

Linköping Studies in Science and Technology

Dissertations, No. 1101

Direct Volume Haptics for Visualization

Karljohan Lundin Palmerius



Linköping University
INSTITUTE OF TECHNOLOGY

Department of Science and Technology
Linköpings universitet

Norrköping 2007

Direct Volume Haptics for Visualization

Karljohan Lundin Palmerius

Cover Image:

Data at Your Fingertips

by Karljohan Lundin Palmerius

Copyright © 2007 Karljohan Lundin Palmerius

Printed by LiU-Tryck, Linköping 2007

ISBN 978-91-85831-98-2

ISSN 0345-7524

Abstract

Visualization is the process of making something perceptible to the mind or imagination. The techniques for producing visual imagery of volumetric data have advanced immensely during the last decades to a point where each produced image can include an overwhelming amount of information. An increasingly viable solution to the limitations of the human sense of visual perception is to make use of not only vision, but also additional senses.

This thesis presents recent work on the development of principles and algorithms for generating representations of volumetric data through the sense of touch for the purpose of visualization. The primary idea introduced in this work is the concept of yielding constraints, that can be used to provide a continuous set of shapes as a representation of features of interest in various types of volumetric data. Some of the earlier identified standard human exploratory procedures can then be used which enables natural, intuitive and effective interaction with the data. The yielding constraints concept is introduced, and an algorithm based on haptic primitives is described, which forms a powerful yet versatile implementation of the yielding constraints. These methods are also extended to handle time-varying, moving and low quality data. A framework for multimodal visualization has been built on the presented methods, and this is used to demonstrate the applicability and versatility of the work through several example applications taken from different areas.

Acknowledgements

I wish to direct my deepest gratitude to my supervisors Anders Ynnerman, Matthew Cooper and Björn Gudmundsson for research assistance, great and sometimes lively discussions, suggestions and proof reading. Thanks also for helping me reach out with my work to both the research community and also the public. A special thanks is given to Anders who gave me this opportunity and has been with me from the beginning. Thanks also to my colleagues at NVIS and at VITA for listening and discussing concepts and details, for suggestions and proof reading.

The Center for Medical Image Science and Visualization is also gratefully acknowledged and thanked for applications for my work and for high quality medical data sets. Special thanks are directed to Anders Persson, Lars Wikström and Andreas Sigfridsson for sets of very special data. Thanks also Mattias Sillén at Saab AB for the high quality fluid dynamics data and for assistance in the implementation of haptic interaction with aerodynamics. The staff at SenseGraphics AB are also gratefully acknowledged for their guidance in building a software package from my research and for help and support in reaching out and spreading my work in the research community.

A big thank-you to my best friend and wife, Anna, for being a solid support in both ups and downs. And to my family and friends, a great thanks for all support and interest.



This work has been partly financed by the National Graduate School in Scientific Computing and by the Foundation for Strategic Research (SSF) under the Strategic Research Center MOVIII.

Contents

I	Context of the Work	1
1	Introduction	3
1.1	Haptics	4
1.1.1	Twofold Perception	4
1.1.2	Computer Interaction	5
1.2	Haptic Disciplines	7
1.2.1	Psychophysics	7
1.2.2	Control Theory	8
1.2.3	Virtual Reality	9
1.3	Volume Visualization	10
1.3.1	Volumetric Data	11
1.3.2	Visual Volume Rendering	12
1.3.3	Haptic Volume Visualization	13
1.4	Research Challenges	14
1.5	Contributions	15
2	Haptic Volume Visualization	17
2.1	Surface Haptics	17
2.2	Haptic Modes for Volume Haptics	19
2.3	Scalar Data Representations	20
2.4	Vector Data Representations	22
2.5	Direct Volume Haptics	24
II	Contributions and Results	25
3	Continuous Shape Representations	27
3.1	Yielding Constraints	27
3.1.1	Proxy-based Implementation	28
3.1.2	Penetrability	29
3.1.3	Viscosity and Friction	31
3.2	Primitives-based Direct Volume Haptics	31
3.2.1	Non-orthogonal Constraints	31

3.2.2	The Primitives-based Approach	34
3.3	Haptic Modes	39
3.3.1	Scalar Data Representations	39
3.3.2	Vector Data Representations	41
3.4	Material Properties	43
3.5	Characteristics	44
3.5.1	Stability	44
3.5.2	Data Reconstruction	45
3.5.3	Psychophysical Aspects	45
4	Low Quality Data	47
4.1	Classification Enhancements	47
4.2	Material Properties	48
4.3	Surface Information	49
5	Dynamics in Time-varying Data	51
5.1	Proxy-point Updating	52
5.2	Dynamic Transforms	53
5.3	Time-varying Data	54
5.3.1	Proxy-point Movements	54
5.3.2	Estimating the Motion Field	55
5.3.3	Conditions	55
6	Volume Haptics Toolkit	57
6.1	Implementation and Technology	57
6.1.1	Haptic Nodes and Rendering	58
6.1.2	Visual Components	58
6.1.3	Data Processing and Miscellaneous Nodes	59
6.2	Demonstration Applications	59
6.2.1	Dichloroethane	59
6.2.2	Sharc UAV	60
6.2.3	Doppler MRI Heart	62
6.2.4	Classification Enhanced MRI	63
6.2.5	Time-varying CT Heart	63
6.2.6	Volume Data Exploration Tool	67
7	Conclusions	69
7.1	Summary of Contributions	69
7.2	Conclusions	70
7.3	Future Work	71
	Bibliography	73

III Appended Papers	77
A Proxy-based Haptic Feedback from Volumetric Density Data	79
B Haptic Visualization of Computational Fluid Dynamics Data Using...	87
C General Proxy-based Haptics for Volume Visualization	101
D The Orthogonal Constraints Problem with the Constraint Approach to...	107
E Enabling Haptic Interaction with Volumetric MRI Data Through...	115
F Enabling Design and Interactive Selection of Haptic Modes	123
G Fast and High Precision Volume Haptics	137
H Haptic Rendering of Dynamic Volumetric Data'	145

THE ARTICLES ARE REMOVED DUE TO COPYRIGHT RESTRICTIONS

Part I

Context of the Work

Chapter 1

Introduction

Computers have become invaluable tools in visualization of data for the purpose of exploring and understanding anything from weather phenomena and car body behaviour in an impact, to medical diagnosis. The techniques for producing visual imagery of volumetric data have advanced immensely during the last decades to a point where each produced image can include an overwhelming amount of information. Still the techniques used for acquiring data are becoming increasingly advanced, producing ever more data that needs to be analyzed more and more accurately and in less time. An increasingly viable solution to the limitations of the human sense of visual perception is to make use of not only human vision, but also other senses. The human perception apparatus is a complex instrument built from at least seven different senses. The five most commonly cited of these are sight, hearing, smell, taste and touch. These senses are closely entangled into a collaborative system used to produce a complete awareness of our current status, situation and surroundings. It is common in computer interaction to make use of only one of these, sight, however at least two more have the potential to increase the awareness of features in data, and thus increase the understanding of its contents. These are the senses of hearing and of touch.

This thesis presents the recent work in a project on the development of algorithms generating representations of data for the sense of touch. This chapter describes the background of the project and the various concepts on which it is based, starting with the concept of haptics in the following section. Section 1.2 describes important areas of research within haptics and section 1.3 provides an introduction to volume visualization. The chapter is finished off with a summary of research challenges and of the contributions of the publications included in this thesis.

The thesis is structured as follows. The next chapter describes and reviews the established ideas and related research in the area. In chapters 3, 4 and 5 the concepts, methods and algorithms developed within this project are described. Chapter 6 then describes a framework for multimodal volume visualization based on these algorithms, and presents demonstration applications built using this framework. A summary of the research and conclusions, are presented in chapter 7. The rest of the thesis contains the included publications.

1.1 Haptics

The two most important concepts in this thesis are volume visualization, which is described in section 1.3, and haptics. While some may refer to haptics as the technology used to generate an artificial sense of touch in the interaction with a computer, a more generally applicable description is that haptics refers to touch regardless of its context or appearance. Haptics is found both in research on the science of touch in computer generated environments and in interaction with the real world, without computers. Over the last decade, haptics has become a huge research area, but the concept dates back far further. For example, the word haptic can be found in old descriptions of touchable art and of some kinds of plants that react to touch. The word is derived from the Greek word $\alpha\pi\tau\acute{o}$ (hapto), which means “tangible”. Simply put, the word means touch, but is generally today only used to describe the concept in a scientific context.

While often discussed in a manner simplified by the context, haptics is a concept of many aspects. As a sense, it is multi-faceted with different collaborating neural systems and psychological components, and as a research area it has many aspects that should be considered. The remainder of this section provides an overview of the most important such systems and aspects.

1.1.1 Twofold Perception

The complete perception of touch is accredited to two different senses: receptors densely populating the skin, and receptors in muscles and joints. While always collaborating to generate the full sensation when touching an object, these two types of receptors are used to perceive different properties of the object at hand.

The skin is populated with three types of *cutaneous* receptors, giving sense of pain, temperature and touch, respectively. Pain is not unimportant, however as a means of conveying information in data it would probably not be very popular. The temperature receptors are capable of determining some surface properties of a palpated object, in particular the thermal conductivity, temperature and permeability of the material. The last class of receptors detects primarily pressure changes at the skin. This includes, however, also shear, vibration and unevenness of a palpated surface. Thus, these receptors are capable of determining textural, roughness and small shape information about what we touch. In common, the sensation produced through these receptors is called *tactile*.

Both muscles and joints have receptors to determine their respective states. This is called *proprioception*, which means perception that responds to an internal stimuli, a reaction to tensions and positions of the limbs. Through intuitive forward kinematics, the body knows the position of, for example, the hand and fingers from the information received through the proprioceptive stimuli. The sense is used for body control, but also to determine weights, forces, friction, viscosity and shapes. This is more commonly known as the *kinesthetic* component of the sense of touch.

These two parts of the haptic sense collaborate to produce the full impression of an examined object. Different properties are identified by each of these and at the end it is often impossible to determine how the properties were identified — the properties of the object are subconsciously mapped from the stimuli in the perception apparatus into a mental image.



Figure 1.1: Two commercially available kinesthetic devices. The left device is a Desktop PHAN-ToM from SensAble Inc. and the one to the right is an Omega.6 from Force Dimension. The red outline shows the part held by the user, which is a pen in these two devices. The green dot shows the position of the probe for each device.

1.1.2 Computer Interaction

In computer science, haptics is used in a type of computer interface where touch is made a part of the information flow between the user and the computer. The human sense of touch is based on the process of palpating objects. It is the dynamic change of stimuli over time as the finger, for example, is moved over a surface that is interpreted as a structure, texture or shape. Thus, the *haptic display unit* (HDU) is suitable for use as an interaction device, such as a mouse, that in the human-computer interface becomes a user input device with haptic feedback. The haptic feedback in this type of interaction is anything from a force response from touching a virtual object, to a vibration giving warning or confirmatory cues.

Since the sense of touch is actually two senses, it is natural to consider two types of HDU used in research: tactile displays and kinesthetic displays. There are, however, also more specialized devices such as *encounter type* devices, knobs and sticks that are no different from any other control except that they show computer controlled behaviour, and *walking units* that simulate the interaction with the ground, such as treadmills and dynamic ground surfaces. While there is active research performed on these types of interfaces, they are still uncommon and have little relation to the work presented here.

Tactile displays include both devices that are capable of generating some sense of tactile touch and those that simply produce vibrations, called vibrotactile units. Examples of tactile units that simulate touch are devices that use pneumatic actuators or servos to manipulate a surface touching the skin, and devices that apply an electrical current to stimulate the cutaneous receptors. Tactile devices are still uncommon, mostly because of the difficulties of building small and effective actuators. Pneumatic and servo-based devices are still large and while electrocutaneous devices can be small, they can be uncomfortable to use. The most promising tactile

display unit to day is the piezoelectric vibrotactile element that can be built into touch screens, thereby enabling confirmatory cues, for example when virtual buttons are pressed. Also, research making use of distributed miniature vibrators over a part of the human body, such as the back or a foot, is not unusual. Is is quite possible to associate vibrations with direction and, through learning, even more abstract notions.

A tactile mouse, an ordinary computer mouse but with an active tactile surface on one or several of the buttons, would be able to produce the sense of touching the appearance of a computer desktop, like a texture. Moving the mouse over the edge of a window or a button, for example, could give the instantaneous feedback feeling like moving the finger over a ridge, a ridge that can be followed by the sense of touch.

Today, the most common haptic devices for computer interaction are the kinæsthetic devices. These work by communicating forces and positions between the user and the computer, and the structural design is often not unlike the form of an industrial robot. The user then interacts with the robot through an end effector, which can be a pen or a ball held by the user. See figure 1.1 for examples. Most common is that the device reads the position, specified by the user through the end effector, and that the output is the force actuated on that same end effector through a set of fast motors in the robot arm. This is called an *impedance control paradigm*, signifying that the user may directly affect the haptic instrument, but that this produces an impeding response in the form of a feedback force. This kind of feedback is commonly referred to as *force feedback*. The alternative type of kinæsthetic device measures a force applied by the user to the end effector. It has absolute control over the position of the end effector and moves it to respond to the applied force. This is called *admittance control paradigm* [dLLFR02], signifying that the haptic instrument admits only certain actions by the user, for example moving the instrument in free space or over a surface, but not through a surface. Devices following this paradigm are generally large to be able to enforce the absolute positioning, and are also strong and produce a superior feedback stability.

Apart from the mechanical characteristics and other implementation specific aspects, kinæsthetic devices are characterized by their *Degrees-of-Freedom* (DoF). This property describes the number of and which independent dimensions that the device is capable of monitoring or controlling. Thus, a device can have different DoF for input and output. In general the 1–3 DoF of a device refers to positional dimensions. In this terminology a device of 2 DoF, for example, is capable of handling positions constrained to a plane. Having 4–6 DoF then refers to full positional motion including one or more rotational axes. Thus, a 6 DoF device is capable of handling full positional and rotational motions.

Most kinæsthetic devices have a single end effector, such as a pen or a ball that the user holds in their hand and through which they interact with the computer. This enables an easy abstraction of the interface between the haptic software and the device drivers: the driver provides a single point as an interface to the haptic device. This point is called the *probe*, which is a point at the base of the end effector of the device, see figure 1.1. For a 6 DoF input device the probe has both a position and an orientation and for a 6 DoF output device, both force and torque can be exerted on the probe. The most common type of commercial high-end haptic device today is a single point impedance control kinæsthetic device of 6 DoF position/orientation input and 3 DoF force output. Examples are the PHANToM series from SensAble inc. [MS94] and the Omega.X

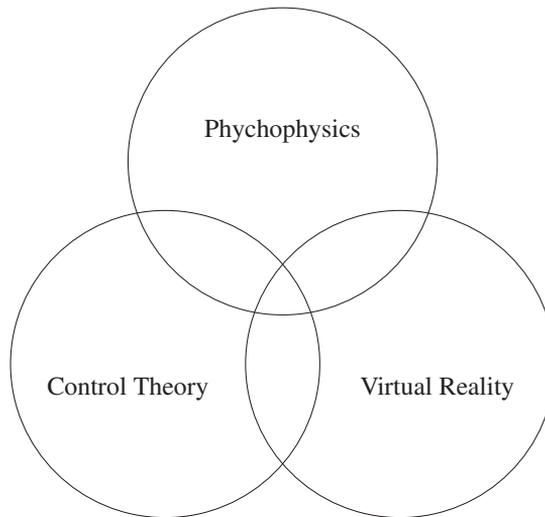


Figure 1.2: The research on haptics is divided into three disciplines with natural overlaps, combinations and collaborations.

family from Force Dimension [For]. The resolution varies but falls generally in the vicinity of $10\ \mu\text{m}$ for position and 0.1° for rotation, while the maximum force that can be exerted is about 10 N. This is the type of haptic device that is considered in this thesis.

1.2 Haptic Disciplines

The research on haptics has become naturally divided into three different disciplines, each with a separate approach to the concept. These are the areas of *psychophysics*, *control theory* and *virtual reality*, see figure 1.2. There are natural overlaps, combinations and collaborations between these disciplines, but they concentrate on different parts of the concept and identify different challenges. The results from these disciplines are often found in separate journals and conferences.

The basic understanding of these different aspects of haptics is important for the clear understanding of the various challenges imposed in the area. The work presented in this thesis is concerned with these challenges, so for the full understanding of the theory, design choices and implementation details presented in this thesis it will be necessary to review these disciplines.

1.2.1 Psychophysics

The oldest branch concerning theories about haptics stems from the field of psychology. This particular branch of psychology is known as *psychophysics*, which is

“...a branch of psychology concerned with the effect of physical processes (as intensity of stimulation) on the mental processes and especially sensations of an organism.” *Merriam-Webster’s Medical Dictionary*¹

In this area it is the implications of touch on human interaction with the world that are the relevant issues. The human part of the haptic process is of most interest and this process is analyzed from the perspective of the human body and mind. Since the psychological concerns are not limited to that which can be simulated using computer devices, research in psychophysics is more concentrated on the tactile sense than the other disciplines. That does not mean that the kinæsthetic part is disregarded.

Topics of interest in this discipline are, for example, the nature of the haptic memory, such as how we remember the tactile impression and shape of objects [KAC88, HB06], and discrimination thresholds [RB02, HTB⁺06, TBS⁺06a, BSH⁺06]. A topic of particular interest for the work presented in this thesis is the process of touch — what forces, features or processes are important for the haptic sense. Research presented by Lederman and Klatzky in [LK87], for example, identifies a set of “exploratory procedures” that are used to identify different properties of the palpated matter:

- *lateral motion*, move a finger across a surface to perceive texture
- *pressure*, press a finger against surface to perceive hardness
- *static contact*, to detect temperature
- *unsupported holding*, to judge weight
- *enclosure*, closing the hand around an object to perceive global shape
- *contour following*, follow the edge or shape of an object to get an image of its shape

Our sense of touch provides a synergistic image produced from both tactile and kinæsthetic components. In interaction with computers and other technical systems, it is important that the equipment allow for both components. Thus, when using a haptic interaction device, a combination of tactile and kinæsthetic stimuli is usually exerted through the instrument. Low-frequency feedback is typically perceived through the joints and muscles, while the high-frequency feedback is registered as vibrations or texture. For the haptic instrument to be able to provide both types of stimuli, it must be quick and able to convey the fastest changes. Thus, most systems update the haptic feedback at very high frequency, which puts a limit on the time for the estimation of the haptic feedback.

1.2.2 Control Theory

In the control theory discipline, haptic interaction is treated as a control system and appropriate theories and methods are applied to analyze the haptic behaviour and improve performance. This system incorporates the following parts:

¹Retrieved November 9, 2006

- the user of the system
- the haptic device
- the computer and its control system

These parts and their mutual communications are shown in figure 1.3. The system of haptic interaction is a dynamic system due to its ability to change the internal states, such as hand position and velocity, over time. The hand/device part of the system is time continuous while the device/computer part is time discrete, which makes it a hybrid system. The connection between these two parts is generally handled through a simple sample-and-hold in the continuous to discrete time conversion, and by applying a piecewise constant output in the opposite direction.

With this view of the haptic interaction, the algorithm that generates the haptic feedback can be viewed as a control system that aims to affect the motion of the users hand through the sampled position data. Since the haptic interaction forms a closed loop, any sudden force or other action will be fed back and affect the system for a time. Thus, a latency in the system will introduce a delay in the feedback that may reinforce oscillation at certain frequencies. This introduces a second reason for updating the haptic feedback at a high frequency To make the control loop stable for a wide range of control algorithms, most systems run at a frequency of at least 1 kHz which gives a latency of less than a millisecond. The higher the sampling frequency, the lower the integration error becomes that is caused by the piecewise constant output from the computer. To simulate hard surfaces, for example, a stiff control is required, which increases the integration error and, in turn, gives rise to instability if the sampling rate is not high enough.

Research of particular significance for the work presented here is that aiming at improving the stability of complex control algorithms. The algorithms for haptics presented in this thesis make use of an approach for increasing the haptic stability called virtual decoupling [CSB94, AH99] (sometimes referred to as virtual coupling). The principle of this approach is discussed in section 2.1.

1.2.3 Virtual Reality

Haptics made an early entry into the realm of virtual reality. In virtual reality, computer graphics is combined with advanced display systems and control devices to provide an immersive and natural interaction with the environment. In this way the fast and effective natural interaction with the real world can be applied in various applications, and real situations can be examined and tested in a fully controlled environment. This discipline is concerned mainly with the algorithms, issues and challenges in integrating the technology of haptics the into graphical environments, for example algorithms for generating the haptic feedback, typically force feedback, and visual feedback from touching virtual objects.

This discipline is concerned with the implementation of applications and simulators such as those for virtual prototyping, building digital mock-up models for testing functionality, feel and usability, and surgery simulators for medical education and pre-operative planning and training. Since haptics is a natural part of our interaction with the real world, it can be introduced into virtual reality and graphics for increased *realism*, *presence* and *immersion*. For example, the

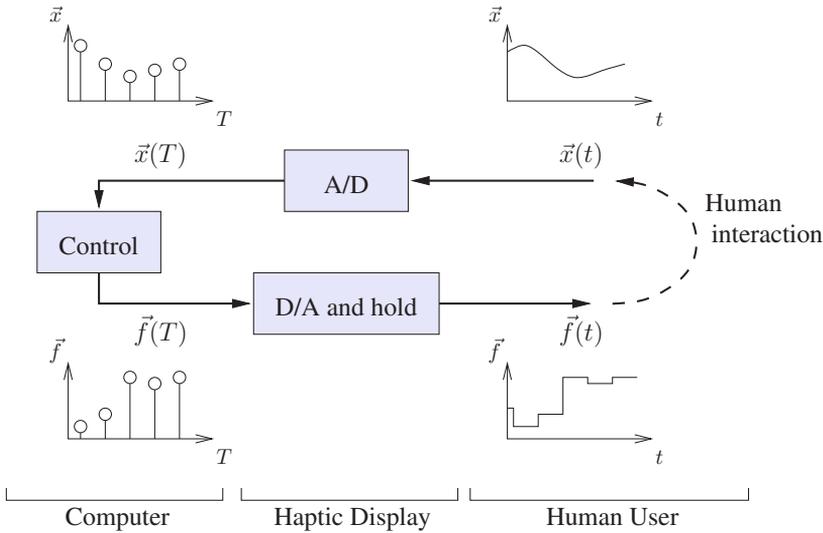


Figure 1.3: Haptic interaction can be considered to be a dynamic, hybrid system with time continuous real world dynamics and a discrete time software control system, connected through the haptic display. The haptic display converts the continuous real world position, \vec{x} , to a digital value through sampling and renders the force feedback, \vec{f} , continuous by holding the time-discrete value specified by the haptic algorithm constant between the samples.

haptic feedback physically prevents the user from moving their hand or body through an object. This type of feedback also allows for *increased control* by activating the proprioceptive control of the limbs, thus potentially speeding up the completion of complex tasks. Evaluations of well defined tasks [WH00, KD02, PNCD01, IN93, AKH01, WPS⁺02] have shown that haptics has the potential to significantly increase both speed and accuracy of human-computer interaction. Haptic feedback also provides additional *information* about the material of panels and buttons in the environment thereby increasing the sense of presence, but potentially also providing a more accurate interpretation and understanding of objects.

A system supporting interaction with more than one sense is called a *multimodal* system. In this thesis visualization of data acquired through different methodologies and with different representations and contents is handled. Systems simultaneously handling multiple types of data are generally also called *multimodal*, hence there is a risk of confusion.

1.3 Volume Visualization

Visualization is the process of making something perceptible to the mind or imagination. In computer science this refers to the rendering of data onto a display to convey a message or make understandable some concrete or abstract features contained in the data. This display is

often a monitor showing visual images, but may also be a haptic display providing a haptic representation of the data, loudspeakers conveying auditory cues or, in the future, maybe even an olfactory display. Volume visualization is the art and science of presenting the information that resides in *volumetric data*, or volumes.

The amount of information contained in a volume can be immense and it is often advantageous to process the data to make important features it more easily perceivable. The process of visualizing volumetric data includes tasks like *data reduction* to lower to amount of displayed data, *data extraction* to automatically select parts of the data that may be of interest, *data enhancement* to improve the definition of faint but potentially important features, *data representation* to convert data into a form that can be shown on the designated display. The main aim is to convert the data from an abstract cloud into a representation that can be perceived and understood, and thereby make best use of the human analytical system.

1.3.1 Volumetric Data

Interaction with most objects in the real world is through their surfaces. For example, we see the skin of our body and the green surface of the leaves on the trees. A volume describes not only the surfaces of geometrical object, such as a car or the skeleton of a human patient, but also the full distribution of information outside and inside of these surfaces. Examples are the air pressure or air flow around the body of a car, or the value of some tissue parameter throughout the body of a human patient. The two examples in figure 1.4 illustrate the difference between volume visualization and an example of non-volume visualization.

A volume is essentially a function in 3D space. A scalar volume is a mapping from a position in space to a scalar parameter ($V : \mathbb{R}^3 \rightarrow \mathbb{R}$) and a vector volume maps the position to a vector property ($\vec{V} : \mathbb{R}^3 \rightarrow \mathbb{R}^N$ where N is typically 3). Volumetric data can, at any observed scale in the real world, be considered continuous down to an unmeasurable granularity. In computers the storage space is limited, so volumes are in computer science generally defined not continuously throughout the occupied space, but only at discrete sample points called voxels. For the purpose of representing data at any position in the volume, interpolation is used.

Examples of volumetric data used in the work presented in this thesis are Computer Tomography (CT) data, Magnetic Resonance Imaging (MRI) data and Computational Fluid Dynamics (CFD) data, which are sampled volumes. A continuous volumetric data set must be defined by one or a set of analytical functions that can be numerically estimated at any position. As an example, an analytically defined function is used in section 6.2.1 to simulate the electropotential of a molecule.

In the context of the work presented here it is unimportant whether a volume is discrete or continuous and all volumes in the remainder of this thesis are considered to be continuous. Volumes that are sampled by nature can simply be interpolated into a continuous function using interpolation. In this work tri-linear interpolation has been used throughout.



Figure 1.4: Two examples of visualization. The left image^a is an example of visualization of car collision deformation and the right image^b is an example of volume visualization. Observe how translucency in the volume visualization reveals internal features not present in visualization of surface data.

^aImage courtesy of Wikipedia (<http://www.wikipedia.org>).

^bData set courtesy of VolVis distribution of SUNY Stony Brook (<http://www.volvis.org>).

1.3.2 Visual Volume Rendering

The full data of a volume cannot be solidly rendered. That would produce the visual impression of a solid cube where only the data at the six sides are visible. Much of the process of creating a visual representation of volumetric data is to remove unwanted and unnecessary parts in the data and enhance the important features through a more sparse visual representation. In this way unimportant, redundant and occluding information in the volume is removed to better show the important parts.

A typical example of removing data is the extraction and rendering of isosurfaces in scalar volumes. Many algorithms exist that extract an explicit isosurface geometry from scalar data, a geometry that can then be rendered on the screen. The rendering of geometrical representations of volumetric data is a type of indirect volume rendering. As such it suffers from the disadvantage that it only represents a pre-selected subset of the data, in this case at positions defined by a simple isovalue. A more powerful approach to produce a visual representation of volumetric data is *direct volume rendering* (DVR). In this approach the full data is considered, but only parts that are important are rendered by manually or semi-automatically making uninteresting parts transparent or semi-transparent. In this way the interesting parts stand out and can be visually identified or explored by a user of the system. The DVR approach can, by considering the full volume, potentially show properties that cannot be represented by simple isosurfaces such as width of a surface. This can represent, for example, the thickness of the skin in medical visualization.

It is common in volume rendering that a set of *transfer functions* is used, each mapping one scalar value to another ($\tau : \mathbb{R} \rightarrow \mathbb{R}$), to map the scalar value of the volume to the red, green, blue and opacity (alpha) components of the visual rendering. The transfer functions are thus used to specify the visual impression of different values in the volume, and which values should be transparent and how transparent they should be. There are also several techniques available for feature enhancements, such as modulating the opacity by the magnitude of the gradient in the data to enhance borders regions with different scalar values, or by modulating it by the curvature

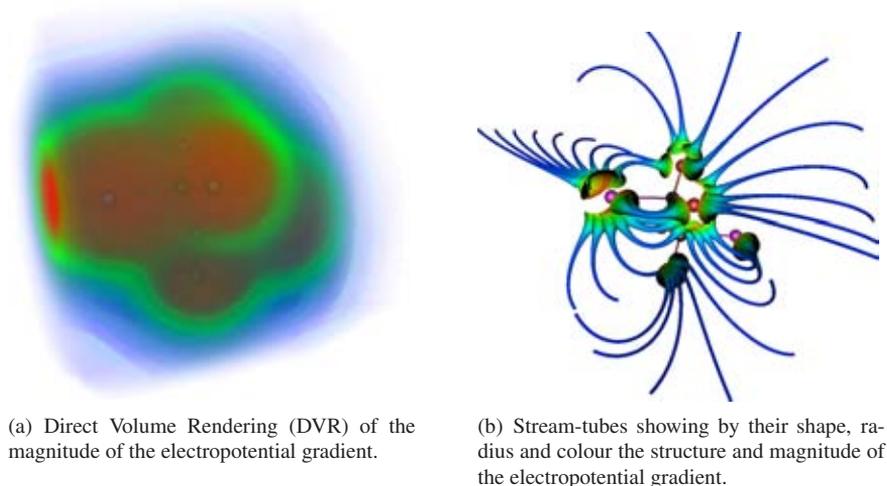


Figure 1.5: Two examples of vector volume visualization. This vector volume represents the electropotential gradient around a dichloroethane molecule.

with respect to the viewer's position to enhance silhouettes.

For vector data, each point in the volume consists of several independent components that makes up a vector. The number of properties that may be of interest increase accordingly. The vector magnitude and divergence, for example, are scalar properties that can be rendered as described above. Vector properties are typically visually rendered using some geometry that, at the position of the geometry, provides a visual representation of the data. An example of this is the use of glyphs, small icons such as arrows, distributed in the volume. These represent the magnitude and direction of the vector field through size, colour and orientation. Another common example is stream-lines, stream-ribbons and stream-tubes — lines, ribbons and tubes that show the path that a particle would flow from a predefined “seed” position if the vector field represented a fluid flow. Stream-ribbons can, by their rotation through the volume, represent rotation of the vector field, while stream-tubes are capable of also representing an additional scalar property through the tube radius. Stream-tube visualization and DVR of the same example data are shown in figure 1.5.

1.3.3 Haptic Volume Visualization

Haptic volume visualization is the art and science of producing haptic feedback that produces guidance and conveys information about structures and features in volumetric data in the purpose of enhancing the understanding of the data. It is typically used as a complement to visual volume visualization in a multimodal approach in an attempt to make use of even more of the power of human perception. The haptic feedback has shown potential for both guidance and conveying

information from the environment, both of which may be of great use in the exploration of complex data. The haptic feedback typically provides local cues and guidance to improve the understanding of data at the palpated position. This provides a natural local enhancement of the global information provided by the visual display.

The work presented in this thesis has aimed at developing new algorithms that provide better haptic feedback from volumetric data for volume visualization and thus more powerful methods for visualization. It is also desirable to find a deeper understanding of the issues and challenges involved in the integration of haptics and graphics in such applications, and the implications of the actual user behaviour, preferences and understanding.

1.4 Research Challenges

While most research in haptics approaches the problem from one of the views or disciplines described earlier, every result provides an overlap between the areas. Similarly, the challenges in haptic volume visualization, while approaching the problem from the virtual reality point of view, have contributions from all three. Challenges in the implementation of algorithms for haptic exploration of volumetric data include

- to ensure high stability so that unwanted vibrations and other artifacts are avoided, and so that the feedback does not harm the user or become active or unnatural
- to provide natural and intuitive feedback, so that learning threshold is kept low
- to provide advanced and versatile feedback, so that the modes of interaction can be adjusted to the situation at hand and provide effective guidance and information for any type of data
- to allow for free exploration by removing the occlusion caused by distinct surfaces in the data
- to provide a haptic representation of the data at any position in the volume

The search for high stability holds for any haptic algorithm because instability introduces unwanted artifacts that disturb the natural interaction. In interplay with real objects, there are no such vibrations. Artifacts and instability often spring from imperfections in the control systems of the software or from a delay calculating the haptic feedback that exceeds the time limit of 1 ms suggested above. The nature of the haptic control system, being a dynamic hybrid system, poses challenges for the control algorithms to dampen numerical errors while still maintaining a high fidelity connection between the virtual objects and the force feedback.

Providing natural and intuitive feedback, and feedback that provides the best bandwidth and guidance for a situation are not necessarily combinable. In certain situations the more natural or intuitive mode of interaction may be much less effective at representing the data, or guiding the user. There may, therefore, be a trade-off between these goals and the challenge is to find methods that provide adjustments and calibrations that supports the search for the optimal balance.

1.5 Contributions

This section contains a short review of the main contributions of each publication included in this thesis. The author of this thesis is first author and main contributor of all the included papers and articles, and is the single author of this thesis.

Paper A The introduction of proxy-based volume haptics and the use of yielding constraints as a shape representation of volumetric data

Paper B Additional haptic modes from yielding constraints, for interaction with vector data

Paper C The introduction of haptic primitives as a means for combining yielding constraints and force functions, and allowing for non-orthogonal constraints in volume exploration

Paper D A proof and description of the orthogonality problem with the constraint-based approach to volume haptics

Paper E Applying knowledge-based tissue separation to enable high fidelity haptic rendering, with tissue specific material properties, of data with low signal to noise ratio and overlapping tissue scalar ranges

Paper F The complete framework for haptic visualization based on the haptic primitives, and a qualitative evaluation identifying important issues and aspects in the utilization of haptic visualization

Paper G The design details on a fast and high precision analytical solver for the haptic primitives and a general numerical solver, and a solver chain design that first selects the high fidelity solver but enables fall-back on the general solver if the analytical solver fails

Paper H The principles and algorithms required for haptic rendering of moving and time-varying volumetric data

Chapter 2

Haptic Volume Visualization

Adding haptic feedback to volume visualization has the potential to increase interaction precision and speed, and improve the understanding of the data. The haptic impression complements the visual with cues that can be recognized and understood both consciously and subconsciously, forming a secondary but synergistic and important part of the experience of the data. The main challenge in haptic volume visualization is to design an algorithm that converts the user actions into a force reaction that represents the local volumetric data in a useful and stable manner. This chapter discusses the alternatives for converting volumetric data into a haptic feedback and reviews algorithms and related research on the topic.

The following section reviews algorithms for haptic interaction with explicit surface data. These are common methods for generating haptic feedback in virtual reality applications, but the techniques used also reappear in algorithms for haptics in volume visualization. The rest of the chapter then reviews methods and algorithms for generating haptic feedback from volumetric data, with particular focus on methods suitable for visualization.

2.1 Surface Haptics

Algorithms for haptic interaction with explicit surface information are designed to generate a force feedback when the haptic probe comes in contact with the surface. Since a force feedback device (impedance control) is incapable of explicitly controlling the position of the haptic probe, the surface simulating control system must allow the probe to penetrate surfaces. When a surface is penetrated, however, a force is applied to stop the probe from penetrating further. This is the basic principle of surface rendering which is common to all algorithms for impedance-based haptic feedback.

The penetration of surfaces is not as serious an issue as it might seem. Since the kinæsthetic sense of touch has a low resolution at low frequencies (e.g. shown in [LT69]), the displacement is not as prominently perceived through touch as it is through vision. Thus, the impression of penetration can be reduced by giving the visual impression of the haptic instrument not penetrating the surface. This way the increasing resistance when applying increasing force and penetrating

the surface further is rather perceived as an increasing force applied to the surface.

The first developed approach used for force feedback from geometrical surfaces apply a force that pulls the haptic probe towards the closest point on the closest polygon of the object boundary, see figure 2.1(a). The force strength is made proportional to the penetration depth, as if a spring was connected between the probe and the surface. This gives a feedback force that is the effect of a “penalty” from the penetration of the polygon surface, which gives it the name *penalty-method* [MS94, SBM⁺95].

The proportionality constant defines the stiffness of the force feedback. This stiffness can, to some degree, be used to simulate the hardness of surfaces. It is, however, primarily a parameter of the control system in which the point on the surface is the reference point and the probe is the controlled signal [CSB94]. The control system is in that respect essentially a classical PID regulator, in this case with zero integration and derivative parameters. A non-zero damping term is sometimes used to improve the feedback fidelity. Since the haptic algorithm is a discrete control system which suffers from noise in the sensor read-off, the instant derivative estimation is not reliable. Furthermore, the controlled model includes the user’s hand and other things that are unknown to the haptic algorithm and may even change during the simulation, so the optimal damping is hard to determine. It is, therefore, not uncommon that the damping term is omitted in the setting of the haptic parameters.

The integration component of the PID regulator removes steady state differences between the input and output signals. Introducing that term would remove the surface penetration after holding the haptic instrument still on a geometrical surface for a while. For the kinæsthetic part of the haptic perception the accurate and high fidelity dynamics is of much higher importance than the removal of a small constant error [LT69, ITR06].

The penalty method suffers from artifacts that make it impractical in real applications. Since the approach represents a static control it has, at the time the feedback is calculated, no memory of which surface was previously palpated. Because of this, the algorithm can suddenly treat another surface, at this instant closer to the probe, as that currently palpated. This gives rise to such artifacts as pop-through of thin objects when the opposite side of the object suddenly becomes closer to the probe than the first palpated side, and discontinuities around edges and corners. To remove these artifacts the system needs a memory of what part of the palpated object was touched the last time the haptic feedback was estimated. This memory is implemented through a virtual object that is left on the surface that the probe penetrates. Each time the feedback is calculated for a haptic frame, the system now knows the previous surface position.

The first implementations following this approach used a single surface point called a *god-object* [ZS95, Hut00], see figure 2.1(b). Using a single point as memory for interaction, however, has some disadvantages. Numerical errors in the estimation of triangle surfaces sometimes leave small gaps between the triangles composing the surface of an object, and such gaps have proved large enough for the god-object. Thus, the god-object could fall through object surfaces. The god-object method was refined by Ruspini et al. in [RKK97] to avoid the need for the explicit topology information by the introduction of a finite-sized spherical *proxy* object, see figure 2.1(c).

The *proxy* is an internal representation of the haptic probe. It is fully controlled by the surface simulation algorithm and can be constrained by surfaces in a stable manner. The force feedback

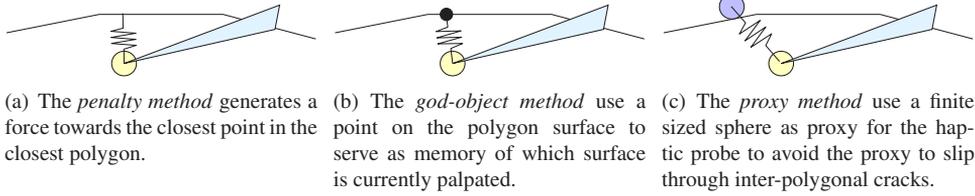


Figure 2.1: The three most common approaches for generating haptic feedback from polygonal surface data.

is then calculated by simulating coupling through a virtual spring damper,

$$\vec{f}_{fb} = -k(\vec{x}_{probe} - \vec{x}_{proxy}) - D(\vec{v}_{probe} - \vec{v}_{proxy}) \quad (2.1)$$

where k is the stiffness of the coupling and D is a dampening term. In free space the proxy is automatically moved to the position of the probe and no feedback is generated through the virtual coupling. When an object is penetrated by the probe, the proxy is moved over the surface towards the probe. By modulating the movement of the proxy over the surface, other effects can be generated, such as friction, texture, haptic shading and even bump-maps. Because both the god-object method and the proxy-based method let surfaces constrain the movements of the proxy object, they are sometimes also referred to as *constraint-based*.

2.2 Haptic Modes for Volume Haptics

When haptic interaction is made available in computer graphics applications, it is generally based on the notion of surfaces. The haptic feedback is generated as a response to touching geometrical representations of surfaces in the virtual environment, and the feedback is perceived as a surface. Different algorithms for haptic interaction with surfaces all strive towards a common goal — more stable and more correct or realistic surface feedback. Haptic interaction with volumetric data is different. In volumetric data there are no explicit surfaces. Some volumetric data sets contain data that could be interpreted as surfaces, but not all. Since the volumetric data can not be directly interpreted in a straightforward haptic form, some haptic representation of the contents must be generated in a selected or designed manner. The term *haptic mode* was suggested by Pao and Lawrence [PL98] as a word describing the unique haptic impression in interaction with volumetric data. A haptic mode is a distinct haptic representation of data that provides a unique connection between data and representation. Thus, different haptic modes applied to the same data may

1. provide haptic representations of different properties of the volumetric data
2. provide different haptic effects representing the same property of the data

Volumetric data sets may contain different types of attribute data, both with respect to the dimensionality (scalar, vector or tensor), and with respect to what the values in the data set

represent. A vector volume may, for example, represent air or fluid flow information generated through CFD, or the strength and orientation of a magnetic field. While a certain haptic mode may be compatible with scalar data, it might not work with vector data. Similarly, a mode compatible with vector data but designed for intuitive interaction with fluid flow data might not be appropriate for exploration of a magnetic field, even if the data is compatible. The mode could be counter intuitive or simply not convey the most important properties of the data.

Here follows a review of previous methods suggested for interaction with volumetric scalar and vector data.

2.3 Scalar Data Representations

In haptic interaction with scalar data, such as CT data, an obvious mode of interaction is with isosurfaces in the data. There are many methods available for isosurface extraction, some readily available in various libraries for visualization. One example is the Marching Cubes algorithm [LC87], a popular approach because of its memory efficiency, speed and relatively straightforward implementation. The haptic feedback is then generated through interaction with this geometrical representation of the data by applying one of the readily available algorithms for surface haptics.

The explicit extraction of global isosurfaces in the volume can be time consuming and prohibits the interactive updating of surface value and position. An alternative is then to use a local intermediate surface representation [KSW⁺99, CHS00]. The local methods require less memory than global and, since they require no pre-processing, they also provide quicker response to changes in the data. These methods do, however, require run-time surface estimation.

If the haptic feedback is generated by a surface that can be implicitly defined, such as an isosurface, the feedback can be generated without the use of an intermediate representation. Implicit surfaces is a straightforward and viable alternative to the use of explicit geometrical surfaces in scalar data. These can be implicit isosurfaces in volumetric data, as suggested in [ST97, KKSD02], but also NURBS (Non-Uniform Rational B-Spline) as described in [TJC97]. The algorithms used for haptic interaction with implicit surfaces are similar to those for interaction with explicit surface representations: when the surface has been penetrated, a force is generated that pulls the probe towards the surface with a force proportional to the distance to the surface. These algorithms even apply a proxy point, similarly to the methods for the haptic rendering of polygonal data.

Some of the characteristics of geometrical representations of haptics may detract from the positive impact of adding haptics to the exploration. The primary common characteristics are the existence of discrete, distinct and predefined or interactively defined constraints. This means that, in a situation where surface interaction is natural, the feedback is generally crisp and stable. The use of distinct, impenetrable constraints, however, suffers from the potential occlusion of important regions in the volume. A user may need to deactivate the haptic feedback to allow the probe to be moved into a new region, and then re-activate the feedback. Furthermore, by limiting the interaction to discrete positions in space, only a subset of the data is represented through the haptic feedback. The full consequence of this can be avoided by providing interactive

redefinition of the haptic geometries. For example, the geometries can be assigned a maximum strength which, when this force is exceeded, causes the current geometry to be redefined at a new position that renders a lower force, as suggested in [IBHJ03]. This, however, has the effect of producing a “snap” sensation when moving the haptic probe between palpated regions.

An alternative to the haptic rendering of explicit or implicit surfaces in scalar data is the *direct force mapping*, or *force function* approach. Here the force feedback is estimated through a vector-valued function of the volumetric data, extracted at the probe position, \vec{x}_{probe} . Sometimes the probe velocity, \vec{v}_{probe} , is also used to produce a viscosity feedback representing the data. Designing a haptic mode is then a matter of designing a function, $\vec{\mathcal{F}}$, that maps the data into an understandable and usable feedback force, \vec{f}_{fb} ,

$$\vec{f}_{\text{fb}} = \vec{\mathcal{F}}(\vec{x}_{\text{probe}}, \vec{v}_{\text{probe}}) \quad (2.2)$$

where $\vec{\mathcal{F}}$ is dependent on the data.

There are many ways the volumetric data can be translated into a vector-valued feedback force. In interaction with scalar data, it is necessary to extract a vector-valued property of the haptic interaction. For example, the velocity of the probe is a suitable vector-valued property to generate a viscosity feedback [AS96, HQK98]. By letting the viscosity scale relative the local scalar value, a haptic sense of the local value is conveyed,

$$\vec{f}_{\text{fb}} = -\tau(V(\vec{x}_{\text{probe}}))\vec{v}_{\text{probe}} \quad (2.3)$$

where V is the scalar volume and τ is a transfer function describing a mapping between the scalar value and the viscosity. The feedback from this function is dependent on the speed of exploration. If the user wants to perform closer examination, the lower examination speed will reduce the magnitude of the feedback force. This is a problem identified by, among others, Aviles and Ranta in [AR99].

A way to represent the relative distribution of the scalars in the volume is to generate the force from the scalar gradient,

$$\vec{f}_{\text{fb}} = \tau(V(\vec{x}_{\text{probe}}))\vec{\nabla}V(\vec{x}_{\text{probe}}) \quad (2.4)$$

as suggested by several researchers in [IN93, AS96, GSM⁺97, HI97, HQK98]. This force function pushes the haptic probe either towards or away from regions with high scalar value, depending on the sign of the transfer function, τ .

The gradient-based force-function feedback can work well with volumes with low frequency contents, producing a soft push towards regions of interest or even generating a feedback similar to surfaces. If the force scaling is set high, however, or the scalar data contains high frequency regions, unstable behaviour can occur in the form of vibration. This is caused by the energy added by the gradient force to the control system as the probe moves through the volume. In regions where the gradient vector changes magnitude or direction quickly, the probe will start moving back and forth causing high or low frequency vibrations. This problem can be reduced by adding a damper but that reduces the feedback fidelity.

A combination of the characteristics of the viscosity and the gradient directed force has also been suggested in [PL98, MGS96]. By projecting the probe velocity onto the gradient vector, a viscosity feedback is produced only in the direction of the gradient,

$$\vec{f}_{\text{fb}} = -\tau(V(\vec{x}_{\text{probe}})) \frac{\vec{v}_{\text{probe}} \cdot \vec{\nabla}V(\vec{x}_{\text{probe}})}{|\vec{\nabla}V(\vec{x}_{\text{probe}})|} \quad (2.5)$$

This function provides a viscosity feedback with a sense of the orientation of the scalar distribution in the volume. Since the viscosity feedback absorbs energy from the haptic interaction, this mode of interaction provides better stability than the gradient force, which adds energy to the system. It suffers, however, from the same disadvantage as ordinary viscosity, as described above, since the feedback depends on the speed of the users movements.

Alternative feedback can be produced by extracting more advanced, even global, properties of the volume prior to the haptic interaction. Bartz et al., for example, propose in [BG00] the use of the gradient force described above, but on pre-processed data. By extracting a scalar data set that describes the distance for every voxel to the closest surface in a pre-segmented data set, the gradient force pushes the haptic probe away from surfaces into the centre of a cavity. By extracting a scalar data set that, for every voxel, describes the length of the shortest path to a target position, the gradient force pushes the probe towards that position. Both these two fields are used to guide the probe through a cavity towards a target location in the data.

A simple method for increasing the feedback quality and generating more advanced haptic feedback is to add a memory to the function and thereby let the feedback be dependent, not only on the position of the probe in the data, but also its path to that position. One example is a haptic mode for exploration of shockwaves in CFD data presented in [LLPN00]. Here the distance that the probe has moved since the feedback was last estimated is projected onto the gradient vector. The resulting vector is accumulated into an estimation of the penetration depth into a shockwave in the scalar data. This penetration depth is then used to estimate the force feedback. When the gradient magnitude is large enough to be deemed part of a shockwave, the penetration depth, D , and force feedback are estimated according to

$$D^n = D^{n-1} + \frac{(\vec{x}_{\text{probe}}^n - \vec{x}_{\text{probe}}^{n-1}) \cdot \vec{\nabla}V(\vec{x}_{\text{probe}}^n)}{|\vec{\nabla}V(\vec{x}_{\text{probe}}^n)|} \quad (2.6)$$

$$\vec{f}_{\text{fb}} = -k D^n \vec{\nabla}V(\vec{x}_{\text{probe}}^n) \quad (2.7)$$

where \vec{x}_{probe}^n and D^n are the probe position and penetration depth at the n th estimation of the haptic feedback, V is the scalar volume and k is the stiffness constant.

2.4 Vector Data Representations

A straightforward haptic representation of vector data is implemented by constraining the haptic probe to follow the vector field along the direction of the vectors. This approach can be

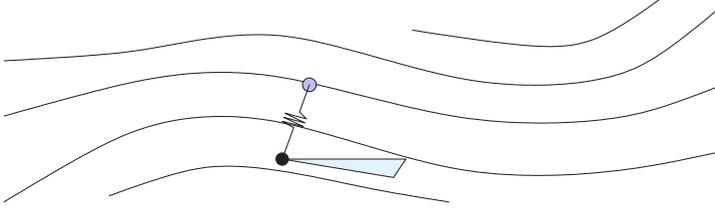


Figure 2.2: The geometrical representation of a vector volume is generated by constraining the proxy object to follow the vector field.

considered the haptic equivalent to the visual stream-lines used in volume visualization. This mode of haptic interaction with vector fields has been suggested and used for example by Pao and Lawrence under the name *virtual constraint* in [PL98], and Donald and Henle under the name *follow mode* in [DH00]. Like the haptic rendering of implicit surfaces in scalar data, the geometrical haptic representation of vector data is implemented by adding a proxy object. This object is then moved towards the probe but always moved, using Euler or Runge-Kutta integration, to follow the vector field, see figure 2.2. The feedback from the probe displacement relative the proxy then pulls the probe towards the stream-line which produces the virtual constraint, or follow mode.

Locking the haptic feedback to implicit stream-lines in the data is equivalent to limiting the haptic rendering to pre defined isovalues in the volume. For vector data, the direct force mapping is also then a possible alternative approach.

When interacting with vector data, in contrast to scalar data, there is already a vector valued property that can be used to generate haptic feedback: the local vector. The simplest form of haptic feedback from volumetric data is to produce a direct mapping from the local vector to the force feedback,

$$\vec{f}_{fb} = C\vec{V}(\vec{x}_{probe}) \quad (2.8)$$

where \vec{V} is the vector volume and C is a force scaling constant. To provide more control over the magnitude of the feedback, the constant C can be replaced with a transfer function, τ , of the magnitude of the local vector and the vector itself is normalized. In this way a non-linear mapping between vector magnitude and feedback strength can be obtained.

Pao et al. suggests, in [PL98], the use of the difference between the probe velocity and the local vector value to generate viscosity and thereby produce a *relative drag*. This gives a feedback similar to flow, which gives a sense of both the direction and magnitude of the vector field. Another suggestion in the same publication is to generate the viscosity only when moving the probe perpendicular to the local vector, called *transverse damping*. This function also describes the direction and magnitude of the vector field, but does so by generating a guiding sensation, a feeling of viscosity only when not moving the probe in the direction of the field.

A more advanced force function is suggested by Lawrence et al. in [LLPN00]. In this function a vector representation of the vorticity in the data is first extracted,

$$\vec{\varphi} = \left(\vec{\nabla} \times \vec{V}(\vec{x}_{probe}) \right) \times \vec{V}(\vec{x}_{probe}) \quad (2.9)$$

where \vec{V} is the vector data. This vector points towards the centre of vortices in the data and, since the function performs only local operations, points towards the centre of rotation for partial vortices as well. By applying this vector with appropriate scaling through transfer functions as force feedback, a haptic representation of the vorticity is obtained,

$$\vec{f}_{fb} = \tau(|\vec{\varphi}|) \frac{\vec{\varphi}}{|\vec{\varphi}|} \quad (2.10)$$

In interaction with a complete vortex the feedback pulls the probe towards the vortex core and guides the user to follow the extent of the vortex.

2.5 Direct Volume Haptics

For the haptic feedback to effectively convey information about the data throughout the volume, the algorithm needs to both allow for free exploration by not introducing occlusion from distinct surfaces in the data, and provide a haptic representation of the data at any position in the volume. These qualities are not shared by the methods described for indirect haptic rendering of the data, methods that use some intermediate representation of the data. By analogy with Direct Volume Rendering (DVR) for visual rendering of volumetric data, to generate haptic feedback directly from the volumetric data, rather than through an intermediate representation, is *Direct Volume Haptics* (DVH).

The force function methods described above are all examples of DVH. They are capable of representing the data at any position in the volume, and generally do not occlude potentially important regions. It can, however, be considered an over-simplistic approach to represent complex volumetric data with a simple force. Especially since research shows that the human ability to discriminate between force directions is poor [TBS⁺06b, BSH⁺06, TBS⁺06a, HTB⁺06]. Furthermore, many forms of force functions are prone to instability for some data.

The work presented in this thesis introduces methods for Direct Volume Haptics that combine the benefits of both force functions and surface-based haptic representations. They represent data at any position, do not suffer from haptic occlusion while providing shape representations of data and retaining the stability of surface-based haptic rendering. This is done by implicitly rendering shape representations simultaneously at all positions in the data, a continuous set of shape representations.

Part II

Contributions and Results

Chapter 3

Continuous Shape Representations

The first major contribution of the work presented in this thesis is the use of a continuous set of shapes as a representation of features in volumetric data. The basic idea is that at any position in the data a haptic shape is rendered with properties representing the local data. Standard exploratory procedures, identified by Lederman and Klatzky [LK87], can be used if the feedback is in the form of shapes, which enables natural and intuitive interaction with the data. The concept of haptic shapes is introduced through an extension of the constraints concept, used in geometry-based surface interaction, into yielding constraints which can be defined throughout the volume. This chapter gives an overview of the key aspects of this contribution, introduced primarily in papers **A** and **B**, and extended in **C**. The motivation for the development of the yielding constraints is found in the need for natural haptic feedback from volumetric density data with representing solid matter, such as CT scans of human tissues. The haptic feedback should give a feeling of the object that becomes mentally coupled with the visual representation of the contact. The basic principle of this approach, however, can also be used to represent other types of data, for example vector data.

The following section presents and describes the concept of yielding constraints. Section 3.2 then describes the haptic primitives — a powerful and versatile implementation of yielding constraints. In section 3.3 the implementation of haptic modes is described and section 3.4 then discusses material properties in haptic interaction in the context of the yielding constraints. The chapter ends with a section on the important characteristics of this interaction method.

3.1 Yielding Constraints

A yielding constraint is a haptic effect that imitates the feedback from a geometric shape, but yields to a certain applied force. By allowing the local shape to yield, the user can move the haptic probe in any direction, even through features in the data. A continuous distribution of shape representations of the volumetric data is obtained by generating a local haptic shape at the position of the haptic probe, regardless of that position, not limited to following isosurfaces or any other distinct pre-defined locations and thus avoiding occlusion of potentially important

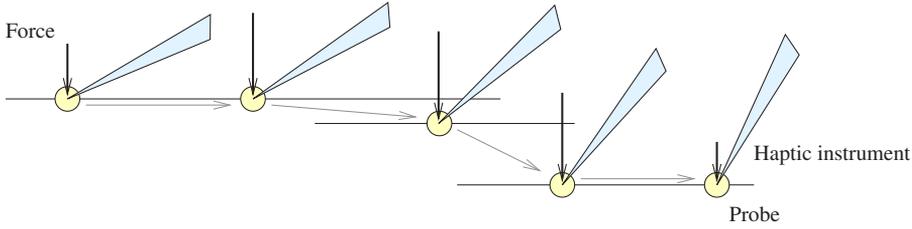


Figure 3.1: The concept of yielding constraints. When a force is applied that exceeds the strength of the constraint, it yields to the force and allows the probe to move through to a new position and a new constraint representing the local data at the new location.

regions. When the user applies a force exceeding the strength of the constraint, the constraint yields, allowing the user to move through it, while the feedback is continuously calculated using new constraints defined by the data at each new position. The principle of yielding constraints is also shown in figure 3.1. The strength of the constraint is a naturally perceived property which can effectively be used to represent important numerical properties of the data.

3.1.1 Proxy-based Implementation

An algorithm for DVH works by generating an implicit local haptic representation of data extracted at the probed position. In paper **A** the yielding constraints are introduced in DVH. The implementation is based on proxy movements which are derived from haptic surface rendering, described in section 2.1, to support the notion of yielding constraints. By handling a proxy point that is constrained by features in the virtual environment and coupling this with the haptic probe through a virtual spring, the features derived from the local data are perceived as distinct shapes. For each haptic time-frame the haptic feedback is estimated in the following steps, also shown in figure 3.2:

1. determine the directions and strengths of the local constraints
2. move the proxy in each (linearly independent) direction
3. estimate the force feedback from the new proxy position using the virtual coupling.

Determine the directions The direction of the constraints are determined from the contents of the data. In paper **A** the primary feature to be represented through the feedback is the notion of a continuous set of virtual surfaces at the probed position in the density data. This surface effect is modelled using a constraint, the orientation of which is estimated through the gradient operator. The second haptic effect used in surface simulation is the friction feedback. Static friction is clearly a yielding constraint, resisting the movement unless a large enough force is exerted. The direction of this constraint is always perpendicular to the first constraint, generating the surface effect, and also directed in the opposite direction from the movement of the haptic probe. This arrangement of constraints is shown in figure 3.3.

Move the proxy When the directions (or orientations) of the constraints have been determined, the proxy should be moved to simulate the effect from these constraints. To simplify the problem, the constraints are limited to be always orthogonal, so that their individual influence on the force feedback becomes linearly independent. In this way the proxy movements can be handled separately in the direction of each constraint. The initial position of the proxy is the location from the previous time step in the haptic loop. For each constraint the proxy is then moved only if the force exerted by the coupling equation in that direction yields a larger force than that specified as the strength of the constraint. The proxy is then moved to the position (in the currently handled dimension) from which the coupling equation in that direction yields the exact constraint strength, see figure 3.4. The proxy is thus moved according to

$$\vec{x}'_{\text{proxy}} = \vec{x}_{\text{proxy}} + \hat{q} \min(0, \hat{q} \cdot (\vec{x}_{\text{probe}} - \vec{x}_{\text{proxy}}) - s/k) \quad (3.1)$$

where \hat{q} is the constraint direction, s is the strength of the constraint and k is the stiffness of the virtual coupling.

The strength of the constraints used in the haptic representation of the data are controlled through transfer functions from numerical properties in the data. This is a type of material property, an effect conveying information from the data, discussed in detail in section 3.4.

Estimate the force feedback Finally, the force feedback for the haptic frame is estimated through the virtual coupling (equation 2.1). Since the constraints are linearly independent, the combined movements yield a position that fully corresponds to the force contributions of each constraint dimension.

These three steps are repeated for each haptic frame, generally at a rate of 1 kHz, producing both a force feedback for that frame and a new proxy position. Thus, the proxy is moved a very small distance in space at each haptic frame. Since the yielding constraints provide local linear approximations of the features in the data, these movements integrate to a full approximation of curved features in the data. A discussion on the accuracy of this integral is provided in section 3.5.2.

3.1.2 Penetrability

When a surface feature representation yields, a new surface feature beneath the previous one is represented through the haptic feedback. The sensation can be described as an anisotropic 3D friction. While this behaviour is natural for friction feedback — as the probe is moved away from the position that gave friction resistance, the new position also provides friction resistance — the way this applies also to surfaces may not be perceived as natural. The study presented in paper **F** shows that it can require some training to understand the nature of the yielding representations of the data and learn the palpation procedure that most effectively extracts information about the data. The study, however, also shows that after only a short training period enough understanding is gained to make effective use of haptic modalities based on yielding surface constraints.

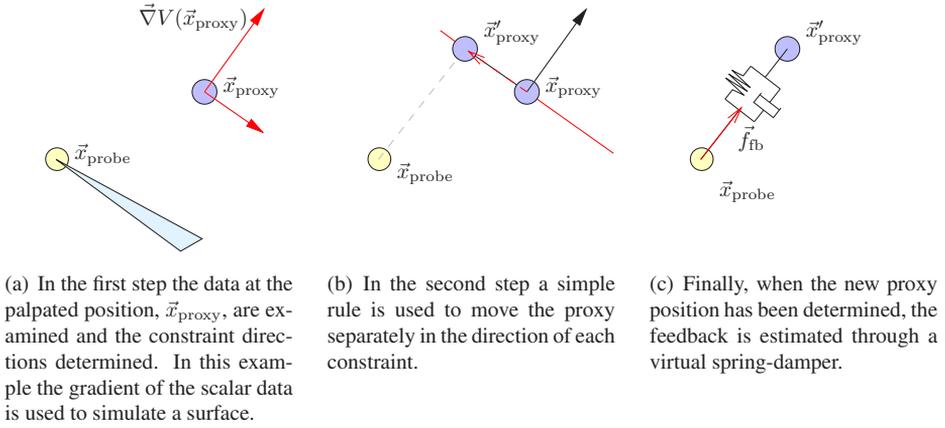


Figure 3.2: The three steps used to update the proxy position and calculate the force feedback.

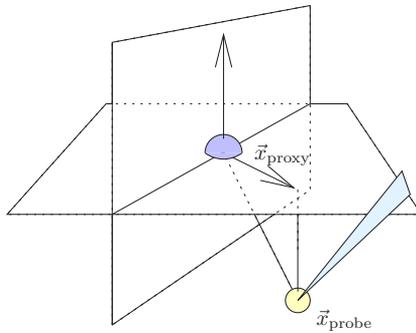


Figure 3.3: The constraint constellation for the surface and friction mode. The surface constraint orientation is defined by the gradient vector and the friction constraint is set orthogonal to it and facing towards the position of the probe.

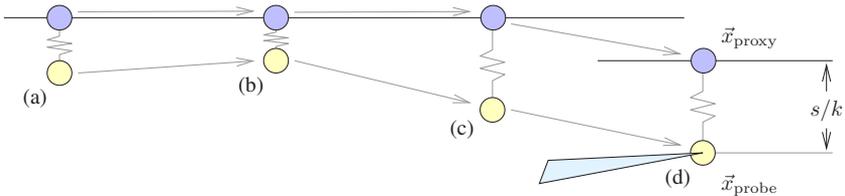


Figure 3.4: The penetration of a surface with the proxy-based approach to yielding constraints. As long as the applied force is less than the strength of the constraint (s), the proxy follows the constraint (a-c). When the force exceeds the strength, the proxy moves through the constraint, ending up at a new constraint below the previous one (d).

3.1.3 Viscosity and Friction

Viscosity can be simulated by using a yielding constraint in the opposite direction from that in which the haptic probe is pulling the proxy ($\vec{x}_{\text{proxy}} - \vec{x}_{\text{probe}}$). This type of viscosity generates a resistive force that is directly defined by the strength of the constraints. This is a simplification of real viscosity, which is proportional to the sum of a constant property and terms from powers of the probe velocity. The strength assigned to the constraint can, if this is desired, be set as a function of the velocity, however removing the higher order terms makes the strength of the feedback directly dependent on the data properties. Removing the dependency on the velocity of the haptic probe, and thus making the feedback independent of the speed of exploration, yields both higher stability and better control of how the information is presented to the user. This increased perceptual stability has also been recognized by Aviles and Ranta in [AR99].

Friction is implemented in a similar way using a yielding constraint. However, in this case the strength is relative to the normal directed force. The proportionality is specified through the friction constant, μ . It is also possible to check if the proxy point is currently moving, and so allow separate friction constants for static and dynamic friction. This static/dynamic friction model is a simplification of the physically correct behaviour of surface friction, but a very popular and widely used one. In fact, the behaviour of a yielding constraint is also similar to the popular *friction cone* model [HM02], an approach used to introduce friction in the god-object based approach to surface haptics, described in section 2.1.

3.2 Primitives-based Direct Volume Haptics

This section describes an implementation of yielding constraints based on haptic primitives. The haptic primitives constitute both a means for estimating the haptic feedback, and an interface for implementing haptic modes. The work presented in this chapter was introduced primarily in paper C but is also discussed and improved in papers F and G.

The programming interface constituted by these primitives has shown to provide an intuitive and powerful means for designing and implementing haptic modes. Through this interface the haptic feedback with shape representations of data is programmed as easily as a force function, by simply defining the equation that extracts the feature of interest from the data. A second motivation for the primitives-based approach is that, in contrast with the implementation described above, it is capable of handling constraints that are not orthogonal.

First, the orthogonality problem is described together with examples of situations where the constraints will not be orthogonal. Section 3.2.2 then discusses the principles of the haptic primitives, how they represent features in volumetric data and how they are implemented using a proxy-based approach, see also figure 3.5.

3.2.1 Non-orthogonal Constraints

The implementation of yielding constraints described above (section 3.1.1) limits itself to orthogonal constraints. With this simplification the proxy movements can be handled individually

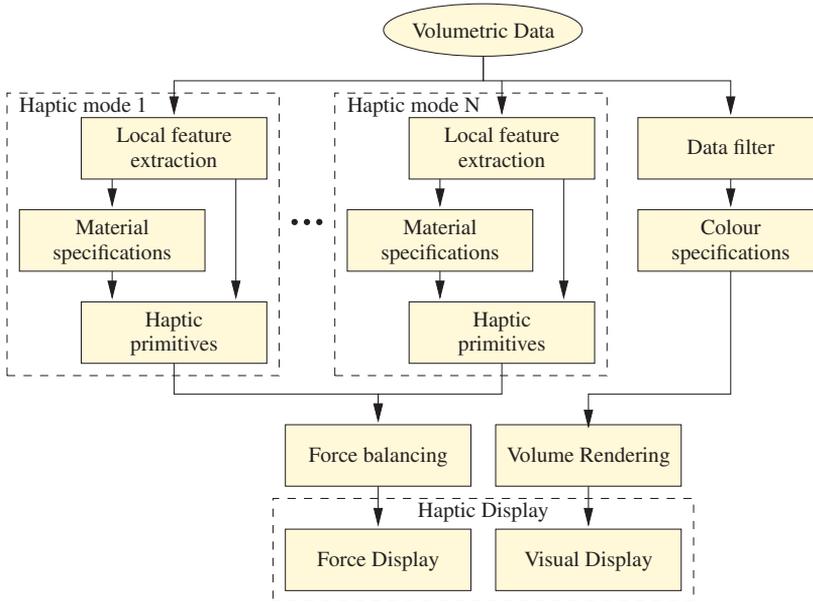


Figure 3.5: The data flow in the primitives-based approach to volume haptics with an analogy to visual volume rendering. The data is first extracted and used to control haptic primitives. This constitutes the haptic mode. These haptic primitives are then combined into a single force feedback that is sent to the force display.

in each constraint direction. If the constraints that are used to define the haptic feedback are not orthogonal, the assumption that the constraints are linearly independent will be wrong and so will be the resulting proxy position. This will result in incorrect haptic feedback and severe artifacts.

Experience has shown that orthogonal constraints are common and that many visualization applications can be handled perfectly well using this approach since, in most cases, a single data set is visualized at a time and a single feature from this data set is of primary interest and is thus represented using haptic feedback. From the very nature of the previous implementation it is clear that it is incapable of handling non-orthogonal constraints, however further analysis of the problem and a proof showing the nature of the problem are provided in paper **D**. That paper also identifies two situations where this will be an issue: in the simultaneous handling of multiple data sets and in haptic interaction with multiple modalities.

Multiple Data Sets

In most visualization applications one data set is loaded into the application and one data set is shown in a single graphical window. If a new data set is loaded the previous set is removed. To

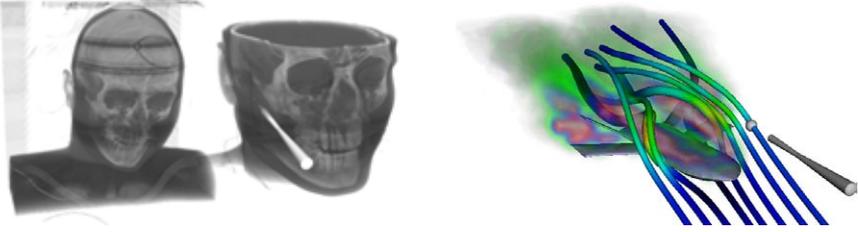


Figure 3.6: Non-orthogonal constraints are common in cases where two visualizations are presented in the same virtual environment (left) and when multiple data properties are used simultaneously to generate haptic feedback (right), in this example flow and curl magnitude.

view two data sets these are generally loaded into separate windows or different view ports of the same window. As immersive and semi-immersive visualization environments become more common and we move closer to virtual reality based multimodal systems this might change.

In a virtual reality environment the user may simultaneously handle multiple visualizations. To compare two data sets the user may hold them side by side. In this situation, the two data sets both provide haptic feedback and since the constraints from the two separate data sets are not correlated the individual features are likely to become non-orthogonal. If this is not handled by the system, the feedback will be incorrect.

Multiple Modalities

Visualization of a single object may also be subject to this orthogonality issue when more than one mode is used simultaneously. In interaction with more complex data the interplay between multiple feature shape properties may be of primary interest. These properties may be features extracted from the same data set. Different features extracted from the same data are often orthogonal, for example the flow vector and vortex vector discussed in section 2.4, but there are many cases when they are not. One example is the flow vector and magnitude gradient that can both be extracted from vector data,

$$\vec{v}_1 = \vec{V}(\vec{x}) \quad (3.2)$$

$$\vec{v}_2 = \vec{\nabla} |\vec{V}(\vec{x})| \quad (3.3)$$

This combination is used in the study presented in paper **F**, performed with an application also described in section 6.2.3.

More common is that different features of interest reside in separate co-registered data sets. Data extracted from MRI and CT, for example, can be co-registered to form a combined multimodal data set with more potential information about the underlying tissues than either by itself. Haptic feedback representing features from two such combined data sets can seldom be guaranteed to be orthogonal. A common example is the overlaid pressure and flow data in CFD data. Exploring the flow behaviour with respect to pressure features can be of great interest in virtual wind tunnel applications, but will inevitably generate non-orthogonal constraints.

3.2.2 The Primitives-based Approach

The implementation of yielding constraints described above only considers one dimensional constraints. To generate the effect of constraint in more than one dimension, such as friction which is two dimensional, the one dimensional constraint is rotated towards the probe position, see figure 3.3. In the haptic primitives approach, each such effect is, instead, represented by a separate type of constraint, a haptic primitive. The haptic primitives are used as an abstraction layer defining the haptic effect, an interface used when implementing the haptic mode. They are also a means of calculating the actual feedback, which is done using a force balancing solver, as shown in figure 3.5. A haptic mode is then implemented by controlling the parameters of the haptic primitives as functions of the volumetric data. This is described at the end of this section and in section 3.3.

Haptic Primitives

In this approach the constraint effect is represented by three separate haptic primitives, one for each number of degrees of freedom in which it can restrict the probe movements:

point primitive is a 3D constraint, providing an omni-directional constraint effect towards a point

line primitive is a 2D constraint, allowing free motion along a line, but constraining perpendicular movements

plane primitive is a 1D, semi-directional constraint, resisting motion against the orientation of the constraint

In addition to these three constraints there is a fourth primitive, designed to provide a means for generating a force function from the data:

force primitive produces a push in the orientation of the primitive, a response similar to that of ordinary static force functions

By including this last primitive as a part of the frame work, the effect from a simple force can be handled together with yielding constraints. It can push the haptic instrument along the constraints, or strengthen or weaken the effect from constraints depending on the direction of the force relative to the effect from the constraints.

The behaviour of each haptic primitive is controlled through a set of parameters. All primitives have strength parameters, s , and the point, line and plane primitives have position parameters, \vec{x} . The line, plane and force primitives have direction parameters, in the form of a unit vector, \hat{q} , specifying the orientation of the constraint or the direction of the force. From these parameters each primitive provides a well-defined haptic effect. A haptic interaction mode is then implemented by selecting one or several haptic primitives that together provide the desired behaviour, and controlling their parameters throughout the simulation. The haptic primitives that are chosen and configured to produce the haptic effect representing the volumetric data are combined into a single force feedback.

Force Balancing

Through the virtual coupling, the proxy position provides a representation of the force feedback, at the same time as being the palpated position in the data. In fact, any feedback or haptic effect can be represented by controlling the proxy position. In primitives-based volume haptics, the proxy is moved to simulate the effect from haptic primitives. For this, each haptic primitive is assigned a force function, $\vec{\mathcal{F}}_i(\vec{x}_{\text{proxy}})$, that acts on the proxy to pull it towards the primitive, in the case of the three constraint primitives, or to push it in the specified direction in the case of the force primitive. The effects of the functions are individual to each primitive and also controlled by the primitive parameters, position, direction and strength. The force functions for the four haptic primitives are listed in table 3.1.

Both the virtual coupling and the force functions of the primitives are functions of the proxy point. Both also represent a force. Thus, to connect the primitive effects to the force feedback, the proxy is positioned so that the virtual coupling and the primitives produce an equal force,

$$k(\vec{x}_{\text{proxy}} - \vec{x}_{\text{probe}}) = \sum_i \vec{\mathcal{F}}_i(\vec{x}_{\text{proxy}}) \quad (3.4)$$

Due to the nature of the included force functions (see table 3.1), this function includes discontinuities. This means that there does not always exist a perfect balance. For example, when a plane primitive is balanced against the force feedback, as shown in figure 3.7, the residual will be non-zero when the solution is found at the primitive position, at the discontinuity. A similar behaviour can be observed from all the constraint primitives. Because of this the proxy position is instead located by finding the minimum of the difference between the primitives' force and the virtual coupling,

$$\operatorname{argmin}_{\vec{x}_{\text{proxy}} \in \mathbb{R}^3} \left| \sum_i \vec{\mathcal{F}}_i(\vec{x}_{\text{proxy}}) - k(\vec{x}_{\text{proxy}} - \vec{x}_{\text{probe}}) \right| \quad (3.5)$$

This is the haptic *balancing equation*, or the *residual equation*. As is demonstrated by the example in figure 3.7, this is not an approximation. The minimization of this function will yield the correct proxy position representing all included primitives.

Force Solvers

The force balancing equation specified above is minimized to provide the proxy position that represents the effect of all involved haptic primitives. This balancing constitutes a non-linear optimization problem. The minimization problem is solved using a tailored separate solver. The input to the solver is the probe position, the stiffness of the coupling equation, the primitives and their configurations, and the output is the proxy position.

Two solvers for the proxy position are presented in paper **G**. The orthogonality problem discussed above is an issue only for advanced virtual reality based visualization applications and special visualization problems. Therefore, the first solver is designed to handle the common case of orthogonal constraints. These prerequisites allow the solver to use an approach similar to the simple proxy movements described in section 3.1.1 (equation 3.1). The proxy is first placed

Point Primitive	$\odot_{\vec{x}}^s(\vec{x}_{\text{proxy}}) = \begin{cases} \vec{0}, & \text{if } \vec{x} - \vec{x}_{\text{proxy}} = 0 \\ s \frac{\vec{x} - \vec{x}_{\text{proxy}}}{ \vec{x} - \vec{x}_{\text{proxy}} }, & \text{if } \vec{x} - \vec{x}_{\text{proxy}} \neq 0 \end{cases}$
Line Primitive	$\odot_{\hat{q}, \vec{x}}^s(\vec{x}_{\text{proxy}}) = \begin{cases} \vec{0}, & \text{if } \vec{m} = 0 \\ s \frac{\vec{m}}{ \vec{m} }, & \text{if } \vec{m} \neq 0 \end{cases}$ $\vec{m} = \hat{q} [\hat{q} \cdot (\vec{x}_{\text{proxy}} - \vec{x})] - (\vec{x}_{\text{proxy}} - \vec{x})$
Plane Primitive	$\oplus_{\hat{q}, \vec{x}}^s(\vec{x}_{\text{proxy}}) = \begin{cases} 0, & \text{if } (\vec{x}_{\text{proxy}} - \vec{x}) \cdot \hat{q} \geq 0 \\ s\hat{q}, & \text{if } (\vec{x}_{\text{proxy}} - \vec{x}) \cdot \hat{q} < 0 \end{cases}$
Force Primitive	$\ominus_{\hat{q}}^s(\vec{x}_{\text{proxy}}) = s\hat{q}$

Table 3.1: The force functions of the haptic primitives with simplified notation. The primitives are controlled through the parameter’s position, \vec{x} , strength, s , and orientation. The last of these is specified through a unit vector, \hat{q} .

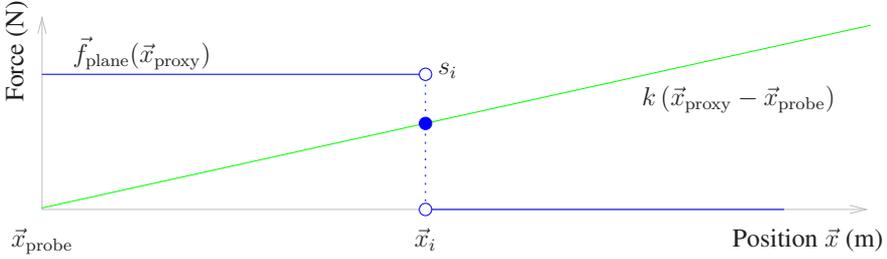


Figure 3.7: The equilibrium between a plane primitive with strength s_i at position \vec{x}_i in balance with the force feedback. In this case the point for which the residual is minimized yields the position of the plane, which means that the proxy will slide on the constraint.

at the probe position, corresponding to a zero force feedback from the coupling equation. It is then iteratively moved in accordance with each haptic primitive in turn. This makes the solver analytical and thus, since the proxy movements are quickly calculated, both fast and precise.

It is shown in paper **G** that this solver is also capable of correctly handling some non-orthogonal constraints. These are those cases where the haptic primitive that generates the one constraint is fully contained in the subspace of the primitive that generates the other non-orthogonal constraint. A common example is when a point primitive is co-located with a line or plane primitive.

In the case of non-orthogonal constraints falling outside of this special case, the system needs to fall back on a solver capable of handling any configuration of primitives. This is done by the second solver. To be able to handle any configuration of primitives this solver is designed as a numerical solver. It iteratively moves the proxy point from an initial position towards a lower residual value, thereby approaching the minimum. In principle this approach constitutes a gradient descent optimizer but, since the function is discontinuous the gradient is not usable.

Instead the residual force is used as the direction of descent. Since the system describes a force balancing, the residual is known to point towards a lower residual. Using a step length of δ , the iteration can be expressed as

$$\vec{x}_{\text{proxy}}^{n+1} = \vec{x}_{\text{proxy}}^n + \delta\varepsilon/|\varepsilon| \quad (3.6)$$

where ε is the residual in the balancing of equation 3.5. Only the discontinuities of the force functions (table 3.1) hinder the straight propagation towards the global minimum.

Yielding Primitives

Each time the haptic feedback is estimated, typically at a rate of 1 kHz, the force balancing places the proxy at a position controlled by the current primitives and their parameters. Since the force function of each primitive is balanced against the force feedback, the feedback will reflect the force function of each primitive used to represent the feedback. For each constraint primitive (point, line and plane) this effect is a pull towards the primitive.

Because of the nature of the primitives' force functions, the proxy ends up at the primitive if the primitive strength is greater than the stiffness multiplied by the distance between the probe and the primitive. This effect is shown in figure 3.8. If the primitive is absolutely positioned in the virtual environment, the probe is simply pulled towards this primitive regardless of how far from the primitive the probe is pulled, as shown in figure 3.8(a). The strength of the feedback will first be proportional to the probe's distance from the primitive, but level out at a magnitude equal to the primitives strength.

When using the haptic primitives to represent some volumetric data the primitives are positioned at the proxy position each time the force feedback is to be estimated. If the proxy ends up in balance between the probe and the primitive, the primitive will be positioned at a different position the next time the feedback is estimated, see figure 3.8(b). This produces a yielding effect, for one primitive identical to that of the earlier constraint-based method. Since any number of haptic primitives can be included in the balancing equation simultaneously, this effect works for any combination of haptic primitives.

Primitives as an Interface

From the description of the haptic primitives they can be considered a link between the formula that extracts the features from the data and the formula that calculates the new proxy position. This new proxy position then, through the virtual coupling, represents the feedback. On one side of the link there are equations extracting feature orientation and numerical properties that are used to configure the selected set of primitives, and on the other there is a solver estimating the proxy position that provides the feedback specified through the primitives. Regarding the configuration of the primitives as the actual specification of the haptic feedback, this is similar to the implementation of direct force mapping. Features from the data, represented by vectors for orientation and by numerical properties, are extracted through formulae and mapped to the feedback definition. In direct force mapping this definition is simply a force. In the primitives-based approach this is the primitives configuration.

The haptic primitives provide intuitive interfaces for specific haptic effects, and earlier restrictions have been removed. This makes it more straightforward to design and implement new haptic modes. Any number of haptic modes can be combined, permitting a large number of possible combinations simultaneously providing haptic representations of multiple properties in one or several data sets. The haptic abstraction interface constituted by the haptic primitives thus becomes a motivation in itself, providing an additional benefit over the original concerns related to non-orthogonal constraints.

3.3 Haptic Modes

While the original motivation for the implementation of yielding constraints was to produce natural interaction with density data the approach can also be effectively applied to other types of data and be designed to produce various haptic feedback from these types. Some of the haptic modes described here are based on force mapping and some on yielding constraints. The modes based on yielding constraints can be implemented using either of the two presented approaches for implementing yielding constraints, but by using the haptic primitives as a basis for all modes it is both easier to design and to implement the modes, and the individual modes can be combined into more advanced feedback schemes. By placing haptic primitives at the proxy position and controlling their parameters as functions of the data, a wide range of haptic modes have been implemented — three modes for scalar data and five for vector data are described here. Some are reimplementations of earlier force function modes, while others have been introduced in the appended publications. The haptic modes for scalar data and for vector data and their respective contribution to the balancing equation (equation 3.5) are listed in tables 3.2 and 3.3, respectively.

The numerical properties of the haptic modes are all controlled through transfer functions. This is discussed in detail in section 3.4.

3.3.1 Scalar Data Representations

Three haptic modes have been implemented using haptic primitives for interaction with scalar data.

Viscosity Mode

The viscosity mode is implemented using the point primitive. This primitive restrains movement in any direction and thus provides a smooth resistance to any motion of the probe. The effect is a friction-like resistance when moving the haptic probe through the volume. The magnitude of the resistance force is directly proportional to the properties of the data without any velocity dependent component, which provides a higher perceptual stability compared to feedback based on real viscosity models.

This haptic mode can be used either as a standalone mode, or in combination with other modes to provide feedback where homogeneous or empty regions produce no feedback from the

Viscosity Mode	$\vec{\mathcal{F}}(\vec{x}'_{\text{proxy}}) = \odot_{\vec{x}=\vec{x}'_{\text{proxy}}}^{s=\tau(V(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}}))}(\vec{x}'_{\text{proxy}})$
Surface and Friction	$\vec{\mathcal{F}}(\vec{x}'_{\text{proxy}}) = \oplus_{\substack{s=\tau_s(V(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}})) \\ \vec{q}=\hat{n}}}(\vec{x}'_{\text{proxy}}) +$ $\ominus_{\substack{s=\tau_\mu(V(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}})) \\ \vec{q}=\hat{n}}}^{N_f}(\vec{x}'_{\text{proxy}})$ $\hat{n} = \frac{\vec{\nabla}V(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}})}{ \vec{\nabla}V(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}}) }$ $N_f = \min(\tau_s(V(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}})), k\hat{n} \cdot (\vec{x}'_{\text{proxy}} - \vec{x}'_{\text{probe}}))$
Gradient Force Mode	$\vec{\mathcal{F}}(\vec{x}'_{\text{proxy}}) = \oplus_{\vec{q}=\hat{n}}^{s=\tau(\vec{\nabla}V(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}}))}(\vec{x}'_{\text{proxy}})$

Table 3.2: The haptic modes for scalar data. Each function, $\vec{\mathcal{F}}(\vec{x}'_{\text{proxy}})$, represents the mode's force contribution in the balancing equation, as a function of the new proxy position, \vec{x}'_{proxy} . These modes provide haptic representations of the local scalar data, V , at the previous proxy position, \vec{x}'_{proxy} .

other modes. With the combination, the feedback from all the modes are perceived simultaneously.

Gradient Force Mode

The gradient force mode is a common haptic mode that can easily be implemented using direct force mapping, as is described in section 2.3. It pushes the haptic probe in the direction of the local gradient in the scalar data. As with any direct force mapping, this mode is implemented by applying the force function to a force primitive.

Like the original force function implementation of this mode, this implementation can suffer from potential vibrations and oscillations in some data. This is because the force primitive is not a yielding constraint, and so adds energy to the system just like direct force mapping.

Surface and Friction Mode

The surface and friction mode generates a haptic feedback that mimics surface feedback from scalar data. This mode was originally designed to be applied to density data, such as CT data, and provides a natural representation of this type of data. The surface features are extracted using the gradient operator on the data at the palpated position and the resulting gradient vector is used as the surface normal to represent the local surface features.

The implementation of the surface and friction mode is done using a combination of two haptic primitives. The surface effect is modelled using a plane primitive. The primitive is oriented using the gradient vector and the strength is controlled by the scalar value and modulated by the gradient magnitude. Friction forces are always perpendicular to the surface that exerts the friction. This effect is provided by a line primitive with the same orientation as the plane primitive. The normal directed force, used to estimate the friction strength, is estimated as the force exerted by the coupling equation in the direction of the surface normal. To estimate the friction strength, this force is multiplied with the data-specific friction constant, μ , see table 3.2.

Force Mode	$\vec{\mathcal{F}}(\vec{x}'_{\text{proxy}}) = \ominus_{\vec{q} = \frac{\vec{V}(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}})}{ \vec{V}(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}}) }}^{s=\tau(\vec{V}(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}}))}(\vec{x}'_{\text{proxy}})$
Follow Mode	$\vec{\mathcal{F}}(\vec{x}'_{\text{proxy}}) = \bigcirc_{\vec{x}=\vec{x}'_{\text{proxy}}, \vec{q} = \frac{\vec{\nabla}V(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}})}{ \vec{\nabla}V(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}}) }}^{s=\tau(\vec{V}(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}}))}(\vec{x}'_{\text{proxy}})$
Front Shape Mode	$\vec{\mathcal{F}}(\vec{x}'_{\text{proxy}}) = \bigoplus_{\vec{q} = \frac{\vec{V}(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}})}{ \vec{V}(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}}) }}^{s=\tau(\vec{V}(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}}))}(\vec{x}'_{\text{proxy}})$
Vortex Force Mode	$\vec{\mathcal{F}}(\vec{x}'_{\text{proxy}}) = \ominus_{\vec{x}=\vec{x}'_{\text{proxy}}, \vec{q} = \frac{\vec{\varphi}}{ \vec{\varphi} }}^{s=\tau(\vec{\varphi})}(\vec{x}'_{\text{proxy}})$ $\vec{\varphi} = \left(\vec{\nabla} \times \vec{V}(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}}) \right) \times \vec{V}(\mathbf{T}^{-1}\vec{x}'_{\text{proxy}})$
Vortex Shape Mode	$\vec{\mathcal{F}}(\vec{x}'_{\text{proxy}}) = \bigoplus_{\vec{x}=\vec{x}'_{\text{proxy}}, \vec{q} = \frac{\vec{\varphi}}{ \vec{\varphi} }}^{s=\tau(\vec{\varphi})}(\vec{x}'_{\text{proxy}})$

Table 3.3: The haptic modes for vector data. Each function, $\vec{\mathcal{F}}(\vec{x}'_{\text{proxy}})$, represents the mode's force contribution to the balancing equation, as a function of the new proxy position, \vec{x}'_{proxy} . These modes provide haptic representations of the local vector data, \vec{V} , at the previous proxy position, \vec{x}_{proxy} .

The continuous data representation provided by this approach allows for the potential definition of surfaces at every location in the volume. Since there might be regions without surfaces, where the gradient operator yields a zero vector, the strength of the surface is modulated using the gradient magnitude. Thus, the continuous representation generates “transparent” regions, similar to visual volume rendering.

3.3.2 Vector Data Representations

Five haptic modes have been implemented for interaction with vector data.

Force Mode

This is the simplest example of direct force mapping possible: the direction and magnitude of the vector data are mapped to the direction and strength of the force feedback. As with any direct force mapping, this mode is easily implemented using a force primitive.

Follow Mode

For interaction with vector data, a haptic mode has been designed to provide resistance when moving the haptic probe perpendicular to the vectors in the field. This is the follow mode, a haptic interaction mode that is particularly suitable for flow, magnetic fields and similar data. The name of the mode is taken from the similar interaction mode used by Donald and Henle, in the work presented in [DH00], to provide interaction with animation data. Pao and Lawrence discuss a similar haptic interaction scheme in [PL98] which they call *virtual constraint*. The

major difference between these two modes, and the follow mode implemented using yielding constraint is that the latter is designed to yield to a configured force. Thus, it allows a user to move the haptic probe through haptic features, in this case perpendicular to the vector field, and so permits free exploration.

Using haptic primitives, the follow mode is as easy to implement as the force mode. Instead of a force primitive, the local vector is used as orientation for a line primitive. The line primitive provides a resistance when moving the probe perpendicular to the vector field, thereby generating the guiding effect and a sense of the magnitude of the field through the strength of the feedback.

The perpendicular resistance has two separate effects, apart from the physical guidance through the vector field that is used in the work mentioned above. The first is that the local shape of the flow can be perceived by moving the haptic instrument around in the local region. In this way information about the local shape of features in the data can be explored through palpation. The second effect is that when the haptic instrument is moved perpendicular to the field, variations in the magnitude of the vector field are clearly perceived through the variations in the resistance. These effects are discussed further in section 3.5.3.

Vortex Force Mode

While the force mode and follow mode directly represent the unprocessed vector data, the vortex force mode does further processing of the data to produce a force representation of the vorticity structures in the data. This mode is a reimplemention of the force function suggested by Lawrence et al. in [LLPN00] to obtain a direct force mapping from vorticity to force feedback. The vector $\vec{\varphi}$ of equation 2.9 points towards the centre of vortices, or towards the centre of rotation in regions of significant vorticity. The mode is implemented by controlling the direction and strength of a force primitive by this vector.

This haptic mode pushes the haptic instrument towards the centre of vortices in the data. This can both give a sense of the general vorticity of the volume, cues about where vortices are, and produce a guidance to help the user follow vortices through the volume.

Vortex Shape Mode

Using a yielding constraint with direction controlled by the same vector as is used in the vortex force mode, the same properties of the volume are, instead, represented by haptic shapes. This is the purpose of the vortex shape mode.

The vortex shape mode is implemented using a plane primitive. The primitive orientation is controlled by $\vec{\varphi}$ thereby converting the vorticity of the data into a haptic shape. This haptic mode produces a tube-like haptic representation of vortices in the data. Moving the haptic probe around the vortex, a shell is perceived surrounding the vortex core with a global shape reflecting the variations and structure of the vortex. Making use of the penetrability, the user can push through the outer shells of a vortex to explore also the inner structures of the same vortex. Since the equation 2.9 extracts only the local structure, incomplete vortices are also represented through the haptic feedback. A turning wind, for example, simply generates a local haptic shape of the bend.

Front Shape Mode

The follow mode provides a guidance and data representation *along* the vector field by controlling the orientation of a line primitive from the orientation of the local data. Using a plane primitive instead generates a representation of the data *across* the field. This is done by the front shape mode, producing a shape representation of the front orientation of the vector data at any position in the field.

For example of the feedback provided by this mode consider the exploration of a vector field representing the wind flow around an aircraft. As the wind splits to flow on both sides of the wing, the vector field diverges in front of the wing. This mode implicitly represents such a divergence by a ridge in front of the wing. Other features in the data are automatically mapped to various other haptic features, which can be both identified by the user and used as a guidance through the volume.

3.4 Material Properties

The yielding constraints representation of features in the data introduces a new and natural way to regard the properties of the haptic feedback — to handle them as material properties. This view of the properties in haptic algorithms has been used in volume haptics before [AS96], but not as extensively as introduced in papers C and F. Here the material properties not only specify how the data is perceived, but are also a means for configuring the haptic interaction algorithm and defining the representation of numerical properties in the data. The surface and friction mode for density data introduced, for example, friction as a representation of numerical data properties, but the strength of the palpated features is also a type of material property. In paper A the stiffness of the virtual coupling is also considered a material property since this can, to some extent, be perceived as the stiffness of the palpated feature, for example providing a means to differentiate between hard bone and soft skin tissues in CT. Through this conceptual approach, the stiffness parameter and the strength parameter of the yielding constraints are specified as representations of properties of the data. For example, the strength of the constraint used in the vortex shape mode can be used to effectively represent the magnitude of the vorticity, thereby providing in this case an intuitive connection between haptic distinctness and feature concreteness.

The material properties of the haptic interaction can be controlled using transfer functions, $\tau : \mathbb{R} \rightarrow \mathbb{R}$, making the adjustments to the haptic parameters in a way very similar to specifying the visual parameters of volume rendering. A separate transfer function is used to control each separate property of the haptic algorithm: the friction constants, the surface strength and the stiffness of the surface and friction mode, the viscosity property of the viscosity mode, and the strength and stiffness for the follow mode and the vortex shape mode. Some examples of this are shown in figure 3.9. By providing this natural connection between the concept of material properties of the haptic behaviour, not only is the haptic feedback more intuitively understood, but also more easily configured. The same principles that are used in volume visualization to define good transfer functions can be applied to the material configuration of the haptic interaction.

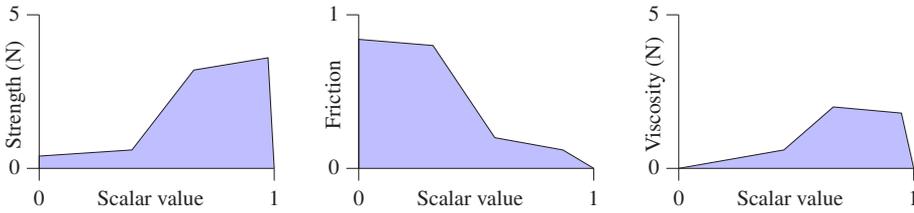


Figure 3.9: Examples of material transfer functions. These transfer functions control the material properties of the haptic modes, thereby constituting an intuitive interface for configuring the haptic feedback.

3.5 Characteristics

The introduction of yielding constraints as a representation of volumetric data combine the effects from two established methods for volume haptics. The concept of using shape representations for features in the data is taken from the geometrical constraints approach. The use of continuous representations of the data is similarly taken from the methods based on force functions. This is done with the hope that the best qualities of the individual original approaches will be combined into a new, more powerful, mode of interaction. This section discusses how the approach answers the demands of stability, accurate data reconstruction and perceptual effects.

3.5.1 Stability

The implementation of continuous shape representations is based on the same virtual coupling principle commonly used for haptic surface rendering. The stability criteria for such virtual coupling have been investigated, for example by Adams and Hannaford as presented in [AH99]. The approach to ensure passivity in the coupling approach is to adjust the stiffness and damper parameters to fit several properties in the interaction: hardware characteristics, sampling frequency and even how tightly the user holds the haptic instrument. Many of these properties are hard or impossible to accurately estimate or simulate, which makes automatic specification of the coupling parameters difficult. Generally, these parameters are estimated through the trial and error approach.

The yielding constraints are similar to the ordinary hard constraints, used for haptic surface rendering. They differ only in one major aspect: when a yielding constraint yields to a force that exceeds the strength of the constraint, the proxy is moved so that the virtual spring of the coupling is shortened. This means that energy in the system is absorbed, generating a damping effect. By providing this damping, the yielding constraints approach is inherently at least as stable as the hard constraints, thus conforming to the same stability measures and estimations with minor adjustments.

3.5.2 Data Reconstruction

It can be argued that the extraction of an explicit geometrical representation yields a far greater accuracy in reconstructing the features in the data than any DVH algorithm can ever reach. This claim is then based on the fact that surface reconstruction algorithms are capable of generating perfect edges, represented for example by the edge between two adjacent triangles, which is hard for DVH algorithms. The most common type of volumetric data, the sampled volumes (CT, MRI, CFD, etc), are frequency limited, so a sharp edge is more likely to be an artifact than an accurate reconstruction. In fact, a DVH algorithm is neither better nor worse at reconstructing the features in the data than the feature extraction method used. For example, the gradient operator, used in the case of the surface and friction mode, generally low-pass filters the data by applying a large kernel that smooths the local data. Alternative, frequency preserving methods can, however, be used. These can be based on quadratic filters or curve fitting amongst others (see [YCK92]).

The proxy movements that are used to control the haptic behaviour are performed for each haptic frame, generally at a rate of about 1 kHz. It can thus be assumed that there are only small changes between each new estimation of the proxy position. Since the constraint-based proxy movements are linear for each frame, the proxy movements constitute an Euler integration of position. Practical experiments suggest that the integral adequately approximates even sharp edges in the data. This has three possible reasons. First, the movements for each estimation of the proxy are generally very small, of the order of $10\ \mu\text{m}$. In fact, when the precision is most significant (in exploration of details) the user tends to move the haptic probe even slower. Volumetric data is also usually frequency limited, which means that the curvatures are generally not particularly sharp. Finally, the human sense of touch, or even vision, is not very accurate compared to the magnitude of the numerical errors involved. Thus, the errors must be relatively large for a possible deviation to be detected and have a negative impact on the guidance or experience.

3.5.3 Psychophysical Aspects

The use of shapes to represent features in volumetric data allows for the use of exploratory procedures, identified by Lederman and Klatzky [LK87]. In comparison, none of the identified procedures can be explicitly used for interaction with haptic feedback generated through direct force mapping. This is by no means a proof showing that direct force mapping is ineffective, however it shows that there are certain characteristics in natural human interaction through touch that can be associated with the means of interaction provided by the yielding constraint approach.

From the six exploratory procedures, described in section 1.2.1, at least three can be used together with haptic modes implemented using yielding constraints. The first applicable procedure is the *contour following procedure*. It is used to explore surfaces and shapes, and provides a sense of the local shape at the palpated region. This is the primary means for exploring feature shapes in the volume and is also applicable to geometry-based haptic feedback. Further, by connecting the data properties to haptic properties and allowing these to vary over the perceived shapes, a direct connection between micro variation and feedback is provided. This effect can then be used to explore the data through the *lateral motion procedure*, which is used to explore texture

properties. Finally, the shape penetrability, discussed in section 3.1.2, allows for perpendicular movements that can be used to explore distinctness and hardness properties of the features in the data. This is done through the third procedure, the *pressure procedure*. Studies have shown [TBS⁺06b, BSH⁺06, TBS⁺06a, HTB⁺06] that human ability to discriminate between force directions is poor, which agrees with this indication that shape representation can, in some cases, provide more effective haptic feedback than direct force mapping.

The study presented in paper **F** identified four general effects from adding haptic feedback in volume data exploration. The two primary effects are the physical guidance, working as a path towards interesting regions or as a physical support when a local region is examined, and the complementary haptic information, providing cues about non-visual properties such as friction or hardness. Defining complementary modalities as “a use of modalities where the interactions available to the user differ per modality”¹, this means that the haptic feedback complements the visual information by representing properties that are not represented through visual cues. The study, however, also showed that even if the haptic feedback generates cues for properties already represented by visual rendering, the combined effect was stronger than either modality alone. These are supplementary cues, where supplementary “describes multimodal applications in which every interaction (input or output) can be carried through in each modality as if it was the only available modality”¹. Finally the study identified a subliminal effect from adding haptic cues to the volume exploration — the user gains mental guidance, a psychological effect most likely connected to the intuitive use of touch as a way to orient in an environment.

These effects disregard the option for the feedback to be natural in its appearance, or abstract in its connection to the underlying data. Naturally, the more like the expected behaviour of the objects the haptic interaction is, the more intuitive the exploration becomes. On the other hand, reasonably, the more easily distinguishable different cues are and the more easily different force or strength levels can be discriminated, the more effectively the information can be perceived. This is a trade-off that requires careful tailoring of the haptic interaction to match the data type, data content, task at hand, and sometimes even the user.

¹Multimodal Interaction Requirements, W3C Note 8 January 2003

Chapter 4

Low Quality Data

This chapter describes methods developed to provide the special treatments required for data that is of a quality not sufficient for the implicit extraction of features. In haptic interaction with volumetric data with low signal-to-noise ratio, the data contains noise which leads to a poor haptic impression of the data. This is the case, for example, for low dosage CT data, but also for MRI data and others. Even moderate noise levels can have a severe impact on the haptic feedback, depending on the equations used to extract features from the data. The gradient operator is particularly vulnerable and since the proxy-based approach provides a dynamic control system, the interaction also accumulates error over time. The noisiness and poor contrast in MRI data thus makes the shape estimation uncertain.

MRI data also suffers from tissues having overlapping scalar ranges: the MRI scanner captures scalar properties in the tissues that are shared between different tissues, so that two or more tissues may fully or partially occupy the same scalar ranges. The classical transfer function approach is then incapable of differentiating between these tissues and some alternative or additional property must be used.

The work presented in paper **E** uses fuzzy classification data to enhance MRI data and thereby provide high quality haptic feedback. Separate surface information for shape rendering is extracted, as described in section 4.3, while the material properties are still defined directly from the MRI data, as described in section 4.2, see also figure 4.1.

4.1 Classification Enhancements

The method used in this work to enable haptic interaction with low quality data is based on classification enhancements. It makes use of a knowledge-based fuzzy classification scheme presented in [LLY05], which applies domain knowledge from radiologists on statistical neighbourhood analysis of the data to produce a *competitive classification certainty* for each voxel in the volume. This classification certainty is a value between -1 and 1 that signifies the certainty for class A or B, respectively. A value of zero thus signifies equal certainty for the two classes.

This classification method separates between two tissues with overlapping scalar ranges.



(a) The transfer function is used to define the material properties of the tissues in the volumetric data.



(b) A separate surface information volume is generated for high quality rendering of the tissue surfaces.

Figure 4.1: In interaction with low quality data the information is divided into separate material property estimation and surface data.



(a) The tissue mask for the liver tissue extracted from the MRI data.

(b) The tissue distance map calculated by applying the distance operator on the tissue mask.

Figure 4.2: 2D intersections of the 3D mask and distance map extracted for haptic interaction with MRI liver data.

These two tissues are then said to be part of class A or B. Any other tissues with different scalar ranges are separated using transfer functions and can be part of either class, or even both. Thus, the classification only separates tissues at the overlapping scalar ranges, and its behaviour at other scalar values is undefined.

4.2 Material Properties

To accurately define material properties, for example which tissue should produce haptic feedback and of what strength, the classification value must be used to separate between tissues with overlapping scalar ranges. To do this, two transfer functions are defined for each material prop-

erty, one for each class. The classification value is then used to select the right transfer function for each region in the data, or interpolate between them in uncertain regions. For example, the strength property, s , is estimated through

$$s(\vec{x}) = \frac{1 - C(\vec{x})}{2} \tau_{\alpha}^s(V(\vec{x})) + \frac{1 + C(\vec{x})}{2} \tau_{\beta}^s(V(\vec{x})) \quad (4.1)$$

where C is the classification volume, V is the MRI data and τ_{α}^s and τ_{β}^s are the strength property transfer functions for class A and B, respectively.

The transfer function pair for a certain material property differs at the scalar ranges occupied by two tissues. At any other scalar ranges, the transfer functions are set identical since the value of the classification volume for these scalar ranges is unpredictable. In this way the classification value becomes insignificant for these ranges.

With this approach, the material properties may still be specified through haptic transfer functions which enables a direct connection between the data and its haptic appearance. The shape information, however, needs to be separately defined, as is described in the following section.

4.3 Surface Information

The optimal solution in volume haptics is that the haptic feedback is provided at any position where the feature that the current haptic mode renders is identified. Because of the poor consistency of the implicitly extracted features in noisy data, this is not possible in interaction with MRI data. Therefore, an explicit intermediate representation of the surface data is required. In this method, the strength property is assumed to define the extent of objects that should produce haptic feedback. By thresholding the strength property, a binary mask is obtained that defines the haptic object, see figure 4.2(a). This mask can then be used to generate a smooth gradient by applying the distance operator, see figure 4.2(b). The resulting gradient from this procedure is suitable for, for example, the surface and friction mode.

The steps described above are performed separately for each tissue type, requiring a separate transfer function pair for each tissue. In this way adjacent tissues with similar strength, that would otherwise be part of the same binary mask region, will provide full surface information. The individual distance map volumes are recombined by simply accumulating their individual scalar values. Since the tissues in medical data are non-overlapping, the final volume contains no more than the individual distance maps in the same volume.

The final distance map volume contains a smooth gradient that can be used to provide surface information to the surface and friction mode. That mode use the gradient of the data to render a local surface. At the same time, the material properties are extracted as discussed above, providing tissue specific material properties of the MRI data, see figure 4.1. Since the distance map generates not only a thin surface, but a thick region of gradient, the approach allows the soft penetration of the haptic surfaces and so avoids occlusion.

Chapter 5

Dynamics in Time-varying Data

This chapter describes the methods required for algorithms rendering yielding constraints to produce a haptic feedback consistent with data that is moving through the virtual environment, and with data that in itself is varying over time. These methods were introduced in paper **H**. In any application for volume data exploration there will most certainly be the possibility to move the data. A two-handed operation interface can, for example, provide a space mouse for the non-dominant hand and a haptic instrument for the dominant hand. In this type of setup the data can be interactively scaled, moved or rotated to show an arbitrary view, while still allowing normal haptic interaction with the data through the haptic instrument. Without the explicit handling of the movements caused by the changing transform, an object may freely move through the haptic probe without generating any haptic feedback, which is counter-intuitive and confusing.

More importantly, however, there is an increasing amount of dynamic data being simulated and captured, such as animated CFD showing the propagation of a shockwave or of vortices around an aircraft, or time sequences of CT data showing the dynamics of the valves in a human heart. In haptic interaction with such data, the changes over time must be fully captured if the dynamics is not to be lost. The dynamics may be of utmost importance and haptics has here a great potential to facilitate and improve the understanding of the dynamic behaviour, a potential that so far has not been utilized.

The force function-based methods for volume haptics simply convert the data at the probe position into a feedback force. It has no distinct position in the data and treats the matter represented by the volume as a shapeless information cloud. By not having a distinct point of interaction, these methods avoid the need to explicitly handle dynamics. On the other hand, the feedback is not very distinct and does not reflect the dynamics of the changing data. The constraint-based methods introduced in the current work, however, use a proxy point as a memory from the previous calculations of the haptic feedback to provide distinct rendering of shapes in the volume. Thus, if the data changes or moves, so that the shapes and objects represented by the data change location over time, the haptic feedback is intuitively expected to follow these changes. The methods presented in the previous chapter, do not correctly handle these changes over time. Since the haptic feedback is calculated at discrete occasions during the haptic simulation, changes between

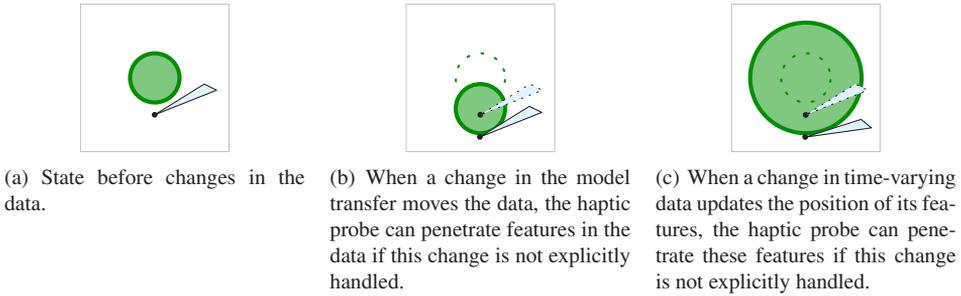


Figure 5.1: In haptic interaction with data that change over time, the haptic probe should reflect this change. The behaviour without explicit handling of dynamics of data is shown as a dotted stylus, while the behaviour with the proposed method is shown as a solid stylus.

these calculations must be explicitly considered for the correct haptic behaviour to be rendered.

5.1 Proxy-point Updating

In the constraint-based approach for volume haptics, the proxy point is used for each calculation of the haptic feedback as a memory for the point of palpation from the previous calculation. If there have been changes since the last estimation where the proxy position was determined, this position will be out of date and no longer correctly represent the point of contact with the data (see figure 5.1). By processing the proxy point in response to changes in the data prior to using the proxy point as a memory in the haptic algorithm, this memory is made up-to-date which produces a correct haptic response to these changes, see figure 5.2.

How the proxy position is made up-to-date is dependent on the representation of the dynamics of the data. The current work identifies and handles two different types of change, and thus moves the proxy separately to respond to these:

- changes in the model transform of static volumetric data — movement of the data with time
- time-varying volumetric data, in the form of sequences of volumes — changes within the data with time

Real-time simulated data is, here, considered a special form of the latter case, a sequence of volumes where the future volumes are generated at run-time and obsolete volumes are continuously discarded. Both these processes are precisely timed using the world clock for the model transform dynamics, and a high precision animation timer for the changes in time-varying data.

With this extra proxy processing step, the full haptic estimation procedure becomes:

1. Move proxy to reflect changes in model transforms

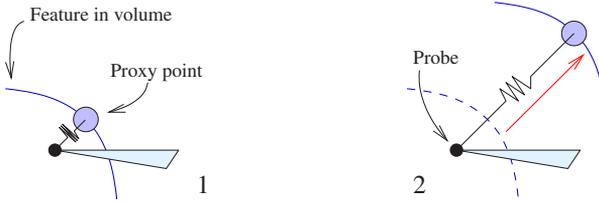


Figure 5.2: As the features in the dynamic data move, the point of interaction, the proxy, must be moved in response to this change in order for the feedback to correctly reflect the dynamics of the data.

2. Move proxy to reflect changes in time-varying data
3. Move proxy to generate feedback from interaction with features in the data
4. Estimate feedback using coupling equation

With the primitives-based approach to volume haptics several modes can be used simultaneously at different positions in the virtual environments, and using different data. Since each haptic mode then may be subject to different changes in the data and the model transform, the proxy point update must be performed individually for each mode. This mode-specific proxy position is used by the mode to determine primitive positions and to extract data. When the force balancing has been performed, the one single proxy point is determined again, representing one single palpated position.

5.2 Dynamic Transforms

With dynamic transforms the changes over time are implicitly represented by the transform matrix. The proxy-point position is brought up to date by applying these changes. Having the previous and current model transform, T_p and T_c , the new proxy position is determined by setting the proxy position equal in the local space in the previous and current transform,

$$\mathbf{T}_c^{-1} \vec{x}'_{\text{proxy}} = \mathbf{T}_p^{-1} \vec{x}_{\text{proxy}} \quad (5.1)$$

That equation can be used to define a new proxy position, \vec{x}'_{proxy} ,

$$\vec{x}'_{\text{proxy}} = \mathbf{T}_c \mathbf{T}_p^{-1} \vec{x}_{\text{proxy}} \quad (5.2)$$

that represents the palpated position after the matrix change.

The transform is, in most visiohaptic systems, controlled in the graphics thread at a much lower rate than the asynchronous haptic feedback estimation. To get smooth haptic feedback from transform changes at a lower rate, haptic interpolation is required. The last and the current model transforms are then used to estimate an interpolated transform, through linear interpolation for the scale and transform components, and through spherical linear interpolation (SLERP) for the rotational component. The most current interpolated transform and the previous interpolated transform are subsequently used in equation 5.2 to update the proxy position.

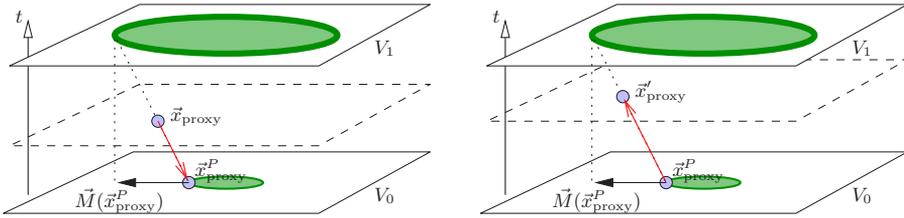


Figure 5.3: The volumes in a sequence representing a time-varying data set can be viewed as hyperplane samples in a 4D space-time continuum. Since the palpation is performed between these planes in time, the proxy must first be back-projected onto a volume to extract data (left). The data at that position can then be used to find the motion vector to move the proxy forwards to its new position between the hyperplanes, as specified by the timer.

5.3 Time-varying Data

In contrast to the handling of the changing model transform described above, the handling of time-varying data suffers from the lack of an implicit definition of the changes in the data. To update the proxy-point position in response to changes in the data, an explicit representation of these changes is required. This is provided by a motion field, \vec{M} , that, with a vector for each position in each volume, describes to which position the features at that position have moved in the following volume. This field can, for example, be automatically estimated from the sequence of volumes or may be readily available from the source of the data sequence, for example from CFD.

For smooth haptic interpolation between volumes in the data sequence, the animation playback needs to be controlled by a floating point timer. The integer part of this timer specifies the current volume to visualize, while the fractional part specifies how far between the current and the next volume the haptic simulation currently is. This timer does not have to be correlated with the world clock and can, therefore, play at any desired animation frame rate.

5.3.1 Proxy-point Movements

The data volumes of a time sequence can be considered hyperplane samples in a four dimensional space-time continuum, see figure 5.3. At any instant in time the haptic interaction is performed in between two such hyperplanes, in a region in which no data is available. The animation timer specifies the location in time, while the 3D proxy position specifies the spatial coordinates.

To be able to extract data for controlling the haptic feedback, the data must be obtained by first back-tracking the proxy point into the previous volume, producing an in-data proxy-point, \vec{x}_{proxy}^P . This is done by projecting the proxy onto this volume along the motion vector of the motion field of this volume. This projection is expressed as

$$\vec{x}_{\text{proxy}}^P + \gamma \vec{M}(\vec{x}_{\text{proxy}}^P) = \vec{x}_{\text{proxy}} \quad (5.3)$$

where γ is a value between zero and one describing the time position of the proxy, that is the fractional part of the animation timer when the current proxy position was determined. The projection equation is easily converted into a fixed-point problem. As such it can be solved by iteratively applying its fixed-point form, starting at some initial position such as the last used in-data proxy position.

From the obtained in-data position the data to control the haptic feedback can be extracted, as well as the local motion field data that defines how that position is changed between the current and the next volume. Using this latter information, the proxy is then moved forward to its new between-data position in 4D space, one step closer to the next volume in the sequence as specified by the current fractional part of the animation timer. After this, the proxy is up-to-date and the selected proxy-based haptic algorithm can be applied to generate the haptic feedback.

5.3.2 Estimating the Motion Field

The motion field required for the haptic rendering of dynamics in time-varying data describes the movements of features in the data. In volumetric data, however, there is no unambiguous sense of features. Each haptic mode may produce a haptic representation of different features in the data and those features may move differently between two volumes in the same time-sequence. Thus, at the time the motion field is defined, this must be generated with respect to the features represented by the designated mode, and multiple haptic modes working on the same sequence of data may require individual motion fields. These are aspects that need consideration when extracting the motion field for interaction with time-varying data.

The motion field describes the difference between two volumes, a difference that is expressed both by algorithms for optical flow, primarily developed for motion estimation in computer vision, and registration algorithms, used to co-register different data in medicine. These algorithms aim at minimizing the error, ε , in

$$\varepsilon^2 = \left\| V_1(\vec{x} + \vec{M}(\vec{x})) - V_2(\vec{x}) \right\|^2 \quad (5.4)$$

that is, find a deformation that projects one data set into the other. Any algorithm that defines a dense field, that is a field that defines the difference for every position in the volume, can be used to estimate the motion field. The algorithm used in this work is the Demons algorithm presented by Thirion in [Thi98].

5.3.3 Conditions

The back-tracking of the proxy point is an ill-posed problem. It can potentially have multiple solutions, and iteratively applying the fix-point form can diverge instead of converge to a single solution. The Banach fixed-point theorem (1922), however, shows that the correct and unique solution to the back-tracking will be found, as long as the following condition is met:

$$\left| \vec{M}(\vec{a}) - \vec{M}(\vec{b}) \right| < \left| \vec{a} - \vec{b} \right| \quad (5.5)$$

where \vec{a} and \vec{b} are points in the region around the correct solution. The local region for which the condition must hold is defined by the initial estimate, the in-data proxy position and the immediate neighbourhood, being the region containing the intermediate values obtained while solving the fixed-point problem.

This indicates that the temporal resolution of the animation has to be high enough relative to the dynamics in the data. With a higher temporal resolution the magnitude of \vec{M} will be smaller, with the consequence that the left hand side of equation 5.5 is likely to become smaller. Low dynamics in the data has a similar effect.

Chapter 6

Volume Haptics Toolkit

The haptic rendering technology based on haptic primitives has been implemented into a general toolkit for volume haptics — the Volume Haptics Toolkit (VHTK). The structure and technologies used in this toolkit, and the implementation details are presented in paper F. This chapter provides an overview of the functionality of this toolkit and examples of demonstration applications built with it.

During the development of the haptic primitives and the related technologies presented earlier in this thesis, VHTK has matured to a point at which it can be widely distributed. It has so far been used primarily as a platform for further research, but has also lately received some attention outside the research community. The toolkit has been made available for free download under the GNU General Public License in cooperation with SenseGraphics AB.

6.1 Implementation and Technology

The current version of the Volume Haptics Toolkit is implemented using the H3D API from SenseGraphics. H3D is an open source and cross platform scene-graph system for multimodal applications, primarily for 3D graphics, haptics and sound. The structure of the system is based on the X3D standard, which is an XML-based scene-graph and virtual reality world definition file format. X3D is also the standard file format for loading and exporting scenes. Some simple animations can be defined through X3D but more advanced behaviour in the scenes is programmed using the Python scripting language. In H3D low-level nodes and handling is implemented using C++. The system is designed to run on a semi-immersive stereoscopic display with co-located and co-registered haptics and graphics display, such as the Reachin Display or the SenseGraphics IW.

VHTK extends H3D by adding the nodes needed for producing haptic interaction with volumetric data including visual feedback, such as haptic nodes, visualization nodes, data container nodes and data processing nodes. The nodes provided by the toolkit are implemented in C++ but can be used and controlled also from X3D and Python, allowing a programmer to build the application in X3D, Python or C++ or a combination thereof, or even extend the toolkit further

using C++.

X3D has an even handling system based on routing between fields in the scene-graph. VHTK makes full use of this system to provide run-time updates when requested data becomes obsolete due to changes elsewhere in the scene-graph. This facilitates the rapid development of highly interactive visualizations.

6.1.1 Haptic Nodes and Rendering

VHTK uses the haptic primitives as the core technology for the rendering of haptic feedback from volumetric data. The primitives approach encapsulates the selection and configuration of haptic primitives into haptic modes. In VHTK these haptic modes are implemented as scene-graph nodes that can be placed in the hierarchy of transforms to position, scale and rotate the haptic representation of volumetric data. The haptic mode nodes are selected and initialized, primarily when configuring the the application scene-graph through X3D. Here, the data sets for which the modes should generate feedback are assigned to the modes, and transfer functions are selected and configured to control the material properties of the data. During run-time these properties can be modified from both Python and C++, and through the X3D routing system.

The toolkit provides nodes for all haptic modes listed in section 3.3. Each mode node provides the mode-specific configuration interface through the X3D node interface, based on the material properties for the mode. The material properties are controlled using transfer functions, as described in section 3.3.

6.1.2 Visual Components

VHTK also provides some scene-graph nodes for visual representation of the volumetric data. These are designed to generate a visual representation in the same position, size and orientation as a haptic mode for the same data and transform, so that a haptic mode and a visualization node sharing a parent transform and data to render will produce co-located and co-registered haptic and visual representations.

An isosurface node is provided that uses the Marching Cubes algorithm to generate a surface at a specified isovalue in a voxel-based volume. The field that specifies the isovalue of the surface is also a trigger for re-estimation of the surface, so that when the value is updated the polygons of this node are regenerated. A volume renderer node implements a simple 3D texture-based proxy geometry renderer, that can either render colour textures or apply transfer functions to colour classify scalar textures. The look-up table for the renderer is automatically updated if any parameter in the specified transfer functions changes and if any parameter changes, for example the look-up table, the number of planes to render or the 3D texture, the renderer updates the visual image.

For the visual rendering of vector data, the toolkit provides stream-ribbons and stream-tubes, generated by separate nodes with similar interfaces. Both these nodes take a list of points as seeds for forward and backward positional integration through a specified vector volume to generate a representation of the structure of the vector field. The colour is controlled through transfer

functions from either a separate scalar volume or from the vector magnitude, and the radius of the stream-tube can be used to represent a separate scalar volume or property.

6.1.3 Data Processing and Miscellaneous Nodes

In addition to the nodes for visual and haptic rendering of data, VHTK provides a list of nodes for data processing and other miscellaneous nodes. Data filters are provided for cropping, scaling and converting the volumetric data, and for extracting properties such as the magnitude of a vector volume as a scalar volume, or the curl as a vector field. These are all connected to the X3D event system so that if any parameter that affects a data source or the result of a filter changes, the volume generates an event that updates any dependent consumer.

The toolkit also provides several types of transfer function nodes, since these are used quite extensively in the configuration of the visual properties and in the specification of haptic material properties. Each transfer function node has a unique interface for specifying the function, such as the parameters level and contrast for one node, or a set of sample points for another. Colour transfer functions can be specified either by composing three transfer functions each of which specifies a separate component — for example red, green and blue, or hue, saturation and brightness — or by automatically generating a temperature scale in a specified range.

6.2 Demonstration Applications

VHTK has been used to create several application prototypes to demonstrate both the functionality of the technology introduced through the work presented in this thesis and also the power and use of haptics in volume visualization. These applications are described in their respective context in papers **B**, **C**, **E**, **F** and **H**.

All but one of the applications have been implemented without low-level programming, by scene-graph design in X3D and Python scripting for user interfaces and for the interactive distribution of stream-ribbons. The application described in section 6.2.4 below makes use of low-level processing and requires C++ programming and compilation.

6.2.1 Dichloroethane

This demonstration application was implemented as a simple but powerful demonstration of the functionality of volume haptics. Here, the simplified electro-potential field of a dichloroethane molecule ($C_2H_4Cl_2$) is explored through multimodal visualization. The scene-graph used to build this visualization is shown in figure 6.1(a) and the result can be seen in figure 6.1(b).

The electro-potential field of the molecule and its gradient are simulated using an analytical volume data node that is available in the toolkit. The data, V , and the gradient, $\vec{\nabla}V$, of this

volume are expressed as

$$V(\vec{x}) = \sum_i \frac{p_i}{|\vec{x} - \vec{x}_i|} \quad (6.1)$$

$$\vec{\nabla}V(\vec{x}) = - \sum_i \frac{p_i (\vec{x} - \vec{x}_i)}{|\vec{x} - \vec{x}_i|^3} \quad (6.2)$$

where p_i and x_i are the charge and position of the i th atom in the molecule. The follow mode is applied to the gradient of this data, to produce a feeling of the strength of the electro-potential at any position in the visualization, and a sense of the orientation of the field.

The primary visualization of the molecule is a ball and stick model built using the ordinary geometries available in X3D. To give a global representation of the electro-potential field, a second volume node is used to generate a scalar electro-potential field for visual rendering. This node is converted into a 3D texture using a rasterizer node, and used as source in the volume renderer. Transfer functions are applied to control the colour and opacity so that positive potential is rendered red, negative blue and zero is green. Observe that since the volume is rasterized for visual rendering, the visual impression is limited in space, but the haptic feedback is generated in interaction with an analytical data set which is without boundaries.

The rasterized volume is also used to generate isosurface for the electro-potential. These are rendered semi-transparent and show the shape of the iso-potentials inside the size of the rasterized volume at potentials of -1 , 0 and 1 .

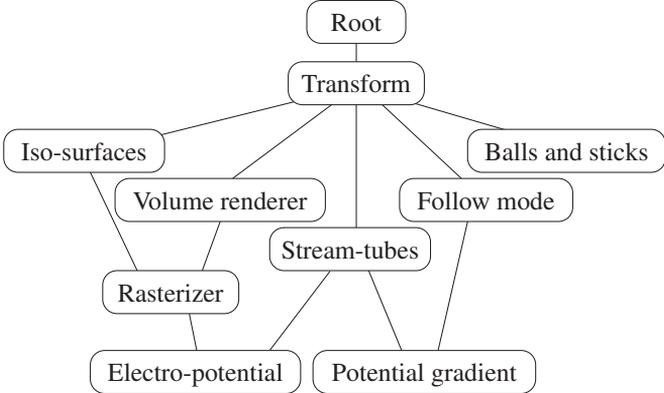
Using a Python script that is a part of the toolkit, the haptic instrument is connected to the seed point list of a stream-tube visualization node. This node renders stream-tubes through the gradient data set to visualize both the structure of the field and also its strength, through the radius of the tube. Here transfer functions are used to control the colour of the tubes as a function of the magnitude of the gradient field. A simple temperature colour scale is used so that high gradient magnitudes are rendered red or white and low magnitudes blue or black.

6.2.2 Sharc UAV

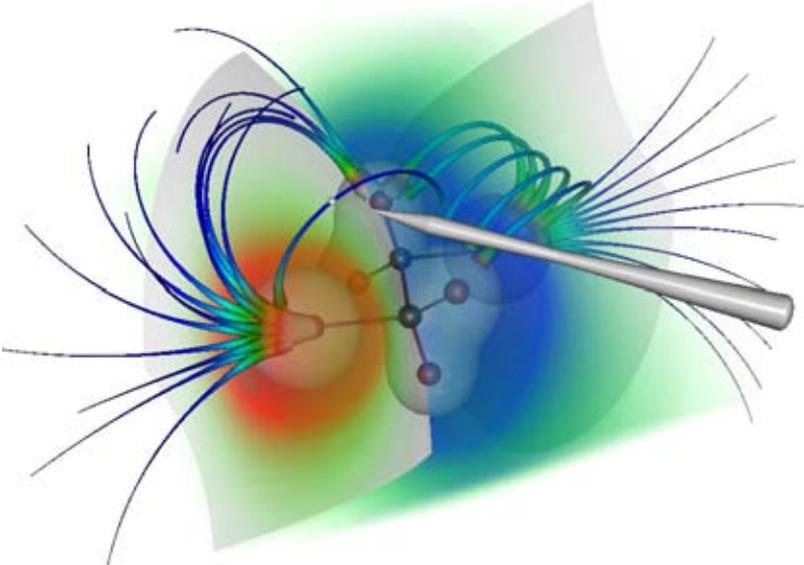
The SHARC aircraft is an experimental unmanned aerial vehicle (UAV). In this demonstration application the data from a CFD simulation of the air flow around the aircraft is explored using multimodal interaction, see figure 6.2. While only simple properties, or data at discrete positions, can be rendered visually without cluttering the display, interactive stream tubes and haptic feedback can be used together to freely explore the full 3D volume.

Haptic interaction with the air pressure is enabled by applying the viscosity mode to the scalar pressure data set obtained from the CFD simulation. This haptic mode can be used individually, or in combination with one or several of the haptic modes used to represent the air flow data.

Three different haptic modes have been shown to work well with the vector flow data: the follow mode, the vortex shape mode and the front shape mode. The follow mode conveys the flow orientation and its strength throughout the volume. By palpating the wind with the haptic instrument an immediate impression of the local configuration of the flow is obtained. This feedback also makes it easy to follow an identified path of wind both forwards and backwards,



(a) The scene-graph of the application.



(b) The visual appearance of the application.

Figure 6.1: The dichloroethane molecule visualization demonstration application. Here, the follow mode provides haptic feedback from the gradient of the electro-potential field.

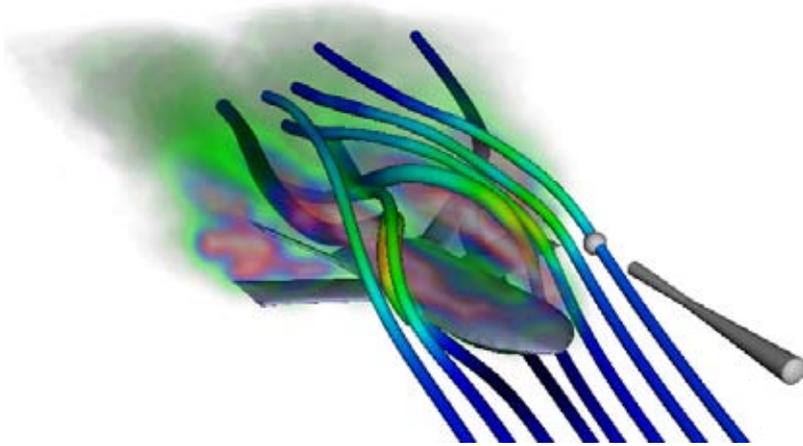


Figure 6.2: The Sharc UAV is visualized using a combination of X3D geometry for the aircraft, volume rendering for the curl of the air flow and stream-tubes for the air flow and its velocity. The haptic feedback is generated from the air flow using the follow mode, the vortex shape mode or the front shape mode, or from the air pressure using the viscosity mode.

for example allowing the user to track the cause for vortices and ease the movement between regions of cause and effect.

The vortex mode produces a haptic surface effect from the vorticity of the vector field. To avoid confusing effects from noise in the data, a threshold of the vorticity is set through the material properties of the mode so that only strong vorticity generates feedback. The generated feedback guides the haptic instrument so that it is easy to find and follow vortices and areas of strongly rotating turning air flow. At and around vortices the impression of a haptic tube is generated, a tube with the extent and shape of the vortex.

While the follow mode generates a haptic guidance *with* the orientation of the vector field, the front shape mode generates a guidance *across* the field. This feedback gives the impression of bumps or ridges where the air flow splits or diverges, such as in front of the wings. The ridge in front of the wing, for example, can be used as a physical support and guidance when releasing stream-tubes around the wing, or to probe the shape of the bifurcation of the air flow.

6.2.3 Doppler MRI Heart

Modern MRI-scanners are capable of acquiring animated blood-flow data from within a beating human heart. The phase contrast pulse sequence used to acquire this kind of flow information produces poor tissue contrast when used to scan full 3D data, so the visual quality of the data set is low. Both the poor tissue contrast and the fact that the noisiness of MRI data makes automatic extraction of features difficult, makes this an interesting target for multimodal interaction methods.

This application demonstrates the value of haptic feedback in the exploration of this type data, see figure 6.3. The haptic feedback helps the user to understand the data and identify blood flow patterns, while the visual volume rendering only provides a coarse sense of orientation. The user can interactively release stream-ribbons in the volume to mark and further visualize the identified flow patterns.

The task of exploring this kind of data is effectively guided using the follow mode, applied to the blood-flow data. This haptic mode provides guidance and information about the local flow. The characteristics of the flow can be recognized through the haptic feedback generated by this mode, and can also be distinguished from noise. This also demonstrates the effectiveness of the haptic perception and what potential it has in visualization. By also adding the force mode that pushes the haptic probe in the direction of the flow, an extra channel of information about the anatomy of the heart is provided.

The gradient mode can also be applied to the flow magnitude, extracted through a filter provided by VHTK. This produces a push towards strong flow which makes it easier to find main blood flows and follow the major vessels. The pushing effect, however, also obfuscates detailed information from the other modes, which makes this mode unsuitable for close examination of the identified flow.

6.2.4 Classification Enhanced MRI

MRI data has low signal-to-noise ratio and also suffers from overlapping scalar ranges for multiple tissues. To produce high quality haptic feedback from such data, it must first be enhanced using classification information, as described in chapter 4. This application demonstrates the use of knowledge-based fuzzy classification to enhance the surface information and material properties of tissues of and around a liver tumour in MRI data. The classification data is generated by a pre-processing filter which produces a competitive classification certainty value describing the certainty for a point in the volume to be a part of tissue class A or B, respectively. This volume is used both to extract surface information of tissues in the data in a second pre-processing step, and to determine the material properties for the different tissues.

The images in figure 6.4 show the visual impression of the MRI data before and after applying the classification. The haptic feedback is generated by a modified surface and friction mode that makes use of the separate surface information, and of the classification data for material property estimation. Without applying the classification data, the haptic feedback is generated from both tumour tissues and surrounding tissues, and is filled with haptic artifacts caused by the noise in the data, that makes any feature in the data undetectable. With the classification data and the proposed enhancements described in chapter 4, the haptic effect is fully controlled and generates smooth haptic feedback from only the tumour tissues.

6.2.5 Time-varying CT Heart

This application demonstrates haptic feedback that reflects the dynamics in time-varying data, see figure 6.5. Here the user can palpate both the surface and the interior of a beating human heart and feel the dynamics of the heart tissues through a full heart cycle. This data set, acquired

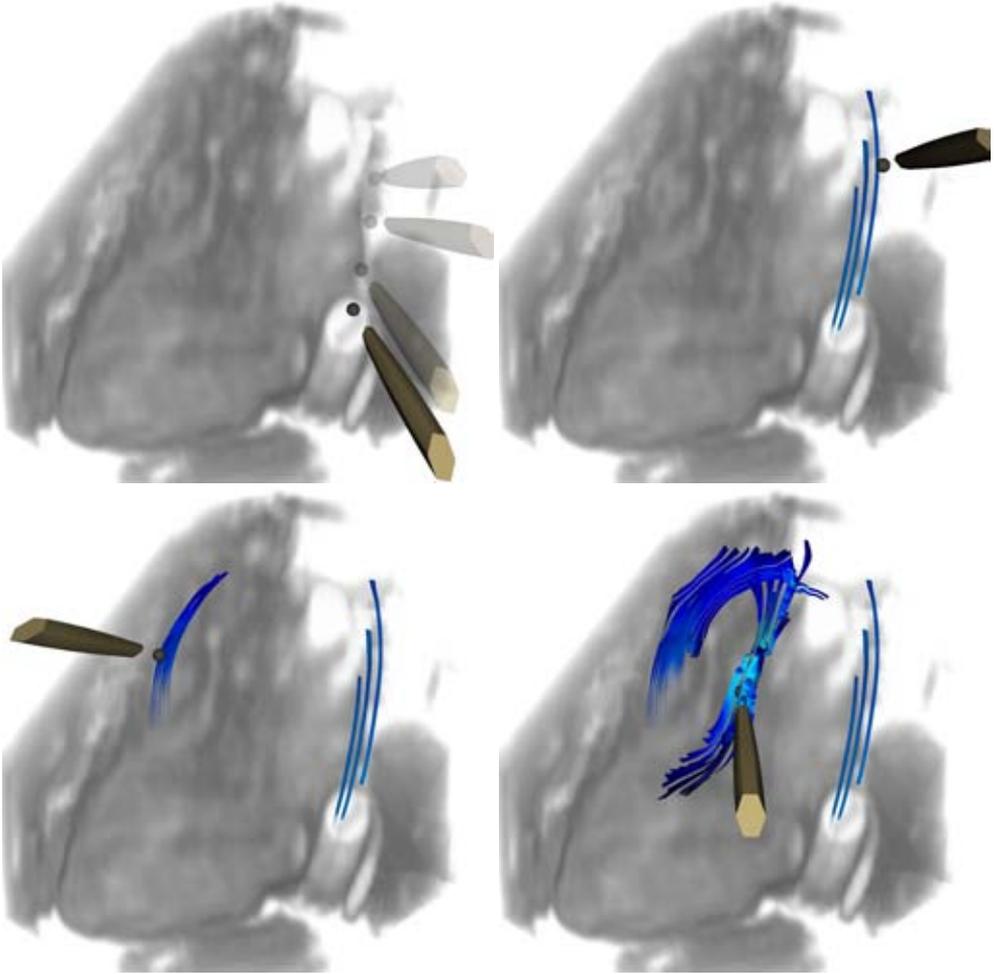
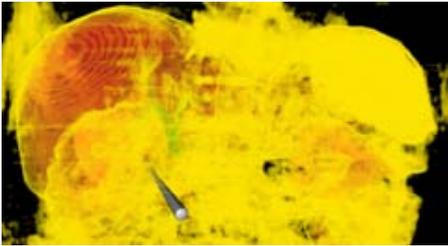
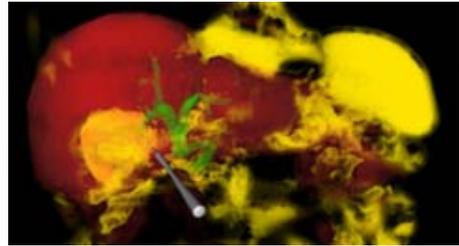


Figure 6.3: In this Doppler MRI exploration application, the user makes use of the haptic feedback to find and explore blood flow paths in a human heart. The poor contrast of the Doppler MRI data makes the haptic feedback an important tool for understanding the anatomy and inner structure of the heart.



(a) Without classification the tissues surrounding the tumour (yellow) and much liver (red) are occluded by the surrounding tissues.



(b) The surrounding tissues are separated using the classification, which reveals both liver (red) and tumour (yellow) tissues.

Figure 6.4: This visualization demonstrates the need for classification for tissue separation when there are overlapping scalar ranges in the data.



Figure 6.5: This application demonstrates the haptic interaction with time-varying data. The surface and friction mode allows the user to palpate the beating surface of the heart and also to push through that surface to also examine the interior chambers.

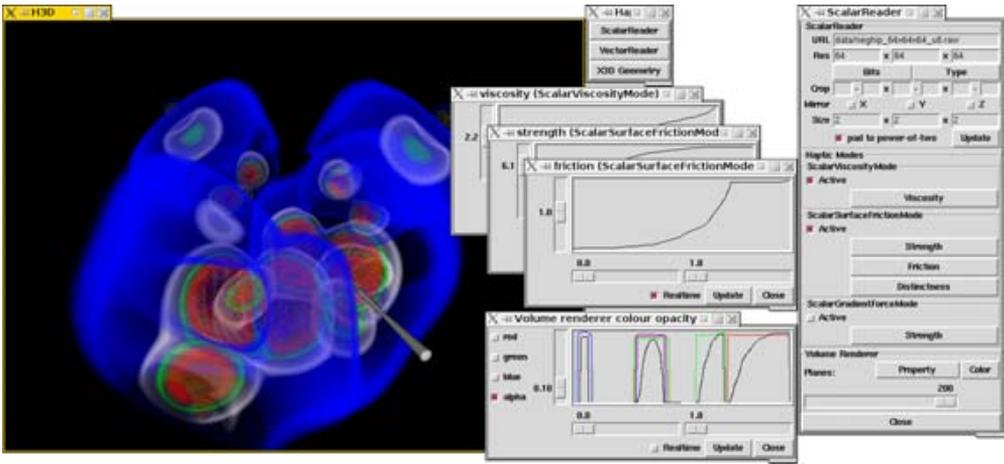


Figure 6.6: This application demonstrates how a graphical user interface can be built that provides real-time control over the haptic and visual parameters of a visualization. The user interface shows free hand drawn transfer functions for viscosity haptic and the surface and friction mode, and for the volume rendering. The data set shows a simulation of the spatial probability distribution of the electrons in a high potential protein molecule, obtained from the VolVis distribution of SUNY Stony Brook (see <http://www.volvis.org>)

through CT synchronized with EKG, is of 10 frames for the full beat cycle with $512^2 \times 400$ voxels size and a resolution of about 0.5 mm/voxel. The exact heart rate at the acquisition is not known, but a typical rate of 60 heart beats per minute is obtained by running the animation at 10 frames per second. The motion fields required to explicitly define changes through the animation were estimated using the implementation of the Demons algorithm [Thi98] that is available in ITK [ISNC03].

The visual rendering of the data set is performed using the volume renderer available in VHTK. Each volume frame is rendered by a separate volume renderer instance and using a switch node, which is a standard X3D node, the system switches between visible frames at runtime. A timer is used to control the haptic animation and interpolation. The floating point frame pointer value of this timer is converted into an integer specifying the frame to show by simply removing the fractional part.

For haptic feedback the surface and friction mode is applied. This mode extracts the gradient vector from the scalar data and uses this as an estimation of a local surface. The effect is the feeling of surfaces at locations where the scalar value in the CT data changes, typically at borders between tissues. The haptic feedback makes it possible to touch and track by touch the moving heart walls, and feel their movements. Touching the heart from the outside, the beating surface can be freely palpated and, by simply applying a force exceeding the strength defined for the heart muscle, the wall can be penetrated and the inside of the heart chamber can be examined.

6.2.6 Volume Data Exploration Tool

To demonstrate the flexibility of VHTK a scripted graphical user interface (GUI) has been built for rapid prototyping of data visualization, see figure 6.6. The GUI is generated using Python through the user interface programming package, Tk. It provides raw data loading functionality and enables basic visualization, the selection between various haptic modes and the visual and haptic configuration and fine tuning. The application allows the user to simultaneously load several data sets and apply individual haptic and visual settings. It is also possible to load geometrical models into the scene.

When a data set is loaded, every haptic mode supporting that particular type of data is added to the scene-graph of the system, but initially deactivated. Through the user interface the individual modes are then configured and activated. The configuration of the modes is performed through free hand drawing of the transfer functions that control the mode specific material properties.

Each loaded volume is visualized using volume rendering. The visual transfer function for the visualization is controlled through free hand drawing, just like for the haptic modes. The user can also select which property from the volume the visualization should represent (scalar value, gradient magnitude, vector magnitude, curl magnitude, vector divergence) and the number of planes to use in the proxy geometry.

The widgets in the Tk GUI send events to the Python script for example when buttons are pressed or transfer function drawn. These events are connected to the H3D system so that pushing a button, for example, produces an event that propagates from the Tk system over to and through the H3D system. Thus the application is real-time interactive and produces instant visual and haptic response to changes in the GUI.

The example in figure 6.6 shows a simulation of the spatial probability distribution of the electrons in a high potential protein molecule. The data was obtained from the VolVis distribution of SUNY Stony Brook (see <http://www.volvis.org>). In this example two haptic modes are activated: the surface and friction mode, and the viscosity mode. The figure also shows the transfer functions for viscosity, surface strength, friction and colours.

Chapter 7

Conclusions

Part II of this thesis gives an overview of the work presented in the included publications. In this chapter these contributions are summarized and some of the key conclusions from the results of the research are presented. The last section then discusses the most important topics for future research in volume haptics for visualization.

7.1 Summary of Contributions

Previous haptic representations of volumetric data have either been in the form of geometrical constraints or of force functions. The geometrical constraints provide a distinct representation of the data, but only for the local subspace defined by the geometry. These impenetrable geometries also introduce occlusion in the exploration environment. The methods based on force functions on the other hand provide a haptic representation of the data at any position in the volume but do not provide distinct impressions of the data.

This work has introduced the concept of yielding constraints as a way of providing haptic shape representation of features in volumetric data without limiting the rendering to a pre-selected subset of the data or introducing occlusion. By using shape representation of the data some natural exploratory procedures, as identified by Lederman and Klatzky, can be used in the palpation of the data. Different data can be rendered in different ways to fit the represented contents, the task at hand and user preferences. A wide range of possible haptic modes are provided from which to select. To provide an intuitive and highly configurable haptic interaction the properties of the haptic modes are defined by an abstraction of material properties, each haptic mode having a set of parameters that can be configured both in pre-rendering and at run-time.

This work also introduces an approach for implementing yielding constraints based on haptic primitives, each of which applies a constraint in one or more dimensions. This approach enables intuitive implementation and deployment of haptic modes, and free combination of any number of haptic modes. The algorithm has been packaged and is available for free download in the form of the Volume Haptics Toolkit.

Methods have been developed that also enable haptic interaction with data that has low signal-to-noise ratio and even data with overlapping scalar ranges for different tissues. For these data, knowledge-based fuzzy classification is used to separate different tissues, and a separate shape information volume is extracted in a pre-processing step to enable high fidelity feedback from the noisy data.

The final contribution of this thesis shows how the dynamics of moving data and time-varying data can be reflected through the haptic feedback to enable better understanding of volumetric dynamics in multiple disciplines, such as aerodynamics and medicine. This support for dynamics in the data is implemented by updating the proxy point representing the position of palpation in response to changes in the data prior to applying the haptic rendering algorithm in each time-frame. This pre-calculation of the proxy position first takes into account changes in the model transform and then changes in the time-varying data.

The applicability and versatility of the proposed methods are demonstrated in a wide range of example applications taken from the very different areas of medicine, molecular physics and aerodynamics.

7.2 Conclusions

Both previous research and the results from studies in the presented work demonstrate the potential for the use of haptics in volume visualization. They also show that volume haptics is an area not fully investigated and that there are still many unanswered questions. Some potential gains and important issues for the technology have been identified and, more importantly, aspects that need special consideration. Nevertheless, there is still scope and potential for improvements.

This work has established shapes as an important concept for the generation of representations of volumetric data. These provide a more natural connection to human perception than the earlier force functions which have been a popular way to produce haptic feedback from volumetric data. The haptic primitives provide an easy-to-use but still powerful interface for developers, enabling easy design and implementation of new haptic modes for new data or tasks. The technology is powerful and versatile, and is now capable not only of reflecting various features, but also dynamics in time-varying data and changes due to movements of the data for instance in a two-handed interface.

Easy deployment also means potential availability for industry. The haptic algorithms enable easy selection, tailoring and fine tuning of haptic modes for each specific data set, task and even user. This supports a wide range of possible applications and potential uses. There is, however, still a long way to go to achieve wide spread industrial use. Even though the haptic technology is becoming mature, the purely visual tools and user interfaces of commercial products are still far more advanced. The current implementation of the presented methods should thus be seen as a reference package and the haptic interaction must, in the future, be built into established systems. Further research efforts are needed to convince industry that the value of haptics supports this. Introducing haptic interaction in industrial applications generally means not only buying a haptic instrument and adding haptic algorithms to an existing program, but also to change the user interface from a mix of 2D and 3D to a fully or semi-immersive environment. A natural path for

introducing haptics is to first offer haptic augmentation in selected sub-tasks, but the full power of virtual reality and haptics can only be released in a haptic environment which supports the full work flow.

7.3 Future Work

More studies are needed to further investigate the nature of the haptic understanding of abstract volumetric data and how to master the identified issues and aspects of haptic visualization. We need to try and understand how haptic guidance and information support better orientation in data and understanding of its nature, and also what haptic modes are suitable for the specific tasks at hand. For example, how is the optimal haptic mode selected for initial exploration, and when and how should another mode be selected for further investigation, if at all? There is also a potential gain in defining standard procedures for defining the transfer functions controlling the material properties of the interaction.

The technology is now also mature enough for the quantitative evaluation needed to determine the magnitude of the clear potential of haptics. Being a mature technology, however, does not mean that there is no room for improvement. The linear primitives provide a simple zero order positional integration over shapes in the data. This has shown to be sufficient for many cases, but the introduction of non-linear primitives as an option for higher order integration would offer higher precision to systems that need it. The solvers for the haptic primitives can also be refined, and new solvers implemented. These can be specialized to certain configurations or simply use an alternative approach, for example the Nelder-Mead simplex method.

The haptic primitives and the other presented algorithms have shown great potential in the example applications. There are, however, other possible applications for the technology, such as surgery simulation and guidance in specific tasks. For certain application areas, the haptic primitives may also have to be extended into 6 DoF feedback or new primitives developed.

Bibliography

- [AH99] Richard J. Adams and Blake Hannaford. Stable haptic interaction with virtual environments. *IEEE Transactions on Robotics and Automation*, 15(3):465–474, 1999.
- [AKH01] Richard J. Adams, Daniel Klowden, and Blake Hannaford. Virtual training for manual assembly task. *Haptics-e, The Electronic Journal of Haptics Research* (www.haptics-e.org), 2(2), October 2001.
- [AR99] Walter Aviles and John Ranta. Haptic interaction with geoscientific data. In *Proceedings of Phantom User Group Workshop '99*, 1999.
- [AS96] R. S. Avila and L. M. Sobierajski. A haptic interaction method for volume visualization. In *Proceedings of IEEE Visualization*, pages 197–204, October 1996.
- [BG00] Dirk Bartz and Özlem Gürvit. Haptic navigation in volumetric datasets. In *Proceedings of PHANToM User Research Symposium*, 2000.
- [BSH⁺06] Federico Barbagli, Ken Salisbury, Cristy Ho, Charles Spence, and Hong Z. Tan. Haptic discrimination of force direction and the influence of visual information. *ACM Transactions on Applied Perception*, 3(2):125–135, 2006.
- [CHS00] Kwong-Wai Chen, Pheng-Ann Heng, and Hanqiu Sun. Direct haptic rendering of isosurface by intermediate representation. In *Proceedings of ACM Symposium on Virtual Reality Software and Technology VRST '00*, pages 188–194, 2000.
- [CSB94] J. Edward Colgate, Michael C. Stanley, and J. Michael Brown. Issues in the haptic display of tool use. In *Proceedings of the American Control Conference*, pages 3236–3240, 1994.
- [DH00] Bruce Randall Donald and Frederick Henle. Using haptics vector fields for animation motion control. In *Proceedings of IEEE International Conference on Robotics and Automation*, 2000.
- [dLLFR02] R. Q. Van der Linde, P. Lammertse, E. Frederiksen, and B. Ruiters. The hapticmaster, a new high-performance haptic interface. In *Proceedings of Eurohaptics*, 2002.
- [For] Force dimension. <http://www.forcedimension.com>.

- [GSM⁺97] Sarah F. Frisken Gibson, Joe Samosky, Andrew Mor, Christina Fyock, W. Eric L. Grimson, Takeo Kanade, Ron Kikinis, Hugh C. Lauer, Neil McKenzie, Shin Nakajima, TakaHide Ohkami, Randy Osborne, and Akira Sawada. Simulating arthroscopic knee surgery using volumetric object representations, real-time volume rendering and haptic feedback. In *CVRMed*, pages 369–378, 1997.
- [HB06] Aren Hunter and Lisa Best. Haptic memory and identification of curvature in pie charts. In *Proceedings of EuroHaptics*, pages 479–481, July 2006.
- [HI97] W. Hashimoto and H. Iwata. A versatile software platform for visual/haptic environment. In *Proceedings of ICAT'97*, pages 106–114, 1997.
- [HM02] William S. Harwin and Nicholas Melder. Improved haptic rendering for multi-finger manipulation using friction cone based god-objects. In *Proceedings of Eurohaptics*, 2002.
- [HQB98] Chewei Huang, Huamin Qu, and Arie Kaufman. Volume rendering with haptic interaction. In *Proceedings of the Third PHANTOM Users Group Workshop*, 1998.
- [HTB⁺06] Cristy Ho, Hong Z. Tan, Federico Barbagli, Ken Salisbury, and Charles Spence. Isotropy and visual modulation of haptic force direction discrimination on the human finger. In *Proceedings of Eurohaptics*, 2006.
- [Hut00] Matthew Hutchins. A constraint equation algebra as a basis for haptic rendering. In *Proceedings of Phantom User Group Workshop'00*, 2000.
- [IBHJ03] Milan Ikits, J. Dean Brederson, Charles D. Hansen, and Christopher R. Johnson. A constraint-based technique for haptic volume exploration. In *Proceedings of IEEE Visualization '03*, pp. 263–269, 2003.
- [IN93] H. Iwata and H. Noma. Volume haptization. In *Proceedings of IEEE 1993 Symposium on Research Frontiers in Virtual Reality*, pages 16–23, October 1993.
- [ISNC03] Luis Ibanez, Will Schroeder, Lydia Ng, and Josh Cates. *The ITK Software Guide: The Insight Segmentation and Registration Toolkit*. Kitware Inc, 2003.
- [ITR06] Ali Israr, Hong Z. Tan, and Charlotte M. Reed. Frequency and amplitude discrimination along the kinesthetic-cutaneous continuum in the presence of masking stimuli. *Journal of the Acoustical Society of America*, 120(5), 2006.
- [KAC88] M. J. Kiphart, B. C. Auday, and H. A. Cross. Short-term haptic memory for three-dimensional objects. *Perceptual and motor skills*, 66(1), 1988.
- [KD02] Arthur E. Kirkpatrick and Sarah A. Douglas. Application-based evaluation of haptic interfaces. In *Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, 2002.

- [KKSD02] Laehyun Kim, Anna Kyrikou, Gaurav S. Sukhatme, and Mathieu Desbrun. An implicit-based haptic rendering technique. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.
- [KSW⁺99] O. Körner, M. Schill, C. Wagner, H.-J. Bender, and R. Männer. Haptic volume rendering with an intermediate local representation. In *Proceedings of the 1st International Workshop on the Haptic Devices in Medical Applications*, pages 79–84, June 1999.
- [LC87] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In Maureen C. Stone, editor, *Proceedings of Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 163–169, 1987.
- [LK87] Susan J. Lederman and Roberta L. Klatzky. Hand movements: A window into haptic object recognition. *Cognitive Psychology*, 19(3):342–368, July 1987.
- [LLPN00] Dale A. Lawrence, Christopher D. Lee, Lucy Y. Pao, and Roman Y. Novoselov. Shock and vortex visualization using a combined visual/haptic interface. In *Proceedings of IEEE Conference on Visualization and Computer Graphics*, 2000.
- [LLY05] Claes Lundström, Patric Ljung, and Anders Ynnerman. Extending and simplifying transfer function design in medical volume rendering using local histograms. In *Proceedings of Eurographics/IEEE VGTC Symposium on Visualization*, 2005.
- [LT69] Susan J. Lederman and M. M. Taylor. Perception of interpolated position and orientation by vision and active touch. *Perception & Psychophysics*, 6(3), 1969.
- [MGS96] A. Mor, S. Gibson, and J. Samosky. Interacting with 3-dimensional medical data: Haptic feedback for surgical simulation. In *Proceedings of Phantom User Group Workshop'96*, 1996.
- [MS94] Thomas H. Massie and J. K. Salisbury. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 1994.
- [PL98] Lucy Pao and Dale Lawrence. Synergistic visual/haptic computer interfaces. In *Proceedings of Japan/USA/Vietnam Workshop on Research and Education in Systems, Computation, and Control Engineering*, 1998.
- [PNCD01] P. J. Passmore, C. F. Nielsen, W. J. Cosh, and A. Darzi. Effects of viewing and orientation on path following in a medical teleoperation environment. In *Proceedings of IEEE Virtual Reality 2001*, 2001.
- [RB02] Beate Riedel and A. Mike Burton. Perception of gradient in haptic graphs by sighted and visually impaired subjects. In *Proceedings of Eurohaptics*, pages 134–137. University of Edinburgh, United Kingdom, 2002.

- [RKK97] Diego C. Ruspini, Krasimir Kolarov, and Oussama Khatib. The haptic display of complex graphical environments. *Computer Graphics*, 31(Annual Conference Series):345–352, 1997.
- [SBM⁺95] K. Salisbury, D. Brock, T. Massie, N. Swarup, and C. Zilles. Haptic rendering: programming touch interaction with virtual objects. In *Proceedings of the 1995 Symposium on Interactive 3D Graphics*, 1995.
- [ST97] Kenneth Salisbury and Christopher Tarr. Haptic rendering of surfaces defined by implicit functions. In *Proceeding of the ASME 6th Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 1997.
- [TBS⁺06a] Hong Z. Tan, Federico Barbagli, Ken Salisbury, Cristy Ho, and Charles Spence. Force-direction discrimination is not influenced by reference force direction. *Haptics-e*, 4(1), 2006.
- [TBS⁺06b] Hong Z. Tan, Federico Barbagli, Ken Salisbury, Cristy Ho, and Charles Spence. Force-direction discrimination is not influenced by reference force direction. *Haptics-e: The Electronic Journal of Haptics Research*, 4(1), 2006.
- [Thi98] Jean-Philippe Thirion. Image matching as a diffusion process: an analogy with Maxwell’s demons. *Medical Image Analysis*, 2(3):243–260, 1998.
- [TJC97] Thomas V. Thompson II, David E. Johnson, and Elaine Cohen. Direct haptic rendering of sculptured models. In *Proceedings of Symposium on Interactive 3D Graphics*, Providence, RI, 1997.
- [WH00] Steven Wall and William Harwin. Quantification of the effects of haptic feedback during a motor skills task in a simulated environment. In *Proceedings of Phantom User Research Symposium’00*, 2000.
- [WPS⁺02] Steven A. Wall, Karin Paynter, Ann Marie Shillito, Mark Wright, and Silvia Scali. The effect of haptic feedback and stereo graphics in a 3D target acquisition task. In *Proceedings of Eurohaptics*. University of Edinburgh, United Kingdom, 2002.
- [YCK92] Roni Yagel, Daniel Cohen, and Arie Kaufman. Normal estimation in 3D discrete space. *The Visual Computer*, 8:278–291, 1992.
- [ZS95] C. B. Zilles and J. K. Salisbury. A constraint-based god-object method for haptic display. In *Proceedings of IEE/RSJ International Conference on Intelligent Robots and Systems, Human Robot Interaction, and Cooperative Robots*, volume 3, pages 146–151, 1995.