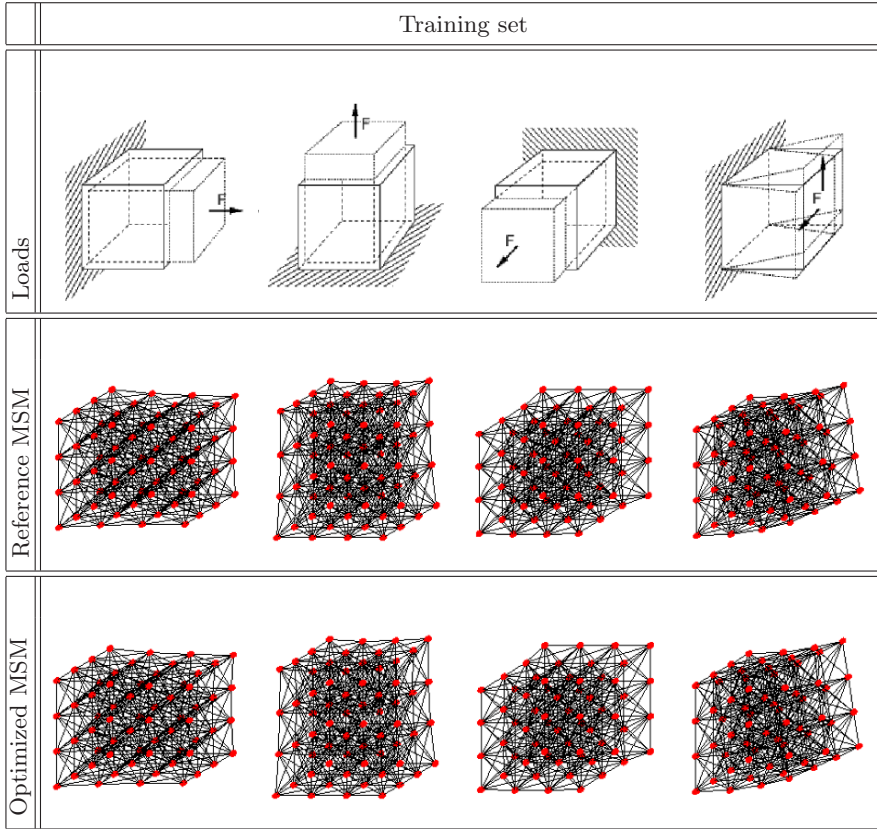
Figure 4.5. Derivation of 3D mesh topology (with kind permission of Springer Science+Business Media, LLC).

evolution, a lower one gives better results in the later stages. Due to this, an adaptive mutation probability is used, which is inversely proportional to the generation number G:

$$P_{mut}(G) = \frac{a}{G} + b \,. \tag{4.13}$$

The coefficients a and b are determined heuristically – in the case of 3D topology identification, setting $a = 0.2$ and $b = 0.001$ has been found to provide the best convergence properties. Using the adaptive mutation strategy, the convergence speed could be considerably improved.

The results in the 3D case using a reference mesh consisting of $4{\times}4{\times}4$ mass points and 468 springs is depicted in Figure 4.5. The first row shows four out of six different load cases used in the training process. The selection of these test cases, including stretching and shearing, was motivated by suggestions made in [73]. The second row depicts the reference MSM deformed by the loads, while

row three illustrates the best topology obtained by the genetic algorithm. Here, 5 of 468 springs are missing; however, the normalized difference of point positions between the reference model and the learning model is negligible.

4.3.4 Simultaneous Topology and Spring Constant Identification

The described approach can be further extended by including the identification of individual spring constants in the process. This is achieved by representing springs in the optimization population as a vector of real-valued positive constants instead of binary connections. A spring is only present if its stiffness value is greater than a predefined threshold, thus allowing the simultaneous identification of topology and stiffness values.

The proposed strategy requires an adaptation of the mutation operator. Instead of swapping bits, a random value is added to the current stiffness. The former is limited to the interval I, as well as normally distributed with zero mean and standard deviation σ. Since a constant σ will cause the random steps to be too high, once the population is close to the optimum, it is decreased during the optimization process.

$$\sigma(G) = P_{mut}(G) \cdot N_I \tag{4.14}$$

where N_I is the size of the variation interval and G the number of the current generation. Crossover probability is set to 80% and population size to 10 individuals. Experiments are performed with a limit of 4000 generations.

In the first step, the recovery of topology and stiffness constants of a known 2D MSM is tested. The mesh has 5×5 points and 72 springs. All spring stiffnesses of the reference are set to the same stiffness value of 5.0. The interval of the random mutation value is assigned to $I = [0.0, 10.0]$. Using the described approach, the complete topology of the reference model could be recovered, i.e., all spring constants are greater than the predefined threshold $t_0 = 0.01$. Furthermore, the mean value of the retrieved stiffness is equal to 5.01 with a standard deviation of 0.65. Nevertheless, instead of approximating a known MSM configuration, a different reference model should be used. This could be a more accurate, offline deformation computation, as well as actual deformations directly acquired from an in vivo study, such as the aspiration experiments described above.

As an initial test, the optimization process is extended to focus on training the learning MSM based on an FEM as reference. The latter consists of quadrilateral 2D elements, where the vertex locations correspond to the mass points in the MSM model. The linear elastic deformations are computed using Young's modulus $E = 1MPa$ and Poisson coefficient $\nu = 0.3$.

Analogously to the previous approaches, the cost function is based on the distance between the FEM and MSM nodes. In addition, spring constants are again limited to the interval $I = [0.0, 10.0]$. Twelve load cases are applied, including shearing and stretching deformations. In order to validate the results, additional test load cases are performed, using the obtained set of MSM
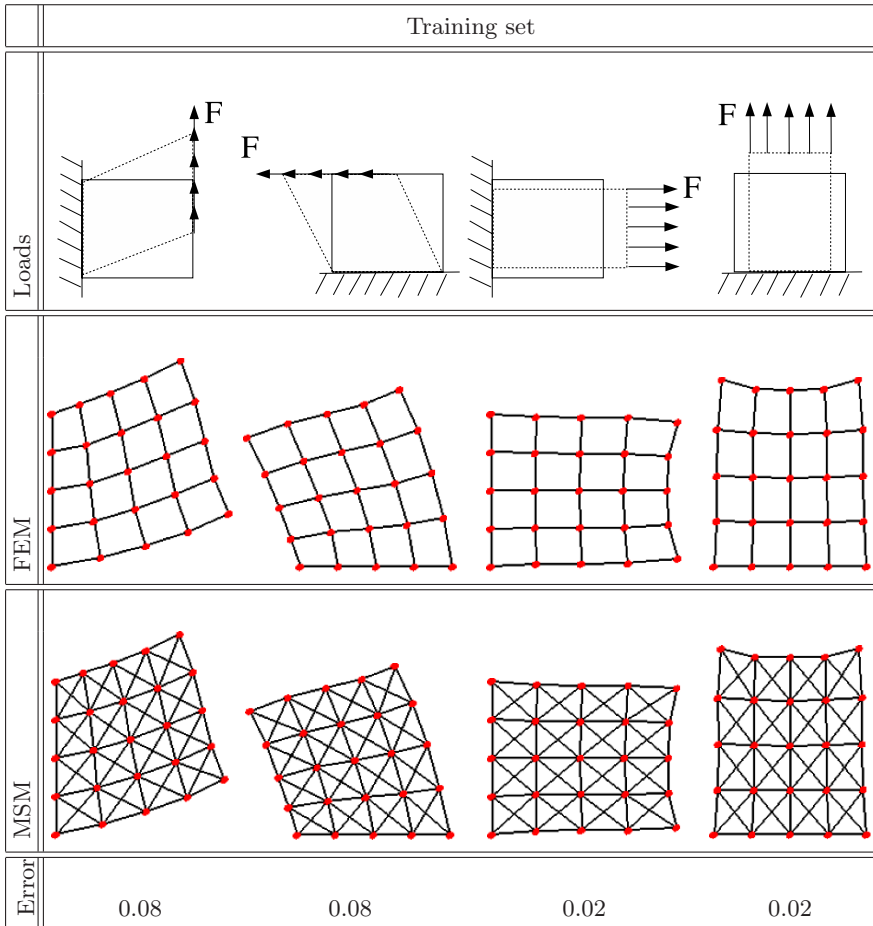
Figure 4.6. Input data and result after training process (with kind permission of Springer Science+Business Media, LLC).

parameters. Figure 4.6 shows the results of the optimization step. The first row depicts four of the twelve test loads applied to the FEM. The second row shows the FEM deformations, while the third row contains the learning MSM with the best stiffness values. The last row shows an error metric for the deformation differences, based on the distance of the FEM and MSM nodes in relation to the smallest spring rest length of the optimized MSM.

In Figure 4.7, the MSM solution is compared for evaluation purposes to three elastic linear FEM deformations not included in the training set. As indicated by the error metric, the stiffness values of the learning MSM are equally well adjusted for both the training and test FEM cases. The obtained stiffness values exhibit a non-Gaussian distribution and are located mainly within the interval $\tilde{I} = [2.0,\ 9.0]$, with peaks around 3.5 as well as 7.0. These
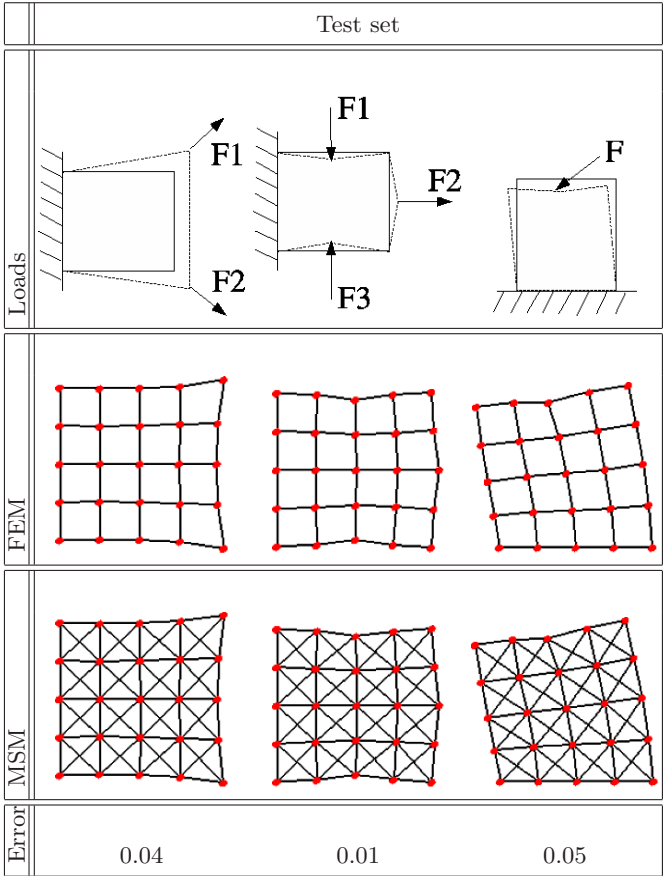
Figure 4.7. Comparison of optimized MSM and FEM with additional test load cases (with kind permission of Springer Science+Business Media, LLC).

results also show that a linear elastic material cannot be approximated with one homogeneous stiffness parameter, which is in line with the results reported in [275].

4.4 Analytical Derivation

4.4.1 Overview

The underlying idea of the second approach is the derivation of analytical expressions for the spring parameters based on more complex reference models [167]. As an initial attempt, a linear elastic FEM formulation is used as the ground truth. After linearizing the MSM equations, the stiffness matrices of both deformation models are equated. Based on this, explicit formulas for

MSM stiffness parameters can be obtained. Moreover, the condition under which these are valid can be explicitly determined. The proposed technique will first be detailed for the case of a constant strain triangular element and thereafter extended to tetrahedral elements. Nevertheless, the strategy is general in the sense that the procedure can be applied to any kind of selected FEM element.

4.4.2 Derivation for Constant Strain Triangle

The first step in the parameter estimation technique is the derivation of the FEM stiffness matrix. To outline the process, a simple triangular constant strain element is used. It is a 3-node triangle with linear displacement functions and nodes $(x_i, y_i), (x_j, y_j), (x_k, y_k)$. In the quasistatic case, the resulting system is

$$\mathbf{f} = \mathbf{K}_{cst}\mathbf{u} \tag{4.15}$$

where $\mathbf{f}$ is the force vector at the triangle nodes (x_i, y_i), $\mathbf{K}_{cst}$ the element stiffness matrix, and $\mathbf{u}$ the vector of corresponding node displacements. For the linear elasticity triangle, the 6×6 stiffness matrix is given by

$$\mathbf{K}_{cst} = \begin{bmatrix} \mathbf{K}_{i,i} & \mathbf{K}_{i,j} & \mathbf{K}_{i,k} \\ \mathbf{K}_{j,i} & \mathbf{K}_{j,j} & \mathbf{K}_{j,k} \\ \mathbf{K}_{k,i} & \mathbf{K}_{k,j} & \mathbf{K}_{k,k} \end{bmatrix} \tag{4.16}$$

with the submatrices defined according to

$$\mathbf{K}_{i,k} = \frac{tE}{4A(1+\nu)(1-\nu)} \begin{bmatrix} \beta_i\beta_k + \gamma_i\gamma_k(\frac{1-\nu}{2}) & \beta_i\gamma_k\nu + \beta_k\gamma_i(\frac{1-\nu}{2}) \\ \beta_k\gamma_i\nu + \beta_i\gamma_k(\frac{1-\nu}{2}) & \gamma_i\gamma_k + \beta_i\beta_k(\frac{1-\nu}{2}) \end{bmatrix} \tag{4.17}$$

where t is the thickness of the elastic model, E the Young's modulus, ν the Poisson ratio, and A the triangle area. Moreover, the following variable substitutions have been used:

$$\begin{aligned} \beta_i &= y_j - y_k & \gamma_i &= x_k - x_j \\ \beta_j &= y_k - y_i & \gamma_j &= x_i - x_k \\ \beta_k &= y_i - y_j & \gamma_k &= x_j - x_i\,. \end{aligned} \tag{4.18}$$

The next step is the derivation of an equivalent matrix for an MSM. Unfortunately, the underlying equations are nonlinear in the node positions, thus requiring a linearization. Under the assumption that parameters can be found such that both models deform the same around the linearized point, the springs are linearized in the displacements. A 2D MSM structure consisting of

three nodes connected by three linear springs is considered. The equilibrium force at node i at position $\mathbf{p}_i = (x_i, y_i)$ connected to spring (i, j) is given by

$$
\begin{pmatrix} f_{(i,j),x} \\ f_{(i,j),y} \end{pmatrix}^T = \kappa_{(i,j)} \left[(\mathbf{p}_i - \mathbf{p}_j)(1 - \frac{l^0_{(i,j)}}{\|\mathbf{p}_i - \mathbf{p}_j\|}) \right] \tag{4.19}
$$

where $f_{(i,j),x}$ is the x-component of the force at node i and $\kappa_{(i,j)}$ the spring constant of the connecting spring. In order to linearize the forces using the Taylor series, the partial derivatives are required. The first derivative of $f_{(i,j),x}$ with respect to x_i is given by

$$
\frac{\partial f_{(i,j),x}}{\partial x_i} = \kappa_{(i,j)} \left[\left(1 - \frac{l^0_{(i,j)}}{\|\mathbf{p}_i - \mathbf{p}_j\|} \right) + \frac{(x_i - x_j)^2 \cdot l^0_{(i,j)}}{\|\mathbf{p}_i - \mathbf{p}_j\|^3} \right] \tag{4.20}
$$

and with respect to y_i by

$$
\frac{\partial f_{(i,j),x}}{\partial y_i} = \kappa_{(i,j)} \frac{(x_i - x_j)(y_i - y_j) \cdot l^0_{(i,j)}}{\|\mathbf{p}_i - \mathbf{p}_j\|^3} . \tag{4.21}
$$

The derivatives of $f_{(i,j),y}$ are determined analogously. Based on the Taylor expansion, the complete system of equations in matrix notation for a linearized MSM is given by

$$
\mathbf{f} \simeq \mathbf{K}_{msm}\mathbf{u} \tag{4.22}
$$

where the symmetric 6×6 MSM stiffness matrix is defined according to

$$
\mathbf{K}_{msm} = \begin{bmatrix} \mathbf{A}_{i,j} + \mathbf{A}_{i,k} & -\mathbf{A}_{i,j} & -\mathbf{A}_{i,k} \\ -\mathbf{A}_{i,j} & \mathbf{A}_{i,j} + \mathbf{A}_{j,k} & -\mathbf{A}_{j,k} \\ -\mathbf{A}_{i,k} & -\mathbf{A}_{j,k} & \mathbf{A}_{i,k} + \mathbf{A}_{j,k} \end{bmatrix} \tag{4.23}
$$

with symmetric submatrices

$$
\mathbf{A}_{i,j} = \frac{\kappa_{(i,j)}}{(l^0_{(i,j)})^2} \begin{bmatrix} (x_i - x_j)^2 & (x_i - x_j)(y_i - y_j) \\ (x_i - x_j)(y_i - y_j) & (y_i - y_j)^2 \end{bmatrix} . \tag{4.24}
$$

The derived linearized MSM matrix and the FEM matrix are then equated to determine the unknown spring parameters. Unfortunately, the 2x2 FEM submatrices $\mathbf{K}_{i,j}$ are in general not symmetric, except in the case of $\nu = \frac{1}{3}$. Therefore, an analytical solution can be derived only in this special case.

Without loss of generality, the considered triangle is transformed, to simplify the calculations. Thus, node $\mathbf{p}_i$ is placed at the origin and $\mathbf{p}_j$ on the
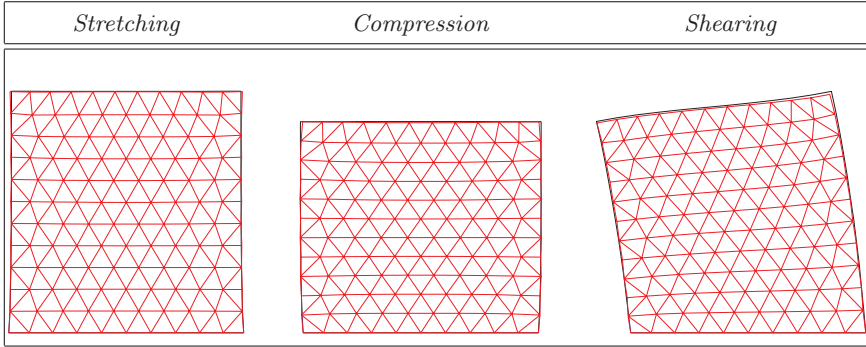
Figure 4.8. Simulation results with triangular mesh (338 springs, 128 nodes, 211 triangles)

x-axis. Equating the individual components of the matrices gives nine equations, three for each stiffness parameter. In the simplified form and with $\nu = \frac{1}{3}$, the equations for $\kappa_{(i,j)}$ are given by

$$\frac{Et}{4A\frac{8}{9}}\left[-y_k{}^2 + \frac{1}{3}x_k{}^2 - \frac{1}{3}x_k x_j\right] = -\kappa_{(i,j)}$$

$$\frac{Et}{4A\frac{8}{9}}\left[-x_k{}^2 + x_k x_j - \frac{1}{3}y_k{}^2\right] = 0 \qquad (4.25)$$

$$\frac{Et}{4A\frac{8}{9}}\left[\frac{2}{3}x_k y_k - \frac{1}{3}x_j y_k\right] = 0 \,.$$

The last two equations are only fulfilled if the triangle is equilateral, i.e., the third vertex is located at $(x_k, y_k) = (\frac{x_j}{2}, \sqrt{3}\frac{x_j}{2})$. Thus, the formula for the stiffness parameter is given (after backtransformation) by

$$\kappa_{(i,j)} = \frac{Et}{A} \cdot \frac{3}{16} \cdot (l^0_{(i,j)})^2 = Et\frac{\sqrt{3}}{4} \,. \qquad (4.26)$$

Because the triangle has to be equilateral, the formulas for the other two springs are identical. In a general MSM system, the contributions of all adjacent triangles have to be added. Thus, the formula becomes

$$\kappa_{(i,j)} = \sum_e Et\frac{\sqrt{3}}{4} \,. \qquad (4.27)$$

An explicit solution can be given for a mesh with equilateral triangles and the special case of $\nu = \frac{1}{3}$. Figure 4.8 shows a comparison of a reference object simulated with FEM (black outline) and an MSM (red edges) with analytically derived parameters according to the proposed formula. Note that the triangles of the MSM mesh are not exactly equilateral. The FEM models

a linear elastic membrane simulated with constrant strain triangles (plain stress case, $E = 14630\,Pa$, $\nu = \frac{1}{3}$). Stretching deformation in the y-direction is shown on the left, compression in the middle, and shearing on the right.

Nevertheless, the proposed method is valid only for a specific Poisson ratio. In order to determine the behavior of a more general material, a corresponding optimization problem can be considered.

$$\hat{\kappa} = \arg\min_{\kappa} \sum_{i,j} \left([\mathbf{K}_{cst}]_{(i,j)} - [\mathbf{K}_{msm}(\kappa)]_{(i,j)}\right)^2 . \tag{4.28}$$

Finding an optimal stiffness parameter for arbitrary Poisson coefficients, but equilateral triangles using this technique shows smaller normalized errors for nearly incompressible materials than for compressible ones. Nevertheless, this path was not further investigated. Instead, a similar approach has been examined for tetrahedral meshes.

4.4.3 Derivation for Tetrahedral Element

In order to apply the same derivation strategy for a tetrahedral element, the stiffness matrix first has to be determined for a single tetrahedron. Assuming linear elasticity with no initial stresses or strains, the potential energy of the body under external loads is given according to [311] by

$$E = \int_{\Omega} \frac{1}{2}\boldsymbol{\epsilon}^T \boldsymbol{\sigma} - \mathbf{fu}\ d\Omega \tag{4.29}$$

where $\mathbf{u} = \mathbf{u}(x, y, z)$ is the displacement field, $\mathbf{f} = \mathbf{f}(x, y, z)$ the loads applied to the body Ω, and $\boldsymbol{\epsilon}$ and $\boldsymbol{\sigma}$ stresses and strains in vector notation, respectively. The normal and shear strains in vector notation are expressed as

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ 2\epsilon_{xy} \\ 2\epsilon_{xz} \\ 2\epsilon_{zx} \end{bmatrix} = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ \partial/\partial y & \partial/\partial x & 0 \\ 0 & \partial/\partial z & \partial/\partial y \\ \partial/\partial z & 0 & \partial/\partial x \end{bmatrix} \mathbf{u} . \tag{4.30}$$

With normal and shear strains also in vector notation, the stress-strain relationship for modest loads is defined by Hooke's Law:

$$\boldsymbol{\sigma} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-2\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-2\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1-2\nu \end{bmatrix} \boldsymbol{\epsilon}$$

$$= \mathbf{D}\,\boldsymbol{\epsilon} \tag{4.31}$$

with Young's modulus E and Poisson ratio ν describing the elastic material. In the Finite Element discretization the deformation field in the tetrahedral element is expressed by shape functions and nodal displacements

$$\mathbf{u} = \sum_{i=1}^{4} N_i(x,y,z)\tilde{\mathbf{u}}_i = \mathbf{N}\tilde{\mathbf{u}}\,. \tag{4.32}$$

In the case of linear interpolation, the shape functions are given by

$$N_i(x,y,z) = \frac{1}{6V}(a_i + b_i x + c_i y + d_i z) \tag{4.33}$$

with V being the volume of the element. Insertion of Equation 4.32 into the strain formula yields

$$\boldsymbol{\epsilon} = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ \partial/\partial y & \partial/\partial x & 0 \\ 0 & \partial/\partial z & \partial/\partial y \\ \partial/\partial z & 0 & \partial/\partial x \end{bmatrix} \mathbf{N}\tilde{\mathbf{u}} = \mathbf{B}\tilde{\mathbf{u}}\,. \tag{4.34}$$

Thereafter, based on Equation 4.29, the total potential energy of the element can be derived as

$$\tilde{E} = \int_{\Omega} \left(\frac{1}{2}(\mathbf{B}\tilde{\mathbf{u}})^T \mathbf{D}(\mathbf{B}\tilde{\mathbf{u}}) - \tilde{\mathbf{u}}\mathbf{f}\,d\Omega\right)\,. \tag{4.35}$$

Minimizing the potential with respect to the unknown displacements yields the matrix system for the element as

$$\mathbf{K}\tilde{\mathbf{u}} = \mathbf{f} \tag{4.36}$$

with the stiffness matrix
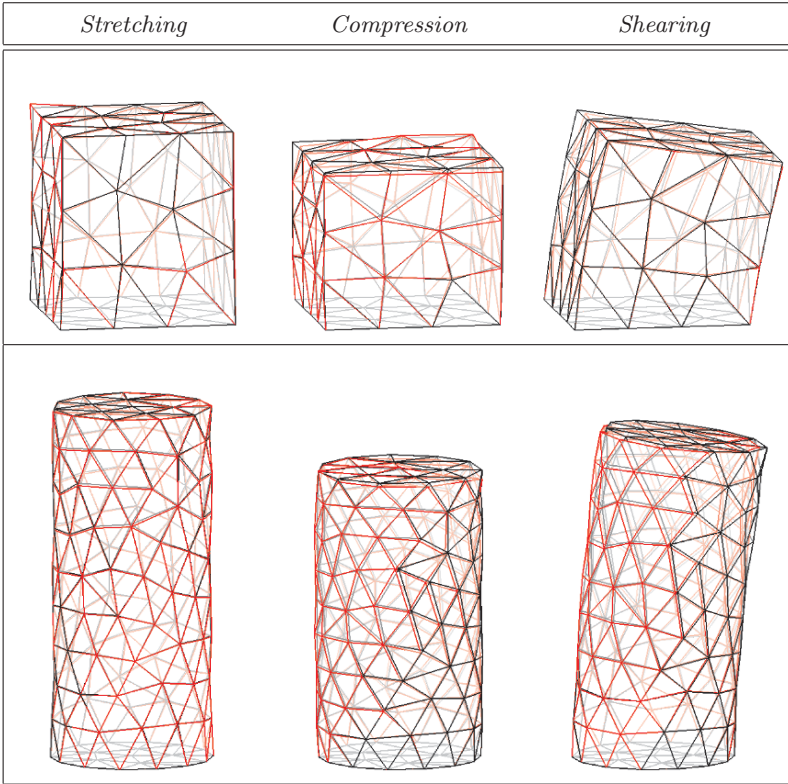
Figure 4.9. Comparison of FEM and MSM deformation in 3D. Top: Cubic object (72 nodes, 323 springs, and 193 tetrahedra), Bottom: Cylindrical object (182 nodes, 916 springs, and 599 tetrahedra)

$$\mathbf{K} = \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} \, d\Omega = V \mathbf{B}^T \mathbf{D} \mathbf{B} \,. \tag{4.37}$$

Setting up the corresponding matrix for the MSM is done analogously to the 2D case. Equating both matrices shows that a solution can be found only for $\nu = \frac{1}{4}$, since only in this case is the FEM stiffness matrix symmetric. Unfortunately, the behavior of such a material would be quite different from the usually assumed near-incompressibility of soft tissue. Therefore, an approximation based on analytically optimizing the spring coefficients can be examined. The latter are adapted to minimize the least squares error between the elements of both matrices:

$$\min_{\theta} = \sum_{i,j} ([K_{FEM}]_{i,j} - [K_{MSM}]_{i,j}) \,. \tag{4.38}$$

Examining a regular tetrahedron with edge length l yields the formula for the spring coefficients of a corresponding MSM tetrahedral configuration:

$$k_{(i,j)} = \frac{2\sqrt{2}}{25} lE \; . \tag{4.39}$$

Unfortunately, optimization for arbitrary shapes can result in negative spring coefficients. Especially badly shaped tetrahedral elements, such as slivers or spindles, cause problems. Experiments similar to the 2D case were carried out to determine the validity of this rather heuristic optimization approach. Figure 4.9 shows the initial results for two 3D meshes.

4.5 Discussion

Both presented approaches only scratch the surface of the complex problem of MSM parameter estimation. Unfortunately, no general framework could be developed which works on arbitrary, large meshes. While initial promising results could be obtained with both strategies, still a number of limitations exist.

The optimization-based technique has a number of weaknesses. The identification process is computationally very expensive, requiring running times on the order of hours. Although processing times can be reduced by performing parallel computations, it is questionable whether large meshes can be handled with this technique. An alternative would be to limit the stiffness constants to only a small number of discrete values; however, the selection of those would be rather arbitrary. Further problems exist due to local minima, stability, and smoothness of the objective functions. In addition, heuristic tuning of the GA parameters is also often necessary.

The analytical derivation of tissue parameters also exhibits several drawbacks. It was not possible to derive a general formulation for arbitrary meshes and materials. Instead, an explicit solution could be obtained only for regular elements and specific material behavior. Approximations for more general properties can, however, be calculated by solving an optimization problem. Moreover, the minimized error term can also be used to predict for which model properties the approximate formulas perform well. An alternative could be to extend the MSM model with additional area and/or volume preserving forces. This could help to approximate the reference model by adding extra degrees of freedom.

It has to be mentioned that another problem of the suggested strategies is the reference model. In both cases a quasistatic, linear elastic representation is used, which in itself is only a rough approximation of soft tissue behavior. Due to the disregard of geometrical nonlinearity, linear elastic approximations produce substantial error for large deformations. It remains to be seen how easily the approaches can be extended to the nonlinear case. Moreover, the reference deformations would optimally be acquired directly from living organs. Nevertheless, it can still be concluded that the proposed methods provide a

more formal approach to obtaining MSM parameters compared to iterative manual tuning of simulated organ models by medical experts.

5

Conclusion

5.1 Summary

The idea of using VR-based surgical simulators for training of prospective surgeons has been a topic of research for more than a decade. However, surgical simulation is still far from being integrated into the medical curriculum. A number of still open questions exist, for instance the level of simulation realism which is needed for effective learning, the identification of the surgical skill components which are to be trained, as well as the validation of the training effect. Current research strives to address these problems with a new generation of highly realistic simulators. A key element of realism is the fidelity and variability of the training scene, reflecting differences in individual patients. In this book the main components of the generation process focusing on case-by-case scenarios have been described.

The first element of the framework is the scene geometry. Statistical modeling of healthy anatomy is used to capture macroscopic variability in individual patients. Based on this, new anatomical instances are derived. In order to generate the model, medical patient data have been collected from volunteers and segmented. Parameters based on clinically meaningful metrics can be specified to replicate a new organ geometry. Thereafter, artificial tumor growth is applied to encode pathology genesis. Tumor geometries are obtained and seamlessly integrated into the models of the healthy anatomy. The large number of possible pathologies as well as the enormous range of their manifestations makes a statistical approach infeasible. Three different models for the generation of tumors are presented – a purely geometrical, CAD-based algorithm, which artificially creates tumor geometries by defining object skeletons and subsequently revolving contour curves; a cellular automaton that is able to simulate the growth of uterine leiomyomas; and a particle-based model, which integrates additional physiological information into the growth process. These procedures result in a triangular surface representation of the scene geometry. In a final step, a tetrahedrization of the generated objects is carried out. This is necessary for a number of different simulator modules, e.g.,

collision detection, collision handling, or interactive scene modifications such as cutting. A particle-based mesh generation procedure has been presented that produces high quality tetrahedral meshes.

The next step in the scenario definition is the object appearance. Providing correct visual information is indispensable in surgical simulation if a realistic training environment is required. A central point is the texture generation for healthy and pathologically altered organs. Correct visualization of pathologies is of central importance for configurable anatomical models, since the visual appearance of diseased organs often differs substantially from that of healthy ones. In order to solve these problems, automatic texture generation methods based on intraoperative images have been presented. Statistical as well as procedural texturing methods have been investigated. Organ specific base textures, i.e., textures without blood vessels, can be computed automatically by means of a texture analysis/synthesis process. Moreover, additional surface detail can be added to the base texture by procedural methods. Thereafter, the created textures are mapped to the 3D mesh geometries. This is done by mesh parameterization, which takes into account the visibility of seams and distortion reduction. Finally, textures are blended across seams and junctions between different objects, e.g., pathologies and healthy tissue, to reduce boundary artifacts.

The last part of the training scene generation focuses on the biomechanical behavior of the objects. The stiffness value identification problem with regard to mass-spring systems has been analyzed. Two main approaches have been followed – optimization techniques as well as the analytical derivation of parameters. The first strategy is based on optimization processes, which try to adapt the behavior of an MSM according to a reference. The presented method is based on the comparison of the deformation behavior of the training model with that of a known reference system. A solution is obtained using genetic algorithms. Topology and stiffness values of small reference MSMs could be determined, and the behavior of FEM deformations approximated. The second strategy focuses on the determination of mathematical relationships of MSM mesh properties based on known values. A new approach to derive analytical expressions for spring parameters based on an isotropic linear elastic reference model is described. Expressions for triangular and tetrahedral mesh topologies are provided.

Combining the three presented strategies allows the definition of variable training scenes for VR-based surgical simulation. The discussed methods have been applied to obtain scenarios for different surgical simulator systems. The main focus of the related developments is hysteroscopy simulation, for which the setup will be briefly presented.

5.2 Hysteroscopy Simulation

The driving application of the presented work is the simulation of hysteroscopic interventions. Therapeutic hysteroscopy has become a common technique in gynecological practice [42]. Nevertheless, a number of potentially dangerous complications exist – some of these being uterine wall perforation, mismanagement of distension fluid, or intrauterine bleeding. In order to exemplarily show the necessity of higher level training, these complications are reviewed in more detail.

Wall perforation. In [214] the rate of complications for therapeutic interventions is reported as 17%. According to [183], 97% of hysteroscopic interventions are performed with resectoscopes, with up to 9% of them leading to perforations. The most critical situation is wall perforation with resectoscope electrodes during cutting procedures, since lesions of intraabdominal organs are likely. In these cases hysteroscopy usually has to be stopped and an emergency abdominal intervention performed.

Fluid mismanagement. A fundamental prerequisite of hysteroscopy is the proper distension of the uterine cavity, also known as hydrometra. Inflow and outflow of the distension fluid is accomplished via the endoscopic tool and controlled with valves, while the pressure of the liquid is provided by a pump. It is essential to select the correct pressure settings for the hydrometra according to muscle tone and uterine wall thickness [209]. The most critical complication is fluid overload due to absorption in tissue or intravasation through sectioned vessels [183]. This can have serious adverse effects, which can lead to cerebral edema or even death.

Bleeding. In addition to the inflation of the uterus, the fluid flow also ensures a clear visibility in the cavity during interventions. Obscurations can be caused by endometrial bleeding, floating tissue fragments, or air bubbles. It is the gynecologist's experience in correctly controlling the inflow and outflow through the valves that leads to a proper view of the scene. For instance, small local bleeding can be stopped by increasing fluid pressure.

Specialized training is necessary to reduce the rate of complications. The acquisition of crucial skills, like the correct handling of the valves, has to be an integral part of the training of every gynecologist. The discussed framework can be used to supply specific training situations in order to rehearse the management of complications. For instance, the resection of tumors close to the Fallopian tubes is especially dangerous, since the thickness of the uterine wall is significantly lower in this area. In order to train for the handling of this situation, a virtual scene could be generated with a pathology placed inside the cavity at a critical location.

All the components necessary for a highly realistic surgical simulator, as outlined in Chapter 1, have been integrated into our training system. These include a specialized haptic interface [250], an immersive environment [114], fluid simulation [241], deformation computation [265], collision detection [118],
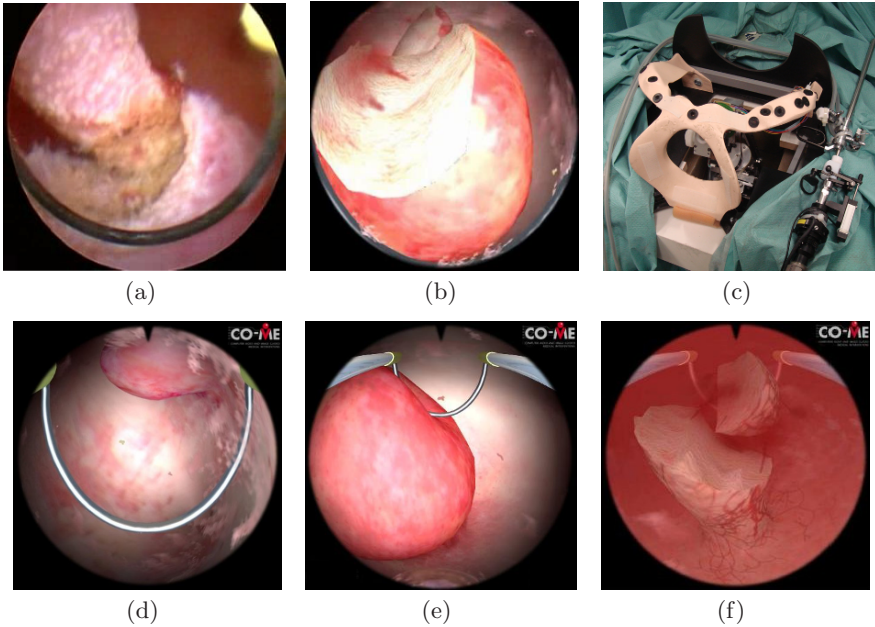
Figure 5.1. Snapshots of various aspects of the hysteroscopy simulator. (a) Cutting of real myoma, (b) Virtual view of cutting procedure, (c) Hysteroscope and haptic mechanism, (d) Virtual surgical scene, (e) Deformation of myoma, (f) Bleeding after incision into vessels

tissue cutting [115, 254], bleeding simulation [304], and enhanced visualization [11]; all these are combined into a specialized software framework [269]. Figure 5.1 shows some of the individual system components. The simulation setup is currently evaluated in an extensive study in collaboration with the OB/GYN department of the University Hospital Zurich. The scenes used in this study were generated with the described scenario definition framework.

5.3 Extension: Vessel Generation

A straightforward addition to the scene definition scheme is the generation and integration of vascular structures. Since this topic has also been investigated in the context of both surgical simulator projects, this extension will be briefly elucidated.

Vessel structures affect surgical simulation on several levels. Firstly, they are the carrier of blood circulation and thus an integral part of a patient physiology model. Moreover, they affect the visual appearance of organ surfaces and thus have to be included in the texturing process. As previously mentioned, vascular patterns contain vital information for the surgeon. Nevertheless,

(a)   (b)   (c)
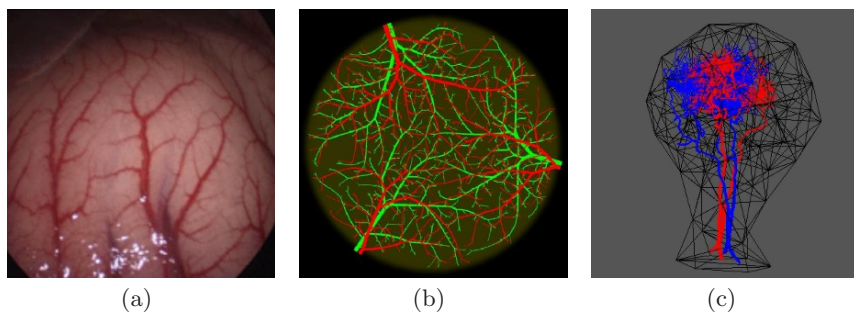
Figure 5.2. Generation of vascular systems (courtesy of Volker Meier and Dominik Szczerba). (a) Real vessels on the stomach, (b) Synthesized arterial and venous networks in 2D, (c) Vascular structure grown in 3D polyp mesh

vessel trees also have to be treated as geometric entities. This is necessary during cutting procedures, to determine the source of bleeding as well as to obtain further information affecting the rendering of the blood flow. Also, processes such as the intravasation occuring in hysteroscopy require a vascular model. Finally, larger vessels have to be treated as deformable objects duing tool-tissue interaction.

Since vessel structures are usually too complex to be synthesized with texture generation approaches, and moreover, have to be included in the simulation as geometric objects, a specialized generation process is required. In the past a number of methods for modeling of vascular systems have been suggested. In [232], a constrained constructive optimization technique is introduced, where new terminal segments are added to an initial root of a tree according to a set of bifurcation rules. Two techniques for vessel tree generation have been suggested in [182]. The first uses stochastic L-systems [163] in a two-stage process. First, bifurcation patterns are generated from the root toward the leaves. Then, geometric information about the lengths and radii of individual segments is added. The second technique includes simple physiological mechanisms in the generation of vascular systems. The metabolic activity of tissue is described by a scalar field modelling oxygen consumption and carbon dioxide production. Perfusion of O_2 and CO_2 through the arterial and venous system controls the respective gas concentrations. Based on this, biochemical transmitters are produced that either stimulate or inhibit the local growth of the system (see also [258]). Finally, an even more comprehensive model has been presented in [257]. The proposed modeling technique consists of two parts, the creation of a preliminary capillary plexus, and the modelling of vascular growth according to biophysical and hemodynamic rules. An extensive coverage of vessel generation methods is beyond the scope of this work. Nevertheless, a glimpse at current results in this domain is provided in Figure 5.2.

5.4 Outlook

The proposed solutions for the definition of the geometry, appearance, and biomechanical behavior of objects in a surgical scene have proven to be versatile tools for scene generation. However, the presented mechanisms need further work to reach the envisioned automatic scenario generation framework that can be controlled by an expert physician via clinically meaningful parameters without requiring specific knowledge about the underlying algorithms. As usual in scientific work, several new problems emerge with each suggested approach. The difficulties encountered as well as possible ways of addressing them are summarized below.

- The correspondences in the point distribution models are currently based on a subdivision process during the segmentation step. It is not clear how much error is introduced by this technique. The influence of small changes in the point correspondences on the validity and performance of the statistical framework has to be analyzed.
- The segmentation approach is dependent on user interaction. Optimally this step would be carried out in an automatic process; however, the quality of the radiological data is not sufficient for currently existing methods. Nevertheless, additional organ segmentations will be needed to extend the shape database.
- The instantiation of new healthy organ models is based on a predefined set of parameters, which were selected to match common gynecological measurements during routine inspections. Further data on patients are currently not included. The process of shape estimation could be extended to include input parameters such as age, weight, or parity. Nevertheless, this will probably result in a highly correlated parameter set, which would be difficult to control.
- The skeleton-based design of pathologies relies on the heuristic derivation of the tumor shape and does not include any physiological background. Moreover, interaction with the surrounding tissue during growth is not taken into account. This could be included by adding collision detection to the approach. For instance, after placing a tumor into the healthy anatomy, its shape could be adjusted to the cavity.
- A drawback of the cellular automata growth model is the voxel-based representation, which requires additional steps to merge the pathology with the uterus mesh geometry. A further problem is the handling of collisions during the growth process.
- A significant drawback of the particle system growth model is the nonintuitive tuning of the growth parameters. Only an experienced user with a thorough understanding of the underlying mechanisms will be able to control the algorithm. Even then, the outcome of the simulation is not easily predicted. In order to make the approach usable for clinical experts, the automatic setting of system parameters to roughly approximate a desired result would be beneficial. Finally, the significant computation time

renders the technique currently noninteractive. Parallelization could be a remedy for this problem.

- All pathology generation schemes can at the current stage only be evaluated qualitatively. A major problem is the limited resolution of imaging technology, which is usually not sufficient to clearly outline all sizes of tumors or myomas in the uterine cavity. As an alternative, images acquired during hysteroscopy could be analyzed to determine typical shapes and sizes of tumors.

- Both tumor growth models currently do not take angiogenic factors into account. The presented schemes should be paired with vessel generation methods to include the effect of blood supply on tumor growth. In turn, tumor genesis would also affect the development of the vascular trees.

- The proposed texture synthesis methods allow for the generation of a large variety of organ textures. However, computation time is directly linked to image resolution. Thus, if detailed textures are desired, several minutes of computation time are usually required, which hampers the interactivity of the process. Pregenerating a database with characteristic textures could alleviate some of these problems.

- Texture synthesis is currently not linked to organ geometry. It would be desirable to include information about geometrical features in the texture generation process, in order to allow variation of surface appearance matching a specific geometry. This also applies to texture adaptation due to the polyp growth simulations, for instance, taking into account collisions during tumor genesis.

- The identification of spring parameters via optimization techniques is computationally very expensive. Further problems are caused by lack of stability and local minima. It is currently questionable whether this technique can be successfully applied to larger meshes with reasonable computation times.

- Although the analytical derivation of tissue parameters yields promising results, it is not possible to derive a formulation for arbitrary meshes and materials. However, the presented approach is generally applicable and could be repeated for other materials and element types.

- The discussed MSM parameter estimation approaches represent only an initial attempt at providing a solution to this complex problem. An extension of the methods to more appropriate soft tissue deformation models has to be examined.

- As indicated above, the main components of the scene generation process are currently not fully integrated. Optimally, they would be combined into a single tool with an interface that does not require any detailed knowledge about the underlying algorithms. User interaction should be minimized as far as possible, and most of the required parameters should be determined based on a patient's clinical record.

References

1. J.A. Adam. A simplified mathematical model of tumor growth. *Mathematical Biosciences*, 81:229–244, 1986.
2. Maneesh Agrawala, Andrew C. Beers, and Marc Levoy. 3D painting on scanned surfaces. In *SI3D '95: Proceedings of the 1995 Symposium on Interactive 3D graphics*, pages 145–150, New York, NY, USA, 1995. ACM Press.
3. M. Agus, A. Giachetti, E. Gobbetti, G. Zanetti, N.W. John, and R.J. Stone. Mastoidectomy simulation with combined visual and haptic feedback. In *Medicine Meets Virtual Reality*, pages 17–23, 2002.
4. M. Alcaniz, V. Grau, C. Monserrat, C. Juan, and S. Albalat. A system for the simulation and planning of orthodontic treatment using a low cost 3D laser scanner for dental anatomy capturing. In *Proc. MMVR*, pages 8–14, 1999.
5. D. Ambrosi and F. Mollica. On the mechanics of a growing tumor. *International Journal of Engineering Science*, 40:1297–1316, 2002.
6. M. Amrani, F. Jaillet, M. Melkemi, and B. Shariat. Simulation of deformable organs with a hybrid approach. In *Tridimensional reconstruction*, pages 213–242, 2001.
7. R.P. Araujo and D.L.S. McElwain. A history of the study of solid tumour growth: The contribution of mathematical modelling. *Bulletin of Mathematical Biology*, 66:1039–1091, 2004.
8. M. Ashikhmin. Synthesizing natural textures. In *SI3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 217–226, 2001.
9. M.A. Audette, A. Fuchs, O. Astley, Y. Koseki, and K. Chinzei. Towards patient-specific anatomical model generation for finite element-based surgical simulation. In *Surgery Simulation and Soft Tissue Modeling (IS4TM)*, pages 340–352, 2003.
10. N.J Avis, F. Kleinermann, and J. McClure. Soft tissue surface-scanning - a comparison of commercial 3D object scanners for surgical simulation content creation and medical education applications. In *International Symposium on Medical Simulation (ISMS)*, pages 210–220, 2004.
11. D. Bachofen, J. Zatonyi, M. Harders, G. Szekely, P. Frueh, and M. Thaler. Enhancing the visual realism of hysteroscopy simulation. In *MMVR*, pages 31–36, 2006.

12. M. Bajka, M. Manestar, J. Hug, G. Szekely, U. Haller, and P. Groscurth. Detailed anatomy of the abdomen and pelvis of the visible human female. *Clinical Anatomy*, 17:252–260, 2004.

13. G.R. Baker, P.G. Norton, V. Flintoft, R. Blais, A. Brown, J. Cox, E. Etchells, W.A. Ghali, P. Hebert, S.R. Majumdar, M. O'Beirne, L. Palacois-Derflingher, R.J. Reid, S. Sheps, and R. Tamblyn. The canadian adverse events study: the incidence of adverse events among hospital patients in canada. *Canadian Medical Association Journal*, 170(11):1678–1686, 2004.

14. M.A. Gonzalez Ballester, A. Zisserman, and J. M. Brady. Measurement of brain structures based on statistical and geometrical 3D segmentation. In *Proc. MICCAI*, pages 499–509, 1998.

15. S. Bar-Meir. A new endoscopic simulator. *Endoscopy*, 32:898–900, 2000.

16. F. Barbagli, K. Salisbury, C. Ho, C. Spence, and H.Z. Tan. Haptic discrimination of force direction and the influence of visual information. *ACM Transactions on Applied Perception*, 3:125–135, 2006.

17. C. Basdogan, M. Sedef, M. Harders, and S. Wesarg. Virtual reality supported simulators for training in minimally invasive surgery. *IEEE Computer Graphics and Applications*, 27(2):54–66, 2007.

18. K.J. Bathe. *Finite Element Procedures*. Prentice Hall, 1996.

19. C. Baur, D. Guzzoni, and O. Georg. Virgy: A virtual reality and force feedback based endoscopy surgery simulator. In *Proc. MMVR*, pages 110–116, 1998.

20. Bill Baxter, Vincent Scheib, Ming C. Lin, and Dinesh Manocha. DAB: interactive haptic painting with 3D virtual brushes. In *SIGGRAPH'01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 461–468, New York, NY, USA, 2001. ACM Press.

21. T. Belytschko, V. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139:3–47, 1996.

22. T. Belytschko and S. Ong. Hourglass control in linear and nonlinear problems. *Computer Methods in Applied Mechanics and Engineering*, 43:251–276, 1984.

23. J.R. Bergen and E.H. Andelson. Early vision and texture perception. *Nature*, 333:363–367, 1988.

24. J.E. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, 36:192–326, 1974.

25. J.E. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, 48(3):259–302, 1986.

26. G. Bianchi, M. Harders, and G. Szekely. Mesh topology identification for mass-spring models. In *MICCAI'03*, volume 1, pages 50–58, 2003.

27. G. Bianchi, B. Solenthaler, G. Szekely, and M. Harders. Simultaneous topology and stiffness identification for mass-spring models based on FEM reference deformations. In *MICCAI'04*, pages 293–301, 2004.

28. James F. Blinn. Simulation of wrinkled surfaces. In *SIGGRAPH'78: Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, pages 286–292, New York, NY, USA, 1978. ACM Press.

29. H. Blum. A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form*, pages 362–380, 1967.

30. P. Boesiger. *Kernspin-Tomographie für die medizinische Diagnostik*. Teubner, 1985.

31. J.S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proc. SIGGRAPH 97*, pages 361–368, 1997.

32. M.L. Braunstein. *Depth Perception through Motion.* Academic Press, New York, 1976.

33. C.A. Brebbia. *The Boundary Element Method for Engineers.* Pentech Press, 1978.

34. C. Brechbuehler, G. Gerig, and O. Kuebler. Parametrization of closed surfaces for 3-D shape description. *Computer Vision and Image Understanding*, 61:154–170, 1995.

35. M. Bro-Nielsen and S. Cotin. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. *Computer Graphics Forum*, 15(3):57–66, 1996.

36. M. Bro-Nielsen, J.L. Tasto, R. Cunningham, and G.L. Merril. Preop endoscopic simulator: A PC-based immersive training system for bronchoscopy. In *Proc. MMVR*, pages 76–82, 1999.

37. F.P. Brooks. What's real about virtual reality? *IEEE Computer Graphics & Applications*, 19(6)):16–27, 1999.

38. I. Brouwer, J. Ustin, L. Bentley, A. Sherman, N. Dhruv, and F. Tendick. Measuring in vivo animal soft tissue properties for haptic modeling in surgical simulation. In *Proc. Medicine Meets Virtual Reality*, pages 69–74, 2001.

39. J.D. Brown, J. Rosen, Y.S. Kim, L. Chang, M. Sinanan, and B. Hannaford. In-vivo and in-situ compressive properties of porcine abdominal soft tissues. In *MMVR 11*, pages 26–32, 2003.

40. J. Bryan, D. Stredney, G. Wiet, and D. Sessanna. Virtual temporal bone dissection: a case study. In *Visualization (VIS'01)*, pages 497–500, 2001.

41. J. Buerki-Cohen, N. Soja, and T. Longridge. Simulator platform motion - the need revisited. *Intl. Journal of Aviation Psychology*, 8(3):293–317, 1998.

42. ACOG Tech. Bulletin. Hysteroscopy. International Journal of Gynecology & Obstetrics, May 1994. 45(2):175–180.

43. G. Burdea and P. Coiffet. *Virtual Reality Technology.* John Wiley & Sons, 1993.

44. A.C. Burton. Rate of growth of solid tumours as a problem of diffusion. *Growth*, 30:157–176, 1966.

45. T.M. Caelli and B. Julesz. Experiments in the visual perception of texture. *Biological Cybernetics*, 28(3):167–175, 1978.

46. O. Camara, O. Colliot, and I. Bloch. Computational modeling of thoracic and abdominal anatomy using spatial relationships for image segmentation. *Real-Time Imaging*, 10(4):263–273, 2004.

47. B.M. Cameron and A. Manduca R.A. Robb. Patient-specific anatomic models. geometric surface generation from three-dimensional medical images using a specified polygonal budget. In *Stud Health Technol Inform (MMVR)*, pages 447–460, 1996.

48. F.J. Carter, T.G. Frank, P.J. Davies, D. McLean, and A. Cuschieri. Measurement and modelling of the compliance of human and porcine organs. *Medical Image Analysis*, 5(4):231–236, 2001.

49. E.E. Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces.* PhD thesis, Department of Computer Science, Univ. of Utah, 1974.

50. A. Caunce and C.J. Taylor. 3D point distribution models of the cortical sulci. In *Proc. of ICCV*, pages 402–407, 1998.

51. M. A. J. Chaplain. Avascular growth, angiogenesis and vascular growth in solid tumours: the mathematical modeling of the stages of tumour development. *Mathematical and Computer Modelling*, 23:47–87, 1996.

52. S.-W. Cheng and T.K. Dey. Quality meshing with weighted delaunay refinement. In *Proc. 13th ACM-SIAM Symposium on Discrete Algorithms*, pages 137–146, 2002.

53. L.P. Chew. Guaranteed-quality delaunay meshing in 3D. In *13th ACM Symposium on Computational Geometry*, pages 391–393, 1997.

54. V. Chopra. Anaesthesia simulators. *Bailliere's Clinical Anaesthesiology*, 10(2):297–315, 1996.

55. CIRS. Critical incidence reporting system. Webpage: www.cirsmedical.ch, 1998. visited Aug 2005.

56. O. Clatz, M. Sermesant, P.-Y. Bondiau, H. Delingette, S. Warfield, G. Malandain, and N. Ayache. Realistic simulation of the 3D growth of brain tumors in mr images coupling diffusion with mass effect. *IEEE Transactions on Medical Imaging*, 24(10):1334–1346, 2005.

57. T.F. Cootes and C.J. Taylor. Active shape models - smart snakes. In *Proc. British Machine Vision Conf.*, pages 266–275, 1992.

58. I.F. Costa and R. Balaniuk. Lem-an approach for real time physically based soft tissue simulation. In *International Conference on Robotics and Automation*, volume 3, pages 2337–2343, 2001.

59. S. Cotin, H. Delingette, and N. Ayache. A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. *The Visual Computer*, 16(8):437–452, 2000.

60. G. Cottet and P. Koumoutsakos. *Vortex Methods - Theory and Practice*. Cambridge University Press, 2000.

61. CPSI. Canadian patient safety institute - current state report on patient simulation in Canada, April 2005.

62. G.R. Cross and A.K. Jain. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(1):25–39, 1983.

63. S. Czanner, R. Durikovic, and H. Inoue. Growth simulation of human embryo brain. In *Proc. of Spring Conference on Computer Graphics*, pages 139–145, 2001.

64. J.G. Daugman. Unvertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of Optical Society of America*, 2(7):1160–1169, 1985.

65. P.J. Davies, F.J. Carter, and A. Cuschieri. Mathematical modelling for keyhole surgery simulations: a biomechanical model for spleen tissue. *Journal of Applied Mathematics*, 67(1):41–67, 2002.

66. R.H. Davies, C.J. Twining, T.F. Cootes, J.C. Waterton, and C.J. Taylor. 3D statistical shape models using direct optimisation of description length. In *European Conference on Computer Vision (ECCV)*, volume 3, pages 3–21, 2002.

67. S. De and K.J. Bathe. The method of finite spheres. *Computational Mechanics*, 25:329–345, 2000.

68. S. De, J. Kim, and M.A. Srinivasan. A meshless numerical technique for physically based real time medical simulations. In *Proc. MMVR*, pages 113–118, 2001.

69. S. De, Y.-J. Lim, M. Manivannan, and M.A. Srinivasan. Physically realistic virtual surgery using the point-associated finite field (paff) approach. *Presence: Teleoperators & Virtual Environments*, 15(3):294–308, 2006.

70. C. de Boor. *A Practical Guide to Splines*. Springer Verlag, New York, 1978.

71. B. Delaunay. Sur la sphere vide. *Otdelenie Matematicheskii i Estestvennyka Nauk*, 7:793–800, 1934.

72. H. Delingette. General object reconstruction based on simplex meshes. *International Journal of Computer Vision*, 2:111–146, 1999.

73. Oliver Deussen, L. Kobbelt, and P. Tuecke. Using simulated annealing to obtain good nodal approximations of deformable objects. In *Comp. Anim. and Simul.*, pages 30–43, 1995.

74. Q. Du and D. Wang. Tetrahedral mesh generation and optimization based on centroidal voronoi tesselations. *Intl. Journal for Numberical Methods in Engineering*, 56:1355–1373, 2003.

75. P. Dubois, P. Meseure, F. Peugnet, and J.F. Rouland. Training simulator for retinal laser photocoagulation: a new approach for surgeons apprenticeships. In *SPIE Conference on Surgical-Assist Systems*, pages 54–62, 1998.

76. R. Durikovic. Growth simulation of digestive system using function representation and skeleton dynamics. *International Journal on Shape Modeling*, 10(1):31–49, 2004.

77. R. Durikovic, K. Kaneda, and H. Yamashita. Animation of biological organ growth based on l-systems. *Computer Graphics Forum*, 17(3):1–13, 1998.

78. M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. *Computer Graphics*, 29:173–182, 1995.

79. A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *International Conference on Computer Vision*, volume 2, pages 1033–1038, September 1999.

80. A.A. Efros and W.T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, 2001.

81. R.D. Eisler, A.K. Chatterjee, and G.H. Burghart. Simulation and modeling of penetrating wounds from small arms. In *Proc. of MMVR*, pages 511–522, 1996.

82. M.A. ElHelw, B.P.L. Lo, A.J. Chung, A. Darzi, and G.-Z. Yang. Photorealistic rendering of large tissue deformation for surgical simulation. In *Proc. of MICCAI*, volume 2, pages 355–362, 2004.

83. M. Farshad, M. Barbezat, P. Flüeler, F. Schmidlin, and P. Graber P. Niederer. Material characterization of the pig kidney in relation with the biomechanical analysis of renal trauma. *Journal of Biomechanics*, 32:417–425, 1999.

84. Kurt W. Fleischer, David H. Laidlaw, Bena L. Currin, and Alan H. Barr. Cellular texture generation. In *SIGGRAPH'95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 239–248, New York, NY, USA, 1995. ACM Press.

85. M.S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14:231–250, 1997.

86. M.S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. In M.S. Floater and M.A. Sabin, editors, *In Advances in Multiresolution for Geometric Modelling*, pages 259–284. Springer Verlag, 2004.

87. J.B. Fowlkes, S.Y. Emelianov, J.G. Pipe, A.R. Skovoroda, R.S. Adler, P.L. Carson, and A.P. Sarvazyan. Magnetic resonance imaging techniques for detection of elasticity variation. *Medical Physics*, 22(11):1771–1778, 1995.

88. M. Froumentin and E. Varlet. Dynamic implicit surface tesselation. In *Virtual Reality Software and Technology*, pages 79–86, 1997.

89. Y.C. Fung. *Biomechanics: mechanical properties of living tissues*. Springer-Verlag, 1993.

90. D.M. Gaba and A. DeAnda. A comprehensive anesthesia simulation environment: re-creating the operating room for research and training. *Anesthesiology*, 69(3):387–394, 1988.

91. D. Gabor. Theory of communication. *Journal of IEE*, 93:429–459, 1946.

92. A. Gagalowicz and S.D. Ma. Sequential synthesis of natural textures. *Computer Vision, Graphics and Image Processing*, 30(3):289–315, 1985.

93. A.G. Gallagher and C.U. Cates. Approval of virtual reality training for carotid stenting. *Journal of the American Medical Association*, 292(24):3024–3026, 2004.

94. Geoffrey Y. Gardner. Simulation of natural scenes using textured quadric surfaces. In *SIGGRAPH'84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 11–20, New York, NY, USA, 1984. ACM Press.

95. M. Garland and P. Heckbert. Surface simplification using quadric error metrics. In *ACM SIGGRAPH*, pages 206–216, 1997.

96. C.F. Gauss. Disquisitiones generales circa superficies curva, 1828.

97. S. Geman and D. Geman. Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

98. P.L. George, F. Hecht, and E. Saltel. Automatic mesh generator with specified boundary. *Computer Methods in Applied Mechanics and Engineering*, 92:269–288, 1991.

99. M. Gervautz and W. Purgathofer. A simple method for color quantization: Octree quantization, 1990.

100. D. Ghazanfarpour and J. M. Dischler. Spectral analysis for automatic 3-D texture generation. In *Computers and Graphics*, pages 413–422, 1995.

101. D. Ghazanfarpour and J. M. Dischler. Generation of 3-D texture using multiple 2-D models analysis. In *Eurographics*, pages 311–323, 1996.

102. S. Gibson, C. Fyock, E. Grimson, T. Kanade, R. Kikinis, H. Lauer, N. McKenzie, A. Mor, S. Nakajima, H. Ohkami, R. Osberne, J. Samosky, and A. Sawada. Simulating arthroscopic knee surgery using volumetric object representations, real-time volume rendering, and haptic feedback. *Medical Image Analysis*, 2(2):121–132, 1998.

103. R. Gingold and J. Monaghan. Smoothed particle hydrodynamics - theory and application to nonspherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375, 1977.

104. E. Gladilin, S. Zachow, P. Deuflhard, and H.C. Hege. A biomechanical model for soft tissue simulation in craniofacial surgery. In *Medical Imaging and Augmented Reality (MIAR)*, pages 137–141, 2001.

105. P.E. Green, T.A. Piantanida, J.W. Hill, I.B. Simon, and R.M. Satava. Telepresence: Dexterous procedures in a virtual operating field. *American Surgeon*, 57:192, 1991.

106. Ned Greene. Environment mapping and other applications of world projections. *IEEE Computer Graphics and Applications*, 6(11):21–29, 1986.

107. L. Greengard and V. Roklin. A fast algorithm for particles simulations. *Journal of Computational Physics*, 73:325–348, 1987.

108. H.P. Greenspan. Models for the growth of a solid tumor by diffusion. *Studies in Applied Mathematics*, 52:317–340, 1972.

109. S. Habib, C. Molina-Paris, and T.S. Deisboeck. Complex dynamics of tumors: modeling an emerging brain tumor system using a set of coupled reaction-diffusion equations. *Physica A.*, 327:501–524, 2003.

110. J. Hahn, R. Kaufman, A. Winick, T. Carleton, Y. Park, O.R. Lindeman, N. Al-Ghreimil, R. Walsh, M. Loew, J. Gerber, and S. Sankar. Training environment for inferior vena caval filter placement. In *Proc. MMVR*, pages 291–297, 1998.

111. J.V. Hajnal, D.L. Hill, and D.J. Hawkes. *Medical Image Registration*. CRC Press, 2001.

112. K. Hale and K. Stanney. Deriving haptic design guidelines from human physiological, psychophysical, and neurological foundations. *IEEE Computer Graphics and Applications*, 24(2):33–39, 2004.

113. Pat Hanrahan and Paul Haeberli. Direct WYSIWYG painting and texturing on 3D shapes. In *SIGGRAPH'90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 215–223, New York, NY, USA, 1990. ACM Press.

114. M. Harders, M. Bajka, U. Spaelter, S. Tuchschmid, and G. Szekely. Highly-realistic, immersive training environment for hysteroscopy. In *Proc. of Medicine Meets Virtual Reality*, pages 176–181, 2006.

115. M. Harders, D. Steinemann, M. Gross, and G. Szekely. A hybrid cutting approach for hysteroscopy simulation. In *Proc. MICCAI*, pages 567–574, 2005.

116. P.S. Heckbert. Survey of texture mapping. *IEEE Computer Graphics and Applications*, 6(11):56–67, 1986.

117. D.J. Heeger and J.R. Bergen. Pyramid-based texture analysis/synthesis. In *Proc. ACM SIGGRAPH)*, volume 29, pages 229–238, 1995.

118. B. Heidelberger, M. Teschner, and M. Gross. Detection of collisions and self-collisions using image-space techniques. In *Proc. WSCG'04*, pages 145–152, 2004.

119. P.A. Heng, C.Y. Cheng, T.T. Wong, Y. Xu, Y.P. Chui, K.M. Chan, and S.K. Tso. A virtual-reality training system for knee arthroscopic surgery. *IEEE Transactions on Information Technology in Biomedicine*, 8(2):217–227, 2004.

120. A.V. Hill. The diffusion of oxygen and lactic acid through tissues. *Proceedings of the Royal Society B*, 104:39–96, 1928.

121. K. Hormann and G. Greiner. MIPS: an efficient global parametrization method, curve and surface design. In *Curve and Surface Design: Saint-Malo*, pages 153–162, 2000.

122. D.H. Hubel and T.N. Wiesel. Functional architecture of macaque monkey visual cortex. *Proceedings of the Royal Society of London, B*, 198:1–59, 1977.

123. J. Hug, C. Brechbuehler, and G. Szekely. Model-based initialisation for segmentation. In *Proc. of ECCV, Part II*, pages 290–306, 2000.

124. R. Hutter, P. Hora, and P. Niederer. Total hourglass control for hyperelastic materials. *Journal of Computer Methods in Applied Mechanics and Engineering*, 189(3):991–1010, 2000.

125. I. Ionescu, J. Guilkey, M. Berzins, R.M. Kirby, and J. Weiss. Computational simulation of penetrating trauma in biological soft tissues using the material point method. In *Proc. of MMVR*, pages 213–218, 2005.

126. A.-C. Jambon, P. Dubois, and S. Karpf. A low-cost training simulator for initial formation in gynecologic laparoscopy. In *CVRMed*, pages 347–356, 1997.

127. D. James and D.K. Pai. A united treatment of elastostatic and rigid contact simulation for real time haptics. *Haptics-e, the Electronic Journal of Haptics Research*, 2(1), 2001.

154   References

128. D.L. James and D.K. Pai. Artdefo - accurate real time deformable objects. In *Siggraph 1999, Computer Graphics Proceedings*, pages 65–72, 1999.
129. W. Jin, Y.-J. Lim, X. Xu, T.P. Singh, and S. De. Improving the visual realism of virtual surgery. In *Proceedings of MMVR 13*, pages 227–233, 2005.
130. David Johnson, Thomas V. Thompson Ii, Matthew Kaplan, Donald Nelson, and Elaine Cohen. Painting textures with a haptic interface. In *VR '99: Proceedings of the IEEE Virtual Reality*, pages 282–285, Washington, DC, USA, 1999. IEEE Computer Society.
131. A. Joukhadar, F. Garat, and Ch. Laugier. Parameter identification for dynamic simulation. In *In Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1928–1933, 1997.
132. B. Julesz. Visual Pattern Discrimination. In *IRE Trans. Inform. Theory*, volume IT-8, pages 84–92, 1962.
133. B. Julesz, E.N. Gilbert, L.A. Shepp, and H.L. Frisch. Inability of humans to discriminate between visual textures that agree in second-order statistics - revisited. *Perception*, 2:391–405, 1973.
134. B. Julesz, E.N. Gilbert, and J.D. Victor. Visual discrimination of textures with identical third-order statistics. *Biological Cybernetics*, 31(3):137–140, 1978.
135. G.A. Kalberer and L. Van Gool. Realistic face animation for speech. *Journal of Visualization and Computer Animation*, 13:97–106, 2002.
136. S. Kang. A survey of image-based rendering techniques. In *Proc. SPIE*, volume 3641, pages 2–16, 1999.
137. A.R. Kansal, S. Torquato, G.R. Harsh, E.A. Chiocca, and T.S. Deisboeck. Simulated brain tumor growth dynamics using a three-dimensional cellular automaton. *Journal of Theoretical Biology*, 203(4):367–382, 2000.
138. M. Kauer. *Inverse Finite Element Characterization of Soft Tissues with Aspiration Experiments*. PhD thesis, ETH Zurich, 2001.
139. M. Kauer, V. Vuskovic, J. Dual, G. Szekely, and M. Bajka. Inverse finite element characterization of soft tissues. *Medical Image Analysis*, 6:275–287, 2002.
140. A. Kelemen, G. Szekely, and G. Gerig. Elastic model-based segmentation of 3-D neuroradiological data sets. *IEEE Transactions on Medical Imaging*, 18(10):828–839, 1999.
141. A.E. Kerdok, M.P. Ottensmeyer, and R.D. Howe. Effects of perfusion on the viscoelastic characteristics of liver. *Journal of Biomechanics*, 39(12):2221–2231, 2006.
142. J. Kim, B.K. Tay, N. Stylopoulos, D.W. Rattner, and M.A. Srinivasan. Characterization of intra-abdominal tissues from in vivo animal experiments for surgical simulation. In *MICCAI*, pages 206–213, 2003.
143. KISTI. Korea institute of science and technology information - visible korean human. `http://vkh3.kisti.re.kr/`, 2001.
144. D. Knapp, J.P. Kerr, and M. Sellberg. Patient specific color texture mapping of ct-based anatomical surface models utilizing cryosection data. In *Medicine Meets Virtual Reality*, pages 608–617. IOS Press, 1997.
145. L.T. Kohn, J.M. Corrigan, and M.S. Donaldson. To err is human: Building a safer health system. National Academy Press, 1999.
146. A.C.W. Kotcheff and C.J. Taylor. Automatic construction of eigenshape models by direct optimization. *Medical Image Analysis*, 2(4):303–314, 1998.
147. U. Kuehnapfel, H.K. Cakmak, and H. Maass. 3D modeling for endoscopic surgery. In *Proc. IEEE Symposium on Simulation*, pages 22–32, 1999.

148. U. Kuehnapfel, H. Krumm, C. Kuhn, M. Huebner, and B. Neisius. Endosurgery simulations with KISMET: A flexible tool for surgical instrument design, operation room planning and vr technology based abdominal surgery training. In *Proc. Virtual reality World*, pages 165–171, 1995.

149. T. Kuroda, N. Motohashi, R. Tominaga, and K. Iwata. Three-dimensional dental cast analyzing system using laser scanning. *American Journal of Orthodontics and Dentofacial Orthopedics*, 110:365–369, 1996.

150. V. Kwatra, A. Schoeld, I. Esse, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Transactions on Graphics*, 22(3):277–286, 2003.

151. S.K. Kyriacou, C. Davatzikos, S.J. Zinreich, and R.N. Bryan. Nonlinear elastic registration of brain images with tumor pathology using a biomechanical model. *IEEE Transactions on Medical Imaging*, 18(7):580–592, 1999.

152. D. Laurendeau, L. Guimond, and D. Poussart. A computer-vision technique for the acquisition of 3D profiles of dental imprints and the detection of teeth in the profiles: An application to orthodontics. *IEEE Transactions on Medical Imaging*, 10(3):453–461, 1991.

153. L.L. Leape. The preventabolity of medical injury. Erlbaum Prublications, 1994. pp. 13–25.

154. V. Leeb, A. Radetzky, and L.M. Auer. Interactive texturing by polyhedron decomposition. In *IEEE Proceedings on Virtual Reality*, pages 165–171, 2001.

155. B.P.F. Lelieveldt, M. Sonka, L. Bolinger, T.D. Scholz, H. Kayser, R. van der Geest, and J.H.C. Reiber. *Anatomical Modeling with Fuzzy Implicit Surface Templates: Application to Automated Localization of the Heart and Lungs in Thoracic MR Volumes*, chapter 5, pages 73–91. CVIU. Elsevier, 2000.

156. J.E. Lennard-Jones. Cohesion. *Proceedings of the Physical Society*, 43:461–482, 1931.

157. P. Leskovsky, M. Harders, and G. Szekely. A web-based repository of surgical simulator projects. In *Proc. of Medicine Meets Virtual Reality*, pages 311–315, 2006.

158. B. Levy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 362–371, 2002.

159. J. P. Lewis. Algorithms for solid noise synthesis. In *SIGGRAPH'89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 263–270, New York, NY, USA, 1989. ACM Press.

160. John-Peter Lewis. Texture synthesis for digital painting. In *SIGGRAPH'84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 245–252, New York, NY, USA, 1984. ACM Press.

161. Lin Liang, Ce Liu, Ying-Qing Xu, Baining Guo, and Heung-Yeung Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics*, 20(3):127–150, 2001.

162. W. Lin and R.A. Robb. Dynamic volume texture mapping and model deformation for visually realistic surgical simulation. In *Proceedings of Medicine Meets Virtual Reality*, pages 198–204. IOS Press, 1999.

163. A. Lindenmayer. Mathematical models for cellular interaction in development. *Journal of Theoretical Biology*, 18:280–315, 1968.

164. L.A. Liotta, G.M. Saidel, and J. Kleinerman. Stochastic model of metastases formation. *Biometrics*, pages 535–550, 1976.

165. A. Liu, F. Tendick, K. Cleary, and C. Kaufmann. A survey of surgical simulation: Applications, technology, and education. *Presence: Teleoperators & Virtual Environments*, 12(6):599–614, 2003.

166. G.R. Liu and M.B. Liu. *Smoothed particle hydrodynamics: a meshfree particle method*. World Scientific, 2003.

167. B. Lloyd, G. Szekely, and M. Harders. Identification of spring parameters for deformable object simulation. *IEEE Transactions on Visualization and Computer Graphics*, 13(5), 2007.

168. S. H. Lo. Volume discretization into tetrahedra-i. verification and orientation of boundary surface. *Computers and Structures*, 39(5):493–500, 1991.

169. R. Lohner. Progress in grid generation via the advancing front technique. *Engineering with Computers*, 12:186–210, 1996.

170. W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Proceedings of SIGGRAPH*, 21(4):163–169, 1987.

171. J. Louchet, X. Provot, and D. Crochemore. Evolutionary identification of cloth animation models. In *Computer Animation and Simulation '95*, pages 44–54. Springer Verlag, 1995.

172. L. Lucy. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, 82:1013, 1977.

173. S.D. Ma and A. Gagalowicz. Determination of local coordinate systems for texture synthesis on 3-D surfaces. In *Eurographics*, pages 109–118, 1985.

174. A. Maciel, R. Boulic, and D. Thalmann. Deformable tissue parameterized by properties of real biological tissue. In *IS4TM*, pages 74–87, 2003.

175. F. Maes, A. Collignon, P. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 16(2):187–198, 1997.

176. Sebastian Magda and David Kriegman. Fast texture synthesis on arbitrary meshes. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering*, pages 82–89, 2003.

177. M.A.B. Makeham, S.M. Dovey, M. County, and M.R. Kidd. An international taxonomy for errors in general practice: a pilot study. *The Medical Journal of Australia*, 177(2):68–72, 2002.

178. J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America*, 5:923–932, 1990.

179. S. Marcelja. Mathematical description of the responses of simple cortical cells. *Journal of Optical Society of America*, 70(11):1297–1300, 1980.

180. J. Marescaux, J.M. Clement, V. Tassetti, C. Koehl, S. Cotin, Y. Russier, D. Mutter, H. Delingette, and N. Ayache. Virtual reality applied to hepatic surgery simulation: the next revolution. *Annals of Surgery*, 228(5):627–634, 1998.

181. D.L.S. McElwain and G. J. Pettet. Cell migration in multicell spheroids: swimming against the tide. *Bulletin of Mathematical Biology*, 55:655–674, 1993.

182. V. Meier. *Realistic visualization of abdominal organs and its application in laparoscopic surgery simulation*. PhD thesis, ETH Zurich, 1999.

183. L. Mencaglia and E. Hamou. Manual of gynecological hysteroscopy - diagnosis and surgery. Endo-Press, Germany, 2001.

184. M.Gardner. The fantastic combinations of john conways's new solitaire game of life. *Scientific American*, 223(4), 1970.

185. Z. Michlewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, 1999.

186. C. Monserrat, U. Meier, M. Alcaniz, F. Chinesta, and M.C. Juan. A new approach for the real-time simulation of tissue deformations in surgery simulation. *Comput Methods Programs Biomed*, 64(2):77–85, 2001.

187. K. Montgomery, L.-R. Heinrichs, C. Bruyns, S. Wildermuth, C. Hasser, S. Ozenne, and D. Bailey. Surgical simulator for hysteroscopy: A case study of visualization in surgical training. In *IEEE Visualization*, 2001.

188. A. Mota, W.S. Klug, M. Ortiz, and A. Pandolfi. Finite-element simulation of firearm injury to the human cranium. *Computational Mechanics*, 31:115–121, 2003.

189. P. Mueller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool. Procedural modeling of buildings. *ACM Transactions on Graphics*, 25(3):614–623, 2006.

190. R. Muthupillai, D.J. Lomas, P.J. Rossman, J.F. Greenleaf, A. Manduca, and R.L. Ehman. Magnetic resonance elastography by direct visualization of propagating acoustic strain waves. *Science*, 269(5232):1854–1857, 1995.

191. A. Nava, E. Mazza, F. Kleinermann, N.J. Avis, and J. McClure. Evaluation of the mechanical properties of human liver and kidney through aspiration experiments. *Technology and Health Care*, 12:269–280, 2004.

192. Andrew Nealen and Marc Alexa. Hybrid texture synthesis. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering*, pages 97–105, 2003.

193. F. Neyret, R. Heiss, and F. Senegas. Realistic rendering of an organ surface in real-time for laparoscopic surgery simulation. *The Visual Computer*, 18(3):135–149, 2002.

194. Fabrice Neyret and Marie-Paule Cani. Pattern-based texturing revisited. In *SIGGRAPH'99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 235–242, 1999.

195. NLM. National library of medicine - visible human project. `http://www.nlm.nih.gov/research/visible/`, 1994.

196. Justin O'Brien and Alan Johnston. When texture is a stronger depth cue than motion. In *Depth perception, Applied Vision Association Meeting*, 1997.

197. G. Ollenschlaeger. Medizinische risiken, fehler und patientensicherheit - zur situation in deutschland. Schweizerische Aerztezeitung, 2001. pp. 1404–1410.

198. J. Ophir, I. Cespedes, H. Ponnekanti, Y. Yazdi, and X. Li. Elastography: a method for imaging the elasticity of biological tissues. *Ultrasonic Imaging*, 13(2):111–134, 1991.

199. D. O'Sullivan. Exploring spatial process dynamics using irregular cellular automaton models. *Geographical analysis*, 33(1):1–18, 2001.

200. M.P. Ottensmeyer. In vivo measurement of solid organ visco-elastic properties. In *MMVR 10*, pages 328–333, 2002.

201. M.P. Ottensmeyer, A.E. Kerdok, R.D. Howe, and S. Dawson. The effects of testing environment on the viscoelastic properties of soft tissues. In *ISMS*, pages 9–18, 2004.

202. S.J. Owen. A survey of unstructured mesh generation technology. In *Proc. 7th International Meshing Roundtable*, pages 239–267, 1998.

203. R. Paget, M. Harders, and G. Szekely. A framework for coherent texturing in surgical simulators. In *Pacific Graphics*, pages 112–114, 2005.

204. R. Paget and D. Longstaff. Texture synthesis via a noncausal nonparametric multiscale Markov random field. *IEEE Transactions on Image Processing*, 7(6):925–931, 1998.

205. A.A. Patel, E.T. Gawlinski, S.K. Lemieux, and R.A. Gatenby. A cellular automaton model of early tumor growth and invasion. *Journal of Theoretical Biology*, 213(3):315–331, 2001.
206. Darwyn R. Peachey. Solid texturing of complex surfaces. In *SIGGRAPH'85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 279–286, New York, NY, USA, 1985. ACM Press.
207. K. Perlin and E. M. Hoffert. Hypertexture. In *SIGGRAPH'89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 253–262, New York, NY, USA, 1989. ACM Press.
208. Ken Perlin. An image synthesizer. In *SIGGRAPH'85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 287–296, New York, NY, USA, 1985. ACM Press.
209. J.C. Petrozza. Hysteroscopy. `http://www.emedicine.com/med/topic3314.html`, July 2004.
210. H. Pfister, M. Zwicker, J. van Baar, and M. Gross. Surfels: Surface elements as rendering primitives. In *SIGGRAPH Proceedings*, pages 335–342, 2000.
211. B. Pflesser, A. Petersik, U. Tiede, K. Hohne, and R. Leuwer. Volume cutting for virtual petrous bone surgery. *Computer Aided Surgery*, 7(2):74–83, 2002.
212. G. Picinbono, H. Delingette, and N. Ayache. Real-time large displacement elasticity for surgery simulation: Non-linear tensor-mass model. In *MICCAI*, pages 643–652, 2000.
213. G. Picinbono, H. Delingette, and N. Ayache. Non-linear and anisotropic elastic soft tissue models for medical simulation. In *ICRA2001: IEEE International Conference Robotics and Automation*, volume 2, pages 1370–1375, 2001.
214. S.B. Pinion, D.E. Parkin, D.R. Abramovich, A. Naji, D.A. Alexander, and I.T. Russel amd H.C. Ktichener. Randomised trial of hysterectomy, endometrial laser ablation, and transcervical endometrial resection for dysfunctional uterine bleeding. *British Medical Journal*, 309:979–983, October 1994.
215. C.P. Please, G.J. Pettet, and D.L.S. McElwain. A new approach to modelling the formation of necrotic regions in tumours. *Applied Mathematics Letters*, 11:89–94, 1998.
216. K. Popat and R. Picard. Novel cluster-based probability model for texture synthesis, classification, and compression. In *In Visual Communications and image Processing*, pages 756–768, 1993.
217. J. Portilla and E.P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *Intl. Journal of Computer Vision*, 40(1):49–71, 2000.
218. C.E. Prakash, J. Kim, M. Manivannan, and M.A. Srinivasan. A new approach for the synthesis of glistening effect in deformable anatomical objects displayed with haptic feedback. In *Proceedings of MMVR 10*, pages 369–375, 2002.
219. Emil Praun, Adam Finkelstein, and Hugues Hoppe. Lapped textures. In *SIGGRAPH'00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 465–470, 2000.
220. A.S. Qi, X. Zheng, C.Y. Du, and B.S.An. A cellular automaton model of cancerous growth. *Journal of Theoretical Biology*, 161(1):1–12, 1993.
221. M. Quicken, Ch. Brechbuehler, J. Hug, H. Blattmann, and G. Szekely. Parameterization of closed surfaces for parametric surface description. In *IEEE CVPR*, volume 1, pages 354–360, 2000.

222. A. Radetzky, A. Nürnberger, and P. Pretschner. Elastodynamic shape modeler: A tool for defining the deformation behavior of virtual tissues. *RadioGraphics*, 20(1):865–881, 2000.

223. W.T. Reeves. Particle systems - a technique for modeling a class of fuzzy objects. In *Proc. SIGGRAPH*, volume 17, pages 359–376, 1983.

224. K.D. Reinig, C.G. Rush, H.L. Pelster, V.M. Spitzer, and J.A. Heath. Real-time visually and haptically accurate surgical simulation. In *Proc. MMVR*, pages 542–545, 1996.

225. C.W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *Proc. SIGGRAPH*, volume 21, pages 25–34, 1987.

226. A. Rhomberg, C. Brechbühler, G. Székely, and G. Tröster. A parallel architecture for interactive FEM computation in a surgery simulator. In *Proc. Parallel Computing Conference (ParCo99)*, pages 225–232, 1999.

227. P.V. Sander, S.J. Gortler, J. Snyder, and H. Hoppe. Signal-specialized parametrization. In *Eurographics Workshop on Rendering*, pages 87–98, 2002.

228. R.M. Satava. Virtual reality surgical simulator: The first steps. *Surg Endosc*, 7:203–205, 1993.

229. Bruce Schacter. Long crested wave models. *Computer Graphics and Image Processing*, 12(2):187–201, 1980.

230. T. Schiemann, J. Freudenberg, B. Plessner, A. Pommert, K. Priesmeyer, M. Riemer, R. Schubert, U. Tiede, and K.-H. Hoehne. Exploring the visible human using the voxel-man framework. In *Comput. Med. Imaging Graph.*, pages 127–132, 2000.

231. M. Schill, C. Wagner, M. Hennen, H. Bender, and R. Maenner. Eyesi - a simulator for intra-ocular surgery. In *Proc. MICCAI*, pages 1166–1174, 1999.

232. W. Schreiner and P.F. Buxbaum. Computer optimization of vascular trees. *IEEE Transactions on Biomedical Engineering*, 40:482–491, 1993.

233. M. Sellberg, D. Murray, D. Knapp, T. Teske, K. Lattie, and M. Vanderploeg. Virtual Human: An Automated Virtual Environment for Computer-Aided Instruction and Biomechanical Analysis. In *Interactive Technology and the New Paradigm for Healthcare*, pages 340–348, 1995.

234. A. Sheffer and E. de Sturler. Smoothing an overlay grid to minimize linear distortion in texture mapping. *ACM Transactions on Graphics*, 21(4):874–890, 2002.

235. A. Sheffer and J.C. Hart. Seamster: inconspicuous low-distortion texture seam layout. In *VIS '02: Proceedings of the conference on Visualization '02*, pages 291–298, 2002.

236. A. Sheffer, B. Levy, M. Mogilnitsky, and A. Bogomyakov. ABF++ : Fast and Robust Angle Based Flattening. *ACM Transactions on Graphics*, 24(2):311–333, 2005.

237. M.S. Shephard and M.K. Georges. Three-dimensional mesh generation by finite octree technique. *Intl. Journal for Numerical Methods in Engineering*, 32:709–749, 1991.

238. J. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications*, 22(1):86–95, 2002.

239. R. Sierra, M. Bajka, and G. Székely. Pathology design for surgical training simulators. In *International Symposium on Surgery Simulation and Soft Tissue Modeling*, pages 375–384, 2003.

240. R. Sierra, G. Szekely, and M. Bajka. Generation of pathologies for surgical training simulators. In *Proc. MICCAI*, volume 2, pages 202–210, 2002.

241. R. Sierra, J. Zatonyi, M. Bajka, G. Szekely, and M. Harders. Hydrometra simulation for vr-based hysteroscopy training. In *MICCAI*, pages 575–582, 2005.

242. R. Sierra, G. Zsemlye, G. Szekely, and M. Bajka. Generation of variable anatomical models for surgical training simulators. *Medical Image Analysis*, 10(2):275–285, 2006.

243. B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

244. E. Simoncelli and J. Portilla. Texture characterization via joint statistics of wavelet coefficient magnitudes. In *In Fifth International Conference on Image Processing*, volume 1, pages 62–66, 1998.

245. K. Sims. Particle animation and rendering using data parallel computation. In *Proc. SIGGRAPH*, volume 24, pages 405–413, 1990.

246. R. Sinkus, S. Weiss, E. Wigger, J. Lorenzen, M. Dargatz, and C. Kuhl. Non-linear elastic tissue properties of the breast measured by mr-elastography - initial in-vitro and in-vivo results. In *ISMRM 10th Annual Meeting*, page 33, 2002.

247. J. Smolle. Cellular automaton simulation of tumour growth: equivocal relationships between simulation parameters and morphologic pattern features. *Analytical Cellular Pathology*, 17(2):71–82, 1998.

248. Cyril Soler, Marie-Paule Cani, and Alexis Angelidis. Hierarchical pattern mapping. In *SIGGRAPH'02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 673–680, 2002.

249. L. Soler, H. Delingette, G. Malandain, N. Ayache, C. Koehl, J.M. Clement, O. Dourthe, and J. Marescaux. An automatic virtual patient reconstruction from ct-scans for hepatic surgical planning. In *Proc. MMVR*, pages 316–322, 2000.

250. U. Spaelter, T. Moix, D. Ilic, M. Bajka, and H. Bleuler. A 4-dof haptic device for hysteroscopy simulation. In *Intl. Conf. on Intelligent Robots and Systems*, pages 3257–3263, 2004.

251. V. Spitzer, M.J. Ackerman, A.L. Scherzinger, and D. Whitlock. The visible human male: A technical report. *Journal of the American Medical Informatics Association*, 3(2):118–130, 1996.

252. M.A. Srinivasan, G.L. Beauregard, and D.L. Brock. The impact of visual information on the haptic perception of stiffness in virtual environments. In *ASME Symposium on Haptic Interfaces*, volume 58, pages 555–559, 1996.

253. L.H. Staib and J.S. Duncan. Boundary finding with parametrically deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:1061–1075, 1992.

254. D. Steinemann, M. Harders, G. Szekely, and M. Gross. Hybrid cutting of deformable solids. In *Proc. of IEEE Computer Society Conference on Virtual Reality*, pages 35–42, 2006.

255. M.A. Styner, K.T. Rajamani, L.P. Nolte, G. Zsemlye, G. Szekely, C.J. Taylor, and R.H. Davies. Evaluation of 3D correspondence methods for model building. In *Information Processing in Medical Imaging*, volume 18, pages 63–75, 2003.

256. K.R. Swanson, C. Bridge, J.D. Murray, and E.C. Alvord. Virtual and real brain tumors: Using mathematical modeling to quantify glioma growth and invasion. *Journal of the Neurological Sciences*, 216(1):1–10, 2003.

257. D. Szczerba and G. Szekely. Macroscopic modelling of vascular systems. In *MICCAI*, pages 284–292, 2002.

258. D. Szczerba and G. Szekely. Simulating vascular systems in arbitrary anatomies. In *MICCAI*, pages 641–648, 2005.

259. G. Szekely, M. Bajka, C. Brechbuehler, J. Dual, R. Enzler, U. Haller J. Hug, R. Hutter, N. Ironmonger, M. Kauer, V. Meier, P. Niederer, A. Rhomberg, P. Schmid, G. schweitzer, M. Thaler, V. Vuskovic, and G. Troester. Virtual reality-based simulation of endoscopic surgery. *Presence - Teleoperators and virtual environments*, 9(3):310–333, 2000.

260. R. Szeliski and D. Tonnesen. Surface modeling with oriented particle systems. *Proc. SIGGRAPH*, 26:185–194, 1992.

261. R. Tait, G. Schaefer, U. Kuehnapfel, and H.K. Cakmak. Interactive spline modelling of human organs for surgical simulators. In *European Simulation Multiconference*, pages 355–359, 2003.

262. G. Taubin. Curve and surface smoothing without shrinkage. In *ICCV*, pages 852–857, 1995.

263. F. Tendick, M. Downes, T. Goktekin, M.C. Cavusoglu, D. Feygin, X. Wu, R. Eyal, M. Hegarty, and L.W. Way. A virtual environment testbed for training laparoscopic surgical skills. *Presence - Teleoperators and virtual environments*, 9(3):236–255, 2000.

264. D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. In *Computer Graphics (Proc. SIGGRAPH'87)*, volume 21, pages 205–214, 1987.

265. M. Teschner, B. Heidelberger, M. Mueller, and M. Gross. A versatile and robust model for geometrically complex deformable solids. In *Proc. Computer Graphics International*, pages 312–319, 2004.

266. D'Arcy W. Thompson. *On Growth and Form*. Cambridge University Press, 1952.

267. J. Tohka. Surface extraction from volumetric images using deformable meshes: A comparative study. In *Proc. of ECCV*, pages 350–364, 2002.

268. N. Tsingos, E. Bittar, and M.-P. Gascuel. Implicit surfaces for semi-automatic medical organs reconstruction. In *Computer Graphics International'95*, pages 3–15, 1995.

269. S. Tuchschmid, M. Grassi, D. Bachofen, P. Frueh, M. Thaler, G. Szekely, and M. Harders. A flexible framework for highly-modular surgical simulation systems. In *ISBMS*, pages 84–92, 2006.

270. Greg Turk. Generating textures on arbitrary surfaces using reaction-diffusion. In *SIGGRAPH'91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 289–298, New York, NY, USA, 1991. ACM Press.

271. Greg Turk. Texture synthesis on surfaces. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 347–354, 2001.

272. W.T. Tutte. Convex representations of graphs. In *Proceedings of the London Mathematical Society*, volume 10, pages 304–320, 1960.

273. R.L. De Valois, D.G. Albrecht, and L. Thorell. Spatial-frequency selectivity of cells in macaque visual cortex. *Vision Research*, 22:545–559, 1982.

274. D. Valtorta and E. Mazza. Dynamic measurements of soft tissue viscoelastic properties with a torsional resonator device. *Medical Image Analysis*, 9(5):481–490, 2005.

275. A. Van Gelder. Approximate simulation of elastic membranes by triangulated spring meshes. *Journal of Graphics Tools*, 3(2):21–42, 1998.

276. L. Van Gool, F. Defoort, J. Hug, G.A. Kalberer, R. Koch, D. Martens, M. Pollefeys, M. Proesmans, M. Vergauen, and A. Zalesny. *Confluence of Computer Vision and Computer Graphics*, chapter Image-based 3d Modeling: Modeling from Reality. Kluwer, 2000.

277. B.C. Vemuri and A. Radisavljevic. Multiresolution stochastic hybrid shape models with fractal priors. *ACM Transactions on Graphics*, 13(2):177–207, 1994.

278. D.R. Veronda and R.A. Westmann. Mechanical characterization of skin-finite deformation. *Journal of Biomechanics*, 3:111–124, 1970.

279. J. Vince. *Introduction to Virtual Reality*. Springer, 2004.

280. C. Vincent, N. Stanhope, and M. Crowley-Murphy. Reasons for not reporting adverse indidents: and empirical study. *Journal of Evaluation in Clinical Practice*, 5(1), 1999.

281. D. Vining. Virtual endoscopy: Is it reality. In *Radiology*, pages 30–31, 1996.

282. John von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966.

283. J. Vozenilek, J.S. Huff, M. Reznek, and J.A. Gordon. See one, do one, teach one: Advanced technology in medical education. *Academic Emergency Medicine*, 11(11):1149–1154, 2004.

284. X. Wang and V. Devarajan. 1D and 2D structured mass-spring models with preload. *The Visual Computer*, 21:429–448, 2005.

285. X. Wang, Y. Shen, and V. Devarajan. Physically accurate mesh simulation in a laparoscopic hernia surgery simulator. In *Medicine Meets Virtual Reality*, pages 568–573, 2006.

286. Y. Wang, C. Chui, H. Lim, Y. Cai, and K. Mak. Real-time interactive simulator for percutaneous coronary revascularization procedures. *Computer Aided Surgery*, 3:211–227, 1998.

287. R. Wasserman and R. Acharya. A patient-specific in vivo tumor model. *Mathematical Biosciences*, 136(2):111–140, 1996.

288. D.F. Watson. Computing the delaunay tesselation with application to voronoi polytopes. *The Computer Journal*, 24(2):167–172, 1981.

289. N. P. Weatherill and O. Hassan. Efficient three-dimensional delaunay triangulation with automatic point creation and imposed boundary constraints. *Intl. Journal for Numerical Methods in Engineering*, 37:2005–2039, 1994.

290. Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH 2000, 27th International Conference on Computer Graphics and Interactive Techniques*, pages 479–488, 2000.

291. Li-Yi Wei and Marc Levoy. Texture synthesis over arbitrary manifold surfaces. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 355–360, 2001.

292. D. Weishaupt, V.D. Koechli, and B. Marincek. *How does MRI work? An introduction to the Physics and Function of Magnetic Resonance Imaging*. Springer, 2003.

293. T. F. Wiegand. Interactive rendering of CSG models. *Computer Graphics Forum*, 15(4):249–261, 1996.

294. R.M. Wilson, W.B. Runciman, R.W. Gibberd, B.T. Harrison, L. Newby, and J.D. Hamilton. The quality in australian health care study. *The Medical Journal of Australia*, 163:458–471, 1995.

295. Andrew Witkin and Michael Kass. Reaction-diffusion textures. In *SIG-GRAPH'91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 299–308, New York, NY, USA, 1991. ACM Press.

296. A.P. Witkin and P.S. Heckbert. Using particles to sample and control implicit surfaces. In *SIGGRAPH Proceedings*, pages 269–277, 1994.

297. Steven Worley. A cellular texture basis function. In *SIGGRAPH'96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 291–294, New York, NY, USA, 1996. ACM Press.

298. Xiaolin Wu. Color quantization by dynamic programming and principal analysis. *ACM Transactions on Graphics*, 11(4):348–372, 1992.

299. X.-Q. Xu, S.-C. Zhu, B. Guo, and H.-Y. Shum. Asymptotically admissible texture synthesis. In *2nd Intl. Workshop of Statistical and Computational Theories of Vision*, 2001.

300. Y.-Q. Xu, B. Guo, and H. Shum. Chaos mosaic: Fast and memory efficient texture synthesis. Technical Report MSR-TR-2000-32, Microsoft Research, 2000.

301. H. Yamada. *Strength of Biological Materials*. Williams & Wilkins Company, 1970.

302. M.A. Yerry and M.S. Shephard. Three-dimensional mesh generation by modified octree technique. *Intl. Journal for Numerical Methods in Engineering*, 20:1965–1990, 1984.

303. Lexing Ying, Aaron Hertzmann, Henning Biermann, and Denis Zorin. Texture and shape synthesis on surfaces. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 301–312, 2001.

304. J. Zatonyi, R. Paget, G. Szekely, and M. Bajka. Real-time synthesis of bleeding for virtual hysteroscopy. In *MICCAI*, pages 67–74, 2003.

305. R. Zayer, C. Rossl, and H.-P. Seidel. Variations on angle based flattening. In *Proceedings of Multiresolution in Geometric Modelling*, pages 285–296, 2003.

306. Jingdan Zhang, Kun Zhou, Luiz Velho, Baining Guo, and Heung-Yeung Shum. Synthesis of progressively-variant textures on arbitrary surfaces. *ACM Transactions on Graphics*, 22(3):295–302, 2003.

307. S.-X. Zhang, P.A. Heng, Z.J. Liu, L.W. Tan, M.G. Qiu, Q.Y. Li, R.X. Liao, K. Li, G.Y. Cui, Y.L. Guo, X.P. Yang, G.J. Liu, J.L. Shan, J.J. Liu, W.G. Zhang, X.H. Chen, J.H. Chen, J. Wang, W. chen, M. Lu, J. You, X.L. Pang, H. Xiao, and Y.M. Xie. Creation of the chinese visible human data set. *The Anatomical Record*, 275B(1):190–195, 2003.

308. S.C. Zhu, X.W. Liu, and Y.N. Wu. Exploring texture ensembles by efficient markov chain monte carlo - towards 'trichromacay' theory of texture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):554–569, 2000.

309. S.C. Zhu, Y.N. Wu, and D. Mumford. Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling. *Intl. Journal of Computer Vision*, 27(2):107–126, 1998.

310. R. Ziegler, W. Mueller, G. Fischer, and M. Goebel. Virtual reality medical training system. In *Computer vision,virtual reality and robotics in medicine (CVRMed)*, pages 282–286, 1995.

311. O.C. Zienkewickz and R.L. Taylor. *The Finite Element Method*. McGraw Hill Book Co., 1987.

Index