Tangible images

Shahzad Rasool

2014

Shahzad Rasool. (2014). Tangible images. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/61841>

<https://doi.org/10.32657/10356/61841>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

TANGIBLE IMAGES

TANGIBLE IMAGES

SHAHZAD RASOOL

SHAHZAD RASOOL

SCHOOL OF COMPUTER ENGINEERING

2014

2014

SHAHZAD RASOOL

TANGIBLE IMAGES

SHAHZAD RASOOL

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2014

Acknowledgements

First of all, I would like to thank my supervisor Associate Professor Alexei Sourin for his invaluable instructions. I am grateful for his help throughout my graduate study at Nanyang Technological University. He helped me understand basic ideas, and to correct theoretical findings. Without his suggestions, ideas and comments, the project could not have reached this point.

I would also like to thank School of Computer Engineering, Nanyang Technological University for providing me the opportunity and financial support to further my study and research. Special thanks go to Fraunhofer IDM@NTU for providing me the first class equipment and technical support.

Lastly, I would like to thank my friend, Dr. Wei Lei, for his precious support and Dr. Fareed H. Y. Kagda for his valuable input and cooperation. Furthermore, I wish to thank my parents. Their selfless concern and encouragement has been the best support during my postgraduate study.

Abstract

Haptic interaction is typically based on detecting collisions of the haptic interaction point with physical models of the virtual objects comprising polygon meshes, point sets, or procedural models topologically collocated with the geometric models of the object in the modeling space. However, it is not always possible or feasible to make such models when it comes to using real images or videos as elements of interaction. Moreover, traditional polygon-based modelling representation yields large sized files and time consuming visual rendering to achieve photorealistic quality. This deficiency could be overcome by employing image-based representations. A survey of the haptic interaction methods related to such representations leads to the conclusion that there is room for research for proposing new algorithms of haptic interaction with two-dimensional images and videos, without reconstruction of 3D models.

Based on the paradigm ‘what we see is what we touch’, a set of novel techniques for making two dimensional images tangible is proposed. A novel depth-based haptic geometry rendering algorithm is devised which is combined with haptic rendering of surface texture and physical properties. The computational complexity of the proposed rendering algorithm is much lower than that of visual and haptic rendering of three dimensional scene representations while photorealistic visual quality is achieved.

An innovative extension of the algorithm to reliably interact with hybrid scenes composed of 3D models and 2D images is proposed. Visible 3D models are used to simulate editable parts of scene while low resolution invisible 3D models are used as haptic containers to augment the visual interaction experience. These haptic models have simplified geometry and are additionally capable of representing non-visual force effects

such as wind, flow, magnetism, etc.

In order to validate the research niche, a comprehensive application example of simulation of a minimally invasive surgical procedure is implemented. The developed image-driven virtual arthroscopy training simulator provides a set of training exercises for learning basic skills required for such minimally invasive surgery. A few other applications are implemented to illustrate the usefulness and feasibility of the proposed algorithms.

Contents

Acknowledgements	1
Abstract	3
Contents.....	5
List of Figures	9
List of Abbreviations.....	17
Chapter 1 Introduction	19
Chapter 2 Survey of Works Related to Image-Driven Haptic Interaction	24
2.1 The haptic interface	25
2.2 Haptic interaction in 3D virtual environments	31
2.2.1 Collision detection	31
2.2.2 Haptic rendering	33
2.3 Haptic interaction with 2D images	37
2.3.1 Direct computation of forces from 2D images.....	38
2.3.2 Force computation using additional information	45
2.4 Critical analysis	53
2.5 Summary	54
Chapter 3 Tangible Images – An Approach to Image-Driven Haptic Interaction	56
3.1 Hypothesis	56
3.2 Approach, research and development novelty	58
3.3 Summary	64

Chapter 4	A Novel Algorithm for Depth-Based Haptic Rendering.....	65
4.1	Depth-based haptic geometry rendering	65
4.1.1	Smooth transitions at object boundaries	70
4.1.2	Rendering haptic textures	75
4.1.3	Rendering material properties of objects	76
4.1.4	Interactive scene modifications.....	77
4.2	Special modes of the core algorithm	79
4.2.1	Handling non-geometric intensity discontinuities.....	80
4.2.2	Emphasizing contours	83
4.2.3	Texture emphasis	87
4.3	Analysis and discussion.....	88
4.4	Summary	90
Chapter 5	Hybrid Approach – Augmenting 2D Images with 3D Object Models.....	92
5.1	2D images and augmented editable 3D models.....	93
5.1.1	3D haptic rendering with depth-based interaction	93
5.1.2	Depth-based interaction with the hybrid scene	95
5.2	Invisible haptic containers	99
5.3	Summary	103
Chapter 6	Constraint-Based Interaction with 2D Images	104
6.1	Multiple contact point interaction.....	104
6.2	Dynamic haptic workspace mapping	109
6.3	Alteration of DOF using physical constraints	115

6.4	Summary	117
Chapter 7	Application Examples.....	119
7.1	Arthroscopy simulation	119
7.1.1	Related work	121
7.1.2	Image-based approach.....	125
7.1.3	Training exercises	136
7.2	Other image-driven medical training simulations	141
7.2.1	Simulation of acupuncture therapy	141
7.2.2	Simulation of palpation	144
7.2.3	Simulation of venipuncture.....	144
7.3	Touch desktop application.....	145
7.4	Other applications	149
7.5	Summary	149
Chapter 8	Conclusion and Future Work.....	151
8.1	Conclusion.....	151
8.2	Future work	154
List of Publications		157
Journal papers.....		157
Conference papers		157
References		159
APPENDIX A: URLs of Supplementary Videos.....		177

APPENDIX B: Point-Plane Distance180

List of Figures

Figure 1.1	Examples of force feedback desktop haptic devices, (a) Geomagic Touch, (b) Novint Falcon, (c) Geomagic Touch X and (d) Geomagic Phantom Premium.....	27
Figure 4.1	(a) A 2D image with brightness reflecting depth of the scene (b) corresponding heightmap (c) surface felt by the haptic device (d) content of the mask at the position indicated in (a).....	67
Figure 4.2	(a) Cross product of two vectors sharing the current HIP gives direction of surface normal, (b) Calculation of surface normal by averaging a number of calculated surface normals.....	68
Figure 4.3	Surface normal adjustment at depth discontinuity (a) surface normal calculated without using previous normal (b) surface normal after the smoothing adjustment (c) angles between previous and current normals with and without using history information.....	71
Figure 4.4	Depth-based haptic geometry rendering workflow.....	72
Figure 4.5	Depth-based haptic rendering algorithm applied to depth stream obtained by Microsoft Kinect. Arrows show the force direction at different parts of the scene.	74
Figure 4.6	Haptic rendering of moving objects with depth stream obtained using Microsoft Kinect.	74
Figure 4.7	Haptic rendering of physical properties by assigning different properties to different regions, (a) an outdoor image, (b)	

	manually created segmentation map.	77
Figure 4.8	(a) An image superimposed by a segmented image patch, (b) animation produced in the segmented patch by interaction using a haptic device.	78
Figure 4.9	Deformation through (a) image warping in a virtual palpation system, (b) superimposition of the image with small texture patches.	79
Figure 4.10	A facial image with areas that will be interpreted as deep cavities	80
Figure 4.11	Handling non-geometric intensity discontinuities (a) test image in normal (default) mode, (b) test image in the offset mode.	81
Figure 4.12	Offset mode using a binary map. (a) 2D input image with blurry transitions, (b) the corresponding binary map with sharp discontinuities, (c) feedback forces in the default mode are directed as if the IKEA letters are carved deep inside the surface, (d) feedback forces in the offset mode are orthogonal to the surface	83
Figure 4.13	An edge with width Δ , an associated normal n and a function defined edge profile.	84
Figure 4.14	Movement of an edge (a) the edge is detected at $t=0$, (b) the edge moves from left to right, (c) the edge moves from right to left.	86
Figure 4.15	Texture emphasis mode: (a) test image (b) the histogram	

	generated by the developed application (c) the perceived surface before and after texture emphasis.....	88
Figure 4.16	Flow chart showing relationship of different parts of the tangible images approach.....	89
Figure 5.1	Interaction with soft tissues in arthroscopic surgery (a,b) probing of meniscus, (c) removal of damaged part of meniscus	94
Figure 5.2	(a) 3D polygonal model of meniscus augmented to an arthroscopic image (b) a grasper tool operating on the 3D meniscus.....	95
Figure 5.3	Mask filling algorithm for hybrid scene composed of 3D model and 2D image.	96
Figure 5.4	Meniscus modeling using implicit functions. (a) Summation of inverse square functions, (b) Relation of the modeled part of the meniscus to the complete meniscus along with different views of the modeled part. (c) Meniscus removal using set theoretic operations.....	98
Figure 5.5	Illustration of using invisible haptic containers (a) a 3D sphere, (b) an image of the sphere on a plane and a surrounding invisible 3D sphere, (c) a front view on the image and the matching geometric sphere, (d) Superposition of the simplified 3D geometry with an actual image.	100
Figure 5.6	A photo of Stanford Bunny and its 3D VRML model used to augment the image.	102

Figure 5.7	(a) A snapshot from a whirlpool video which is superimposed by the container in (b) at appropriate orientation, and the clockwise force effect on the haptic device, (b) a helical force field inside a transparent cylindrical container.	103
Figure 6.1	A cubic haptic cursor with eight haptic contact points (HCPs) interacting with two planes. The feedback force is applied at the HIP which lies at the center of the cubic avatar. (a) A single HCP is in contact with one of the planes. (b) Two HCPs in contact with one plane. (c) Two HCPs in collision with different planes.	108
Figure 6.2	2D representation of the cubic haptic cursor with HCPs at its corners interacting with two closely placed planes.....	108
Figure 6.3	Tool-scene penetration. A comparison between (a) single point interaction and (b) multi-point interaction.	109
Figure 6.4	Dynamic haptic workspace mapping. Top view of a virtual environment is shown with depth values increasing from bottom to top. Near and far planes of the scene are marked.	112
Figure 6.5	Haptic workspace is mapped to different regions of interest to allow for exploration of fine details.	112
Figure 6.6	Dynamic mapping of haptic workspace onto a 16-bit grayscale image.	114
Figure 6.7	Application of physical constraints on the device for simulation of minimally invasive surgery. (a-b) creation of device handle	

	extension using medical instruments, (c-d) simulation of OR environment using the extended device.	116
Figure 6.8	Change in effective motion range when the device stylus is moved (a) inwards, (b) outwards.	117
Figure 7.1	Arthroscope motions and rotations changing the camera view.	120
Figure 7.2	Manual stitching of the individual frames into a panoramic image of the knee meniscus. Gaps are left for illustration purposes.	126
Figure 7.3	Panoramic image created by stitching images obtained from the actual surgical video and the hybrid simulated view.	127
Figure 7.4	Preprocessing of arthroscopic images for depth estimation. (a) RGB image, (b) partially filtered GB channels, (c) grayscale image without GB filtering, and (d) grayscale image with GB filtering.	128
Figure 7.5	Haptic rendering of different physical properties in arthroscopy simulation, (a) actual image with arrows showing direction of rendered force, (b) segmentation map.	129
Figure 7.6	Geomagic Touch haptic devices used for simulation of minimally invasive surgery.	130
Figure 7.7	Location of HIP, proxy point and incision point and the relative distances between them with reference to the modified haptic device.	131
Figure 7.8	Parameters defining the virtual arthroscopic camera in the	

	virtual environment. (a) at initialization, (b) at subsequent frames.....	132
Figure 7.9	Positioning of the textured hemisphere (a) at initialization, (b) at subsequent frames. Here, 0 is the origin and the dotted hemisphere in (b) shows the previous hemisphere location.	135
Figure 7.10	(a) Hemisphere with panoramic image as a texture mapped on its inside surface, (b) Views generated by the camera looking at various parts of the hemisphere.....	135
Figure 7.11	Triangulation: Collision of instrument with the virtual arthroscope.....	137
Figure 7.12	Opening and closing of grasper jaws using multiple contact interaction (a) location of HCPs with respect to the instrument (b) forces calculated with 2-point instrument.	139
Figure 7.13	Six point pyramid used for grasping.	140
Figure 7.14	Simulation of skin deformation by superimposition of small texture patch onto the background image at the point of interaction.....	142
Figure 7.15	Screenshots of acupuncture simulation (a) insertion of needle at incorrect point (b) insertion of needle at correct point.....	143
Figure 7.16	Screenshot of a simulation of acupuncture therapy by super imposing a 3D haptic container onto a 2D image.....	143
Figure 7.17	A virtual palm palpating a malignant and enlarged liver. Notice the deformation near the belly button in the left image.	144

Figure 7.18	Image-driven venipuncture simulation (a) image and the function-defined 3D model of the veins, (b) screenshot of the implemented system.....	145
Figure 7.19	Snapshots of the touch screen application.	146
Figure 7.20	Use of desktop tangible image application for haptic interaction with graphical user interfaces in contour perception mode. The direction of the force at the current location of the HIP is shown as an arrow.	147
Figure 7.21	Haptic video communication: Proposed algorithm applied to video communication on the web. The direction of the haptic force at the current location of the HIP (at the nose) is shown as a red arrow.	147
Figure 7.22	Offset mode applied to facial images. (a) 2D input image with blurry transitions, (b) the corresponding binary map with sharp discontinuities, (c) force feedback in the default mode, (d) force feedback in the offset mode.	148
Figure 7.23	Interaction with the moving object in haptic video communication. The handle of the haptic device moves up and down following the motion of the hand in the Skype window.	149

List of Abbreviations

2D	Two dimensional
3D	Three dimensional
ACL	Anterior Cruciate Ligament
API	Application Programming Interface
CBIR	Content-Based Image Retrieval
CT	Computed Tomography
CUDA	Compute Unified Device Architecture
DOF	Degree of Freedom
FLANN	Fast Library for Approximate Nearest Neighbors
GAMMA	Geometric Algorithms for Modeling, Motion and Animation
GIF	Graphics Interchange Format
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HCP	Haptic Contact Point
HDAL	Haptic Device Abstraction Layer
HDAPI	Haptic Device API
HIP	Haptic Interaction Point / Haptic Interface Point
HLAPI	Haptic Library API
LSB	Least Significant Bit
MDA	Media Development Authority
MOE	Ministry Of Education

MPEG	Moving Picture Experts Group
MRI	Magnetic Resonance Imaging
NRF	National Research Foundation
NURBS	Non-Uniform Rational B-Spline
OB	Oriented Bounding Box
ODE	Open Dynamics Engine
OEM	Original Equipment Manufacturer
OpenGL	Open Graphics Library
OpenNI	Open Natural Interaction
PDE	Partial Differential Equation
R&D	Research and Development
RGB	Red Green Blue
RGBD	Red Green Blue Depth
SDK	Software Development Kit
SFS	Shape From Shading
SIRS	Siena Robotics and Systems
SSR	Single Scale Retinex
SURF	Speeded-Up Robust Features
VRML	Virtual Reality Modeling Language

Chapter 1 Introduction

Haptics refers to acquisition of information and manipulation of objects through the sense of touch. Haptic technology is a sensory feedback technology for simulating tactile interaction based on applying forces, vibrations and motions to the user by means of specialized hardware. Such mechanical stimulation of touch receptors is used to assist in the creation of virtual objects and interaction with them. Compared to visual rendering, which simulates a rather passive visual information collection process and requires 30-60 Hz refresh rate, haptic interaction is a bidirectional information exchange process, which requires much higher refresh rates of ~1000 Hz. We gather information by manipulating objects as well as exerting forces on them. Moreover, we either receive a force feedback from them or perceive properties of the objects due to stimulation of receptors in the skin upon contact. Based on neural inputs generated upon contact with objects, the human haptic system can be classified into two types of haptic feedback namely cutaneous and kinesthetic feedback systems (Klatzky & Lederman, 2003). The stimulation of neurons caused by receptors in the skin produces cutaneous feedback, while neural stimulation due to receptors located in muscles, tendons and joints results in kinesthetic feedback. Kinesthetic feedback allows humans to perceive the spatial configuration of the hand and fingers while cutaneous feedback (also known as tactile feedback) results in perception of contact conditions between the skin and objects in the real world. In this research we deal with kinesthetic feedback also known as force feedback.

Typically, haptic interaction is based on collision detection of the haptic interaction point with 3D virtual objects defined by polygon meshes, point sets, or procedural

models. However, it is not always possible or feasible to create such models for complex scenes, especially when photorealistic quality is desired.

As an alternative approach, we propose *tangible images* which are a set of algorithms and methods for image-driven haptic rendering where a displayed image, real or simulated, is used as a source for the force feedback calculations at any given pixel of the image. The rationale for this approach lies in image-driven visualization which is used as an alternative to the conventional method of 3D modeling followed by visual rendering onto a 2D display. The examples are various interactive panoramic images of 3D environments, image-based virtual walking in interactive street views, and zooming views at compressed images. Hence, instead of creating haptic models and displaying 3D models of the objects as 2D images on the screen, we make maximum use of the final 2D computer generated images or actual photographs or video to understand the three-dimensionality of the objects and to calculate the feedback forces as if the haptic device interacts with the 3D scene displayed in the image.

To confirm the research direction, we surveyed the relevant papers and commercial products and concluded that there are two major groups of relevant works: those which deal with images to obtain haptic forces from them directly, and the methods which use additional supplementary information besides the 2D image to perform haptic rendering. We concluded that the existing approaches for converting information extracted from images directly into haptic forces fail to provide haptic perception of the overall visible 3D geometry of objects in the scene but rather produce touch sensation of small perturbation of visible edges and surface textures. On the other hand, methods appending additional information to images require significant authoring and editing efforts.

Sometimes they also require the use of sensors at the time of capture to obtain physical property data related to the scene. Moreover, 3D reconstruction followed by haptic rendering requires significant computational resources and fails to provide high precision as well as overall consistency of the reconstructed scenes. Therefore, we have concluded that there is a room for research in direct haptic interaction with images without full 3D reconstruction. *We then hypothesize that images of real scenes can be turned into tangible images and felt with haptic devices as if they were real 3D scenes.* Feedback forces can be mostly derived for visible surfaces of the objects straight from the images with pixel precision. Feedback force components are also obtained by segmenting the image and assigning physical properties to their parts, and by augmenting the images with invisible haptic objects. The three degrees of freedom force feedback allows the user to haptically explore the visible object's geometry, texture and surface properties, such as friction and elasticity, as well as inner properties such as viscosity. Interactive modifications to the scenes are modeled either by image warping or by augmenting 2D images with editable parts. The resulting haptic rendering of visible surfaces provides 2.5D interaction which we call “**What You See Is What You Touch**”.

The **research novelty** of the proposed project is in the way in which the haptic forces are calculated. In contrast to existing approaches, to derive forces from images we use a combined technique where not only fine details but visible three-dimensionality of the objects is perceived through the sense of touch. This is done in run-time without performing 3D reconstruction of the scene. Research novelty also lies in the haptic rendering of hybrid scenes comprising 2D images and augmented 3D objects. We use mathematical functions and procedures to define geometry, elasticity, friction and

viscosity of these editable or invisible 3D objects so that the user feels believable force feedback when touching or penetrating the objects with the haptic device. Being small in size and providing any desired level of detail, these additional models can be efficiently used when such tangible images are exchanged across the network. We also propose novel haptic collision detection algorithms that permit us to use common desktop haptic devices with one haptic interaction point for solving haptic rendering problems that require multiple haptic interaction points.

The **development novelty** lies in a few pilot applications developed to illustrate the usefulness of the proposed research¹. Firstly, a comprehensive implementation of an arthroscopy training simulator highlights various algorithms proposed in this thesis. The three DOF desktop haptic devices are modified by creating physical constraints to the device to match the real surgical tool movement. The resulting increase in number of degrees of freedom of the device is used for modeling arthroscopic cameras and surgical tools. The modeled oblique-viewing camera is suitable for acquisition of hand-eye coordination skills necessary for minimally invasive joint surgery. Panoramic images constructed from real arthroscopic surgery videos are displayed on the inside of a moving hemisphere and different views seen through the oblique-viewing camera are generated. Secondly, a desktop interaction software tool makes the entire Microsoft Windows desktop available for touch interaction by mapping the haptic workspace onto the entire or a part of the screen. It allows for haptic perception of graphical user interfaces through emphasis of contours. We also demonstrate the applicability of the proposed approach to

¹ This research is part of the externally funded project “Collaborative Haptic Modeling for Orthopaedic Surgery Training in Cyberspace” (MOE Tier 2 grant)

dynamic video sequences by presenting a haptic enabled video communication tool which makes a Skype video tangible such that the involved parties are able to not only see each other but also touch the video stream.

The thesis is organized as follows. In Chapter 2, we survey the existing works on image-driven haptic interaction and highlight the room for research. Chapter 3 provides an analysis on the problem and outlines the proposed approach. A novel depth-based haptic rendering algorithm is presented in Chapter 4 while interaction with hybrid scenes comprised of 2D images and 3D models is presented in Chapter 5. In Chapter 6, we present image-driven interaction in the presence of virtual or physical constraints. A number of application examples are presented in Chapter 7 followed by concluding remarks in Chapter 8.

Chapter 2 Survey of Works Related to Image-Driven Haptic Interaction

Haptic interaction with virtual environments enhances the sense of immersion experienced by a user as it augments the visual interaction experience by communicating more information with the scene. The procedure starting at user manipulation to the final feedback sensation delivered to the user is called *haptic interaction*. Haptic interaction has found application in a number of diverse application areas including digital painting (Gary, 2004; Johnson et al., 1999), sculpting (Jing & Hong, 2002; Zhu & Lee, 2004), medical training (Coles et al., 2011b), data visualization (Paneels & Roberts, 2010), rehabilitation (Reinkensmeyer et al., 2001), virtual mock-up (Aubry et al., 2007; Thouvenin, 2005), virtual assembly (Chen et al., 2004; Iglesias et al., 2007), design review (Huang et al., 2001; Sidharta et al., 2006; Xiangyu, 2006), etc. It includes collision detection, haptic rendering and application of forces to the user through haptic devices. Collision detection is the determination of all contact points between the probe and scene elements along with the time of each contact while haptic rendering is defined as the process of computing and generating forces in response to user interactions with virtual objects (Salisbury et al., 1995). Haptic rendering algorithms usually return a force vector and a torque vector which are applied to the force-reflecting device. Before we move on to surveying the works related to image-driven haptic rendering, we will briefly review some basic principles of haptic interaction in 3D virtual environments along with the hardware and software available for such interaction.

2.1 The haptic interface

In order to display a 3D scene, the visual properties of the scene are sampled and reproduced with a graphic display. Similarly, tangible physical features of objects in the scene can be sampled and rendered with haptic devices. Typically, haptic devices extend the user's hand or fingers to monitor their movements and mimic them accordingly. The type of feedback used to convey the sense of touch is determined by the type of haptic device (cutaneous or kinaesthetic) being used. Since cutaneous and kinesthetic senses stimulate different sensors, therefore devices for providing these two types of feedback have different requirements. Kinesthetic (or force feedback or proprioceptive) devices are electromechanical devices that deliver directional forces based on solid boundaries, weight, mechanical compliance and inertia. These devices deliver forces to the user to trigger limb motion. Examples of kinesthetic devices include desktop haptic devices and exoskeleton based devices among others. Cutaneous or tactile feedback devices aim to provide spatially distributed sensations at the skin based on pressure, texture, thermal properties, wetness etc. These devices simulate skin-to-object contact. They include Braille printer, vibro-tactile devices, surface displays, pin-array type devices, etc. (Hayward & Maclean, 2007). There are a number of device manufacturers producing different haptic devices and Software Development Kits (SDKs). Here we introduce the most common kinaesthetic feedback haptic devices.

Invented by SensAble Technologies Inc. and acquired by Geomagic in April 2012, the Phantom product line (Geomagic, 2013) has various models that meet varying needs of both research and commercial customers. The product line ranges from high-precision high-DOF Phantom Premium devices to the low-end but more affordable Geomagic®

Touch™ (formerly Phantom Omni) devices along with OpenHaptics toolkit, the second generation of SDK with a comprehensive architecture composed of two layers having different capabilities.

Immersion Corporation (Immersion-Corporation, 2014) provides both tactile and force feedback devices in addition to a development platform for various applications. Its TouchSense series of tactile feedback devices are usable in a wide range of applications such as mobile devices and industrial applications. Immersions MOTIV™ is a platform for development of android applications. It simplifies haptic integration for original equipment manufacturers (OEMs) and application developers through MOTIV Integrator and MOTIV SDK, respectively. It allows access to hundreds of pre-designed haptic effects, code samples and resources.

A Switzerland based company named Force Dimension (Force-Dimension, 2013), provides a range of different haptic devices and software. Their low-end devices are called *Delta* devices while *Omega* series devices are optimized for high-end force feedback and are able to provide different levels of performance and modularity. The most advanced device from Force Dimension is the *Sigma* device which produces 7 active DOF. *Omega.medical* is a commercially available specialized haptic interface for medical robotics applications. Force Dimension SDK is a software interface for these products that allows combining haptics and robotics capabilities into applications, however, Force Dimension also supports the open source software library for computer haptics, visualization and real-time simulation from Stanford university known as CHAI 3D (Conti et al., 2005).

Novint Technologies Inc. (Novint-Technologies-Inc., 2013) produces gaming

devices with force feedback capabilities. Their most popular device is Novint Falcon which is a low cost and high fidelity haptic device. Although designed for gaming purposes, it is one of the cheapest devices on the market, being used for research purposes. Novint Technologies also offers a free SDK to support various haptic applications.

Figure 1.1 Examples of force feedback desktop haptic devices, (a) Geomagic Touch, (b) Novint Falcon, (c) Geomagic Touch X and (d) Geomagic Phantom Premium.

CyberGlove Systems LLC (CyberGlove, 2013) markets data glove hardware solutions along with associated development software such as VirtualHand SDK. Different haptic data gloves are produced capable of delivering haptic feedback to the users' finger, hand or arm. CyberGrasp line of gloves produces kinaesthetic as well as tactile feedback. CyberForce is their highest-end product, which adds whole-arm force-feedback as it allows for experiencing the complete sensation of object

manipulation. It is attached to a desktop platform and is developed under license from Sensable Technologies Inc. There are also a few other haptic device manufacturers including Quanser company (Quanser-Inc., 2014) and MPB Technologies (MPB-Technologies-Inc., 2013) producing haptic feedback devices.

MOOG Inc. produces a haptic rehabilitation unit using a 3DOF force-controlled haptic device interface (van der Linde, 2003), known as the HapticMaster, which provides high stiffness and high force sensitivity. The device comes with a simple grip handle but also be changed to grip which provides additional 3DOF for orientation measurement. Customized handles can also be fitted to the HapticMaster robot arm. It has been used for rehabilitation and assembly training applications.

Most haptic device manufacturers provide well-documented and optimized device drivers and programming interfaces. Low-level tasks of communication with the haptic device, such as tracking user motion and delivering haptic output to the user, are handled by device driver while SDKs provide a programming interface by hiding unnecessary device interface details and simplifying application development. Open source and cross-platform drivers and interface libraries that support different operating systems have also been developed by various educational and research institutions. A few popular SDKs are introduced here.

CHAI3D is an open source SDK designed by Stanford University for combining 3D modelling with haptic rendering capabilities. Written in C++, it supports different devices from different manufacturers. It is a multi-platform framework which provides visualization through OpenGL, rigid body interaction through Open Dynamics Engine (ODE), and simulation of deformable objects through GEL Dynamics Engine.

Reachin API was a development platform which provided rapid application development as well as low level control for haptic enabled applications (Reachin-Technologies-AB, 2013). The hardware independent API was developed by a Swedish company named Reachin Technologies AB which was dissolved in October 2013. It provided implementation freedom by offering support for various programming languages including C++, Python and VRML. It included ForceModel™ library which allowed for easy addition of haptic effects to developed applications. It also included provision for multiple users to interact over a network. Now defunct, Reachin Technologies also offered a haptic engine for easy and fast development called HaptX. It was originally developed for game engines and was capable of interacting with different physics and graphics engines.

Haptik Library (De Pascale & Prattichizzo, 2007) developed by SIRS lab, Italy is a component based library that provides a hardware abstraction layer for accessing haptic devices. It supports both OpenGL and DirectX and is designed to be user friendly but is not tied to any particular physics engine. It is written in C++ but can also be used with Matlab, Simulink and Java applets.

H3D API (SceneGraphics-AB, 2013) developed by SenseGraphics is a high-level scene graph programming interface. Haptic operations are performed by the lower layer called HAPI while OpenGL is employed for visual rendering of the scene represented using the X3D file format. It is written in C++ and available both as open source as well as in commercial licenses. The cross platform API also provides interface to Python scripts and is a good tool for fast prototyping of haptic applications.

Open Haptics Toolkit (Itkowitz et al., 2005) is a commercial API that supports

devices produced by Geomagic 3D Systems. It consists of a foundation layer called Haptic Device API (HDAPI) and a high level layer known as Haptic Library API (HLAPI). HDAPI provides more flexibility to the user by allowing for the forces to be sent to the device directly. It also provides control over low-level runtime details through the driver. On the other hand, HLAPI is built on top of HDAPI and is designed for fast incorporation of haptic effects into existing programs allowing significant reuse of code. High level collision detection routines and functions for direct handling of geometric primitives and transformations are bundled into HLAPI but its usage is subject to the developer having knowledge of polygons making up the scene.

HDAL (Haptic Device Abstraction Layer) (Novint-Technologies-Inc., 2013) is a low level commercial closed-source SDK produced by Novint Technologies Inc. It supports only Novint devices working under Microsoft Windows. An open source driver libNiFalcon, supporting Novint devices across other platforms is also available (Machulis, 2009).

Each of these software packages has its own advantages. Low level APIs such as HDAL, libNiFalcon, HDAPI and low level versions of CHAI3D and HAPI hide the kinematics algorithm implementation. The developers can work directly with position, orientation and force vectors in the application. High level APIs such as H3D, HLAPI, etc. allow incorporation of haptics into the applications based on using functions analogous to those provided by graphics libraries. A more detailed comparison between these SDKs along with their shortcomings can be found in (Kadlecek, 2011; Wei, 2011).

2.2 Haptic interaction in 3D virtual environments

The principle behind touch interaction with virtual environments is that the position and orientation of the device end-effector are continuously tracked and their relationship with objects populating the environment determines if a force response needs to be computed. Since time is considered as a discrete entity in digital systems, the position of the device end-effector and the corresponding haptic interface point can only be tracked at discrete intervals. Typically, this has to be performed 1000 times per second for believable haptic interaction experience, however in some cases subjective user perception has also been reported as believable at a frequency as low as 300-600 Hz (Booth et al., 2003; Burdea, 1996). On the other hand, 30 frames per second is a sufficient update frequency for interactive visualization. Therefore, synchronization between the haptic and graphic pipelines must be ensured. If no force is to be rendered, the haptic interface point (virtual representation of the haptic interface) is allowed free movement without resistance. As the user moves the device effector, the application is responsible to track and calculate feedback forces that hinder movement of the haptic interface point into solid objects in the virtual environment.

2.2.1 Collision detection

The virtual representation of the end-effector can be a point, a vector frame of reference, or a 3D object. Its position inside a virtual environment is controlled by the user through the haptic device and its contact with objects in the scene triggers action and reaction forces. Collision detection algorithms check for collisions between the avatar and the objects in the scene at every frame. These algorithms return information about the

location of collision in the scene, time of collision and in some cases information about the extent of collision, i.e. penetration into an object.

Collision detection in haptic interaction is more constrained due to the higher refresh rate required by haptic rendering pipeline. A number of collision detection algorithms for interactive virtual environments and haptic enabled applications have been proposed by the Geometric Algorithms for Modelling, Motion and Animation (GAMMA) group at University of North Carolina Including H-COLLIDE (Gregory et al., 2005), which is a framework that provides a fast collision detection mechanism specifically for haptic interaction. It utilizes spatial decomposition of the 3D space into a grid of cells, OBBTrees and inter-frame coherence to ensure matching high update rate required by haptic rendering pipeline.

Ideally, all objects in the 3D scene should be tested for intersection with each other but the time required for processing complex scenes with thousands of polygons is too high to achieve interactive rendering rates. Typically, algorithms reduce the number of object-object collision tests by checking for collisions of an object only in its vicinity. Further reduction in number of operations is achieved by using hierarchical description of the scene and multi-level collision detection. Ruspini et al. (Ruspini et al., 1997) proposed the use of hierarchical bounding spheres for haptic collision detection. Haptic collision detection can also be accelerated by using physics engines which support fast collision detection. Chan (Chan & Kup-Sze, 2009) has used PhysX engine and Compute Unified Device Architecture (CUDA) to accelerate collision detection for haptic-enabled applications. An intermediate force simulation layer is constructed between the physics engine and the haptic API. Similarly, CHAI 3D supports haptic interaction using ODE

and GEM engines for accelerating haptic collision detection (Conti et al., 2005).

Collisions with polygons rely on knowledge of primitives, i.e. vertices and edges. This information is easily accessible for standalone applications however it becomes infeasible for other applications which may be embedded into existing software tools as plug-ins. An alternative way is to use implicit functions to define objects in a virtual environment. Collision detection becomes trivial in such cases (Salisbury & Tarr, 1997; Sourin & Wei, 2008). Collision detection algorithms have also been proposed for virtual environments composed of curved solid models (Schoemer et al., 2002), high density point clouds (El-Far et al., 2007), subdivision surfaces (Raymaekers et al., 2001), polyhedral models (Mirtich, 1998), super quadric models (Moustakas et al., 2007), and implicit functions (Laehyun et al., 2002).

2.2.2 Haptic rendering

Haptic rendering refers to the calculation of the force response to a detected collision. Rendering algorithms compute the response of the environment based on input from the user. This response is delivered to the user through hardware devices. Generated feedback forces should approximate as closely as possible the forces generated upon contact with real objects.

As the user manipulates the haptic probe, the new state, i.e. position and orientation of the end-effector, are sensed by the device, and collision detection is performed based on virtual scene definition. If a collision is detected, the force response is calculated depending upon the physical properties of the virtual environment, mechanics of interaction and the depth of penetration of the haptic interface point into the object.

Several haptic rendering algorithms have been proposed to render 3D objects. These algorithms can be classified based on how the 3D objects are defined (Salisbury et al., 2004). Surface-based definitions of 3D objects are based on analytical functions or polygonal representation. A number of possible ways of haptic rendering with polygon-based object definitions exist. Ruspini et al. (Ruspini et al., 1997) has broadly classified these into two categories: penalty-based methods and constraint-based methods. Penalty-based methods (Massie & Salisbury, 1994) define a force vector for every point in space by directly mapping the position and orientation of haptic contact point into the haptic scene. At every frame the location of haptic interface point (HIP) is tested for collisions with objects in the scene. When the haptic interface point is not in contact with any object in the scene, i.e. it is in free space, magnitude of the force vector is zero. Otherwise, force magnitude is proportional to the depth of penetration. The penetration depth is calculated as the distance from the HIP to the point on object surface closest to it. These methods only consider the haptic interface point location in the current frame and neglect any history information of the end-effector. Since collisions are checked at discrete time intervals, penetrations through thin objects may go unnoticed. Another problem that occurs with penalty-based methods is when the penetration into an object is too deep. In such cases, the HIP may be pushed out in a different direction than the direction it had penetrated the object from. Constraint-based methods define a god-object or virtual proxy of the haptic interface point. The proxy follows the haptic interface point in free space but rests on the surface of an object if the object is penetrated by the HIP. The feedback force is calculated based on the distance between the proxy and the haptic interface point. Typically, the virtual proxy is point sized but it can also be a sphere or

another shape. The proxy is not only constrained to the surface of the object but also moves to the point on the surface which results in the shortest distance between the proxy and the HIP. At each frame, the proxy attempts to update its position to the HIP by direct motion. During this update process, if it encounters any primitive in the scene, its motion is restricted to follow that primitive as a constraint such that the shortest distance to the HIP is found while lying on the constraint primitive. Each primitive of the surface acts as a constraint on the proxy, and finding the location of the proxy becomes a constrained optimization problem. Haptic rendering algorithms for other descriptions of objects in a scene have also been proposed. These include voxel-based surface definitions (Laehyun et al., 2002; McNeely et al., 1999), implicit surface representation (Hua & Hong, 2001; Salisbury & Tarr, 1997), hybrid surface representations (Laehyun et al., 2004) and NURBS models (Thomas V. Thompson & Cohen, 2005; Thomas V. Thompson et al., 1997).

Another way of categorizing haptic rendering techniques is based on the type of interaction (Srinivasan & Basdogan, 1997) provided by the haptic probe: point-based interaction (Salisbury et al., 1995) where the end-effector point interacts with objects in the scene, ray-based interaction (Basdogan et al., 1997) where the avatar is modelled as a finite ray that interacts with objects, and object based interaction (McNeely et al., 1999) where the haptic probe is modelled as a 3D object.

Scene geometry is not the only feature that needs to be haptically perceived. Texture and material properties of an object along with its geometry define its complete haptic nature. A number of techniques of haptic texture rendering exist, which can be broadly classified into two classes: image-based techniques and procedural haptic texturing

methods (Basdogan & Srinivasan, 2002). In the former, a texture field is constructed from a 2D image and the texture mapping techniques of computer graphics are employed. Forces are perturbed based on height value of each point resulting in haptic perception of the surface texture (Ho et al., 1999). Procedural haptic texturing on the other hand is used to generate synthetic textures using mathematical functions. The height field is defined by a function taking coordinates as input and returning height value and gradient which can be used to perturb force direction (Siira & Pai, 1996). There have also been other attempts at rendering haptic textures such as recording real surface textures using haptic devices and playing them back (Vasudevan & Manivannan, 2006).

A number of challenges exist in haptic interaction with 3D virtual environments. Firstly, it involves creation of complex 3D models for most of the objects in the scene if not the entire virtual space. Although, the processes of creating and manipulating 3D models have become easier with devices like 3D scanners that are becoming more and more affordable, the task of obtaining models that match real world quality is still laborious and time consuming. Moreover, in many cases accurate modelling and believable reconstruction of the 3D scene may not be feasible at all due to environment and application constraints.

In addition, the size of these models increases proportionally with the details provided by them and rendering large numbers of polygons at the update frequency of the haptic pipeline remains a challenge. Typically, the polygon count for which smooth haptic interaction can be performed is way below the maximum number of polygons that can be visualized smoothly. It has been reported in (Wei, 2011) that for smooth haptic rendering the haptic update rate can only be achieved with up to one million polygons

where similar system configuration is capable of smoothly rendering several million polygons visually. A number of approaches have been proposed to solve this problem such as using bounding volumes, spatial partitioning, GPU based acceleration and velocity driven rendering approach. However, computational complexity of visual and haptic rendering increases further when the end-effector of the haptic device is represented by a complex geometric object. Object-object collisions and object-avatar collisions become computationally more demanding. Adding physics simulations would reduce performance further. Moreover, the size of 3D scenes becomes critical in shared virtual spaces where the amount of data exchanged between peers needs to be reduced as much as possible due to limited memory and transmission bandwidth.

Thus, haptic interaction needs to be enabled with other scene representations which ensure small size and guarantee photorealistic quality, e.g., two dimensional images and videos. Also, a large proportion of the existing multimedia content is two dimensional images and videos. Adding another modality to interact with this content makes the user interaction experience more intuitive and realistic.

2.3 Haptic interaction with 2D images

In this section, we review in detail existing works that can be used to realise the idea of adding haptic interaction to two dimensional images and videos. Since different applications of 2D tangible media have different requirements, approaches to make 2D images touchable can be classified in a number of ways.

Firstly, the techniques can be classified based on the type of interaction required by the application. Some applications may require an entire scene in an image to be

haptically accessible while other applications may require only certain parts or regions of the image to be “touched”. Some applications may require the user to follow certain paths and only these trajectories need to be accessible through touch. In case of video streams, some applications may require haptic interaction with only selected frames of the video.

Another way to categorize methods of exploring images by touch can be based on the freedom of interaction offered to the user when interacting with the 2D image. Some applications may allow the user only to passively perceive the objects depicted in the image through touch. On the other hand, there may be applications where the scene responds to user actions. The user can trigger events in the scene, manipulate the objects in the image or modify the scene.

However, a broader classification of the techniques of haptic interaction with 2D images can be made based on the information used to compute feedback forces delivered to the user. Thus, we consider the following two classes:

1. Firstly, there are methods that generate feedback forces based on information present within the image or information derived from the image.
2. Secondly, there are methods that use additional supplementary information along with the information present in the image to calculate feedback forces.

2.3.1 Direct computation of forces from 2D images

A number of methods of adding touch interaction with 2D images based on information available in the image have been proposed. Since these methods do not require any additional data to produce haptic effect, force generation can often be performed on-the-fly. Various types of information are available from the image directly or

indirectly such as colour, intensity, histogram, etc. One or more of these attributes of the image are used by this class of techniques to generate force feedback. These techniques often put stringent constraints on the images for which haptic sensation can be generated. This is due to the fact that the content of the image does not have much information about haptic sensation or material properties of the scene, e.g., it is difficult to predict if an object is hard or soft just by looking at its image. Even though people can to some extent predict material properties of a scene based on their knowledge and experience of the real world, computers cannot achieve such accuracy in predicting physical properties of objects in an image.

Often these techniques use illumination and shading information to predict haptic features of the scene. The principle behind using shading information is that convex surfaces are illuminated brighter than concave surfaces given that the image is obtained with frontal lighting. Some of these shading-based methods assume a direct mapping from the visual modality to the haptic modality. For example, the work in (Vasudevan & Manivannan, 2008) utilizes shading information in the image for haptic texture generation through Prewitt filters. The filter is applied only in the vicinity of the haptic interface point making it applicable to real-time applications. The size of the neighbourhood on which the filter is applied is chosen so that all the details can be haptically perceived while avoiding the effect of noise on haptic perception of the scene. The higher the contrast in the image, the higher is the magnitude of gradient of intensity. Therefore image contrast controls the magnitude of the force vector. Another example of direct conversion of shading information into haptic forces is presented in (Li et al., 2010b), where haptic textures are rendered using image processing routines. The force vector is

composed of two components: a normal and a tangential component. The magnitude of the normal component is based on colour temperature or luminance values in the image. Tangential component is based on relative colour variation between haptic interface point and the neighbouring pixels.

Other shading-based methods employ 3D reconstruction to generate geometric surfaces for haptic rendering. For example, the authors of (Li et al., 2010a) use the shape-from-shading (SFS) algorithm of Tsai and Shah (Ping-Sing & Shah, 1994) to construct a height map which is used to construct a virtual surface. Haptic texture is then rendered from this surface using a spring model for calculating the magnitude of the normal force. 3D reconstruction is also used in (Shi & Pai, 1997), where 3D models are reconstructed from 2D stereo pairs and haptic interaction is provided with the 3D model through a 2 DOF haptic device. In (Basdogan, 2007), multiple images are used to generate 3D models for haptic rendering while auto-stereoscopic display is used to visualize these 3D models. Closely related are other works that infer from a 2D image three-dimensionality of objects in some form. So, far these works have not been used for image-driven haptic rendering but they can be used as inspiration for haptic interaction with 2D images. For example, one method of 3D reconstruction from a single image is given in (Hoiem et al., 2005). The 3D reconstruction model uses a single image as input and produces a piecewise planar model in VRML but it is not guaranteed to work for any image. The process involves grouping pixels into super-pixels which are then grouped into constellations. These constellations are labelled as sky, ground or vertical plane. Classification of image regions is based on colour, texture, 3D geometry and horizon position. The method generally works well for outdoor scenes. Scenes with crowds or a

large number of slanted and irregular edge structures are not well reconstructed. An illumination based reconstruction method presented in (Siu-Yueng & Chow, 2001) uses neural networks and radial basis function to estimate the total reflection in a scene. The method presented however produces unrealistic geometry when illuminating light is not frontal.

A number of shading based reconstruction methods have been proposed over the years. A survey that compares six different implementation schemes of SFS algorithms is presented in (Zhang et al., 1999). The performance is evaluated based on synthesized images. Mean and standard deviation of depth and gradient error along with CPU timing are used to compare performance. It classifies techniques into three categories. First, there are minimization approaches that work by minimizing an energy function. Secondly, there are approaches that approximate image irradiance. Such an approximation can be based on the assumption of type of surfaces or can be based on computation using a linearization of the reflectance map. Lastly, there are propagation approaches which propagate shape information from a set of surface points to the whole image. The paper concludes that each algorithm works well for some images and is not suitable for others, and minimization approaches are more robust while others are faster. Another updated survey was later presented by Durou et al. (Durou et al., 2008), which classifies SFS methods into 3 classes: methods involving resolution of PDEs, minimization based methods, and methods approximation image irradiance. In PDE-based methods some boundary conditions are always involved which are usually unknown for real images. Even if they are known, these boundary conditions fail to solve the convex/concave ambiguity. Minimization methods consist of a set of unknown variables, a cost function

and an optimization method. A number of attempts have been made using various combinations for these cost functions and optimization algorithms, but each of them has certain shortcomings primarily due to the inherent assumptions of the SFS algorithms. The assumptions include surfaces having a constant albedo, orthographic projection, distant point light sources, and absence of cast shadows. Effects of inter-reflections or secondary illumination are also ignored. Real world on the other hand has diffuse light sources, inter-reflections between surfaces, varying light absorption of surfaces and other such phenomena. A more recent attempt to solving SFS problem without boundary conditions is presented in (Ecker & Jepson, 2010), which provides a complete solution for ideal polyhedral objects. It uses semi-definite programming (SDP) relaxation with new smoothness constraints for folds at edges. This method is also based on the assumption of Lambertian reflection model. In (Alexa & Matusik, 2010), a single image is used to construct a relief surface whose diffuse reflection is such that it matches the image under known illumination direction. This is done by using a discrete model for the area in the relief surface that corresponds to a pixel in the desired image. An iterative least squares optimization is used to determine the resulting relief. The height of the relief is kept intentionally small and viewing direction is assumed at $(0, 0, 1)$.

Partial 3D information about a scene can also be inferred by recognizing objects in an image and selecting their corresponding 3D models from an online database (Klank et al., 2009). Alternatively, a 3-Sweep method recently proposed by Chen et al. (Chen et al., 2013) allows for interactive extraction of simple 3D shapes from a single image by understanding the components of the shape, their projection and interrelation.

For image-based haptic rendering, it is usually assumed that the image brightness

intensity matches the height map of the scene to some extent as in (Juan et al., 2007), where Gaussian filters were used to smooth the input image to generate haptic textures. The resulting force vector is based on the magnitude of height map and depth of penetration into the surface. In (Xu et al., 2012), the Gaussian filter was replaced by an improved switching median filter to make sure that the fine details in the image are preserved. These methods are applicable only to images of objects of the type that produces an image which reflects its height map.

One method of generating tactile textures from image data is presented in (Ikei et al., 1998). It first establishes a correspondence between image data and geometrical shape data, and then transforms the histogram based on template matching. User tests are applied to select the appropriate template as well as for validating the results. The input image is however subjected to strict criterion, e.g., the histogram distribution should cover the brightness range moderately to avoid sharp contrast and that the texture size should be smaller than the fingertip size. In (Adi & Sulaiman, 2009), visual textures are represented in terms of Fourier series, and wavelet transformation is used to generate haptic texture from the visual texture information. The surface is obtained in the form of sine waves and wavelet coefficients from image data. The resulting height map is produced without the use of a 3D scanning device. The technique has the advantage of generating haptic textures in real-time; however has stability issues in haptic rendering of large force changes.

Contours or edges in an image are a prominent feature that helps us analyse the scene in an image. Edge information has been used to produce haptic interaction with 2D images for the purpose of image editing and enhancement (Kagawa et al., 2009; Kagawa

et al., 2010; Kagawa et al., 2007). The image is represented as a 3D plane in the virtual environment, and each edge is represented as a crest or a trough. This is particularly useful to trace contours of objects in the scene which is hard to imitate accurately without haptic feedback. The surface of the plane is represented as a sine function with frequency, amplitude and phase modulated by luminance, colour and visual texture information. The resulting height map and the edge information provide basis for the image edition tools presented. In (Wang, 1999), object contours are extracted as a boundary map. These edges are haptically perceived as wedges that peak at object boundaries. A haptic image exploration system presented in (Lareau & Jochen, 2012; Lareau & Lang, 2013) transforms contours in an image to 3D grooves and uses them for haptic rendering. Moreover, it allows haptic exploration of the image at different levels of detail through hierarchical image segmentation and also attempts to encode relative sizes of segments in haptic textures as decaying sinusoidal vibrations. Edge information has also been used for automatic visual to tactile conversion (Way & Barner, 1997), contour tracing (Janabi-Sharifi et al., 2007), Multi-Modal Image Display (MMID) system (Wang, 1999), enabling haptic interaction with vector graphics (Krufka et al., 2007) and multi-resolution curves (Pai & Reissell, 1997).

A few methods of generating 2 DOF feedback force from a single image have also been presented. In (Kim & Kwon, 2013), the authors estimate pose of planar objects in the image to perturb surface normal of each plane so that assistive or resistive forces can be applied to the force feedback device. Minsky (Minsky et al., 1990) proposed a ‘sandpaper’ system for synthesizing haptic textures from 2D images by generating 2D forces while Lecuyer (Lecuyer et al., 2004) presented a method of generating

pseudo-haptic textures by varying the visual cursor speed to create the illusion of climbing up or down a bump without actual force feedback.

Other visual cues, such as deformation, elasticity, segmentation, etc., have also been used to produce haptic sensation of physical features of the scene. For example, (Jungsik et al., 2010; Jungsik et al., 2007) propose an artificial reaction force when a slave system performs manipulation tasks on deformable objects such as a tissue cell. The proposed approach uses image processing methods to recognize objects in the captured image and generate 2D deformable models. Feedback forces are generated by estimating physical phenomena such as elasticity from the deformed image. Forces calculated at low update rates are extrapolated to achieve 1 kHz haptic update rate.

2.3.2 Force computation using additional information

The information present within the image is usually not sufficient for providing a sense of scene geometry. Deriving physical properties of the scene from this information is even more difficult. Therefore, supplementary information is sometimes added to 2D images which can be used to provide geometric cues and material property information for the objects in the scene. A number of approaches have been proposed following this idea, the majority of which are targeted towards broadcast media applications. Additional information is transmitted along with the 2D video stream to the client computer. This information, which has some correspondence to image content, is used in combination with the information within the image to generate haptic information at the receiver.

Gaw et al. (Gaw et al., 2006) proposed a system for annotating recorded haptic information to movies in a time referenced manner. It can be played back to the user who

feels the recorded haptic sensation. An authoring environment with several tools for editing haptic force information is proposed. Force information and position trajectories are augmented with the video at selected frames. The viewer grasps the haptic handle and receives haptic feedback while viewing the video. The client device follows the trajectories prescribed by the author of the movie and the user is guided through the scene. Such haptic feedback is useful to teach specific motor skills.

When adding haptic sensation to images/videos, sometimes the image or frame is divided into a grid of cells and the entire cell is assigned a single haptic sensation. One such method is presented in (Mee Young et al., 2009), where each frame of the video stream is divided into a grid and each cell is assigned a ‘Material ID’. A table specifying relationship between Material ID and physical properties is formed. Table 2.1 is a sample table that illustrates this relationship.

Material ID	Physical properties (normalized values)			
	Stiffness	Damping	Static friction	Dynamic friction
1	1.0	1.0	1.0	1.0
2	0.5	0.5	0.5	0.5
3	0.0	0.0	0.0	0.0
...	...			

Table 2.1 A sample table formed by assignment of different physical property values to different Material IDs.

Corresponding to each video frame, there is a ‘haptic frame’ which contains Material ID for each cell. The video stream received by the viewer includes the video frames, the audio frames, haptic info (the table), and the haptic frames. As the user navigates through the scene, the feedback force is computed based on its Material ID in the corresponding haptic frame and the associated physical property values. The authoring procedure is time consuming and the values associated to each Material ID are based on user tests. Since

different people have different perception of physical properties, such an assignment may not be entirely accurate.

Depth information is often used as the additional data source for generating haptic feedback from 2D images. A depth map consists of distances from each pixel in the image plane to the surface of the object. Such depth maps have a simple and regular structure so they can be processed and compressed using existing image manipulation and compression algorithms. A novel method of Depth Image-Based Representation (DIBR) for representing static and animated objects is proposed in (Bayakovski et al., 2002). A set of reference images that cover the entire 3D object combined with corresponding depth maps are used to represent the object. Real world scenes can thus be rendered without millions of polygons and the influence of the geometric complexity of the objects on the rendering time is eliminated. The presented representation has been adopted into MPEG-4 Animation Framework eXtension (AFX). In (Cha et al., 2008; Cha et al., 2005a), Cha et al. presented a method of haptic interaction with depth video media. The depth information is synchronized to the colour video stream in a per-pixel per-frame manner and transmitted along with the video stream. The authors propose a modification to the proxy graph algorithm which is a polygon-based haptic rendering algorithm for very large static topographic maps (Walker & Salisbury, 2003). The proposed technique is capable of detecting collisions between a virtual point and time-varying scenes while avoiding sudden changes in contact forces. The idea is to triangulate the depth image using a vertically monotone structure. In other words, the depth values are mapped into an array of elevations in pixels and their tips are connected horizontally, vertically and diagonally. Once a collision has occurred, the proxy is allowed to move into the object only through

triangular primitives and not through edges or vertices. Surface properties are applied through friction cone algorithm and depth value perturbations based on bump maps. These Depth Image-Based Haptic Rendering (DIBHR) algorithms were used to construct a 3D video player system in (Cha et al., 2006). It consists of a haptic subsystem in addition to generation, compression, transmission and display subsystems. Using depth video as geometric data, 3D stereoscopic view is presented by the graphic renderer. Haptic renderer tracks the user position from the interface and calculates the feedback force based on the depth video using the modified proxy graph algorithm.

The idea of using depth information for haptic interaction with 2D videos was also employed in the touchable video system presented in (Cha et al., 2009). The authors propose a complete 3D video system where the viewers can touch a scene with haptic devices. The video stream consists of four images per frame: a 24-bit colour image (three 8-bit channels for RGB), a 24-bit haptic image (three 8-bit channels for stiffness, static friction and dynamic friction), an 8-bit depth image, and an 8-bit heightmap image.

Physical property (stiffness, static friction and dynamic friction) values are assigned to each pixel using a single image with three 8-bit channels to ensure a format similar to an RGB image. The heightmap image is used to represent roughness, i.e. micro geometry, while the depth image represents the macro geometry. Such a representation provides for image processing and compression techniques to remain applicable however intrinsic camera parameters must be known at the receiver for 3D reconstruction. Content generation can take place either online by using depth cameras and tactile sensors or offline by manual assignment of physical properties. Transmission of such content is proposed based on MPEG-4 framework by specifying a Binary Format for Scenes (BIFS)

node. At the receiver, the viewing and interaction are performed by using proxy-based haptic rendering methods. In order to speed up processing at the receiver, 3D is reconstructed in the neighbourhood of the haptic interface point only. Due to their simplicity, median filters are used to remove noise in the depth image. These filters are also applied locally as the rendering algorithm only uses depth values in the small area around the HIP. The presented system has the advantage of real-time rendering at the receiver but is only suitable for pre-processed content. Moreover, due to the dynamic nature of videos, some collisions may go undetected.

Another method of haptic interaction engaging 3D reconstruction based on depth information presented in (Kim et al., 2006) uses depth enhancement mechanisms to eliminate ruggedness in haptic rendering of smooth surfaces. A two-step smoothing algorithm is used. In the first step, median filtering is used in 2D image space for noise removal, then first gradient analysis is used to adaptively sample feature points which are then subjected to Delaunay triangulation. Secondly, Gaussian smoothing is used for 3D mesh refinement. In the presented method, only moving objects are incorporated in the depth map and the haptic rendering algorithm occasionally fails to detect collisions at frame transition.

A method of haptic interaction with depth augmented video data that does not require 3D reconstruction is presented in (Dindar et al., 2010). The depth map is first segmented to get object depths and an approximate mass is assigned to each object. Then velocity fields for objects are computed using the Lukas-Kanade optical flow estimation algorithm. This velocity field information is used to compute the dynamic component of the force while static component is calculated based on depth values. The resulting force, which

consists of static and dynamic components, is interpolated to match the haptic update rate of a thousand frames per second.

In (O'Modhain & Oakley, 2003), O'Madhorain and Oakley study two scenarios of touch interaction with video streams in a television: authored content such as cartoons and real-time content such as sports broadcasting. The feasibility study presented concludes that the feedback devices for such applications must integrate into the real world easily and provide rich haptic sensation while being cost effective. The generated haptic feedback may not correspond to any objects in the scene rather it may correspond to certain events in the video, e.g., a cartoon character jumping off a cliff could be mapped to vibration at a certain frequency. However, this requires the user to be trained to the system. For real-time content, the use of sensors, that capture physical properties from the scene, is proposed, e.g., by attaching accelerometers to a vehicle. The generated feedback could be felt as vibrations through a haptic device such as a couch-shaker. It distinguishes active and passive interaction based on scene responsiveness to user input. A touch-TV scenario with active touch, where the scene responds to user input, is proposed in (O'Modhain & Oakley, 2004). The idea is that the haptic and visual scene starts and ends at the same instant as authored irrespective of the user interaction. However, input from the user can result in object manipulation within the scene. Another haptic broadcast media system based on four stages is presented in (Cha et al., 2005b). In the first stage, the scene is captured with depth information using depth sensing cameras. Haptic information is also captured using sensors such as an accelerometer inside a sports ball. Spatial and temporal coordination of data for different modalities is performed manually in the edit stage. In the transmission stage, the authored program is sent to the viewers

through the Internet. The last stage takes place at the receiver side, where the video and haptic data is processed to produce a stereoscopic display and force feedback respectively. More recently, an authoring/editing framework has been proposed (Cha et al., 2007), which provides passive haptic sensation through kinaesthetic as well as tactile feedback. The data is transmitted using MPEG4 BIFS with separate nodes proposed for kinaesthetic and tactile sensation.

A Haptic Generation Filter (HGF) is proposed in (Yamaguchi et al., 2006), which analyzes metadata of a Flash animation to list property values of objects and generate force feedback. Haptic effect is generated by mapping the dynamic movement of objects in the 2D graphic scene to a physical world using metadata of the scene data. Information such as position, height, width, rotation, etc., is extracted from metadata and bounding boxes are created for each object. Acceleration and amount of motion is calculated for each object, and based on these values force is computed and exerted by each object on a colliding pointer/avatar. The haptic rendering algorithm assumes a direct relationship between the size of an object and its weight. The calculated force feedback is delivered to the user through a 2 DOF haptic mouse.

Depth information obtained from a capturing device has been used for real-time haptic interaction with 2D video streams in (Ryden et al., 2011). The authors use Microsoft Kinect as a depth sensing device to capture point cloud representation of the scene. Haptic virtual fixtures are then generated automatically from the point clouds in real-time. The proxy-based algorithm involves calculating force vectors for each point in a defined neighbourhood of the haptic interface point and averaging them. Although the proposed method has applicability to video streams due to its real-time nature, it fails to

provide haptic sensation of physical properties. Moreover, due to noise in the depth map and a lower resolution sensor, rigid objects can only be rendered with limited stiffness.

When adding haptic interaction to existing 2D images, it is important to preserve compatibility of the content with current image formats and compression techniques. Kim et al. (Seung-Chan et al., 2011) propose a method to embed haptic information into 2D images by using least significant bit (LSB) manipulation for hiding haptic information in the image. Human vision system has a low sensitivity to high frequency brightness variations which allows for modification of the input image without significant perceivable distortion. Geometric information is defined as a depth map while stiffness and viscous damping information is used to represent physical properties of the scene. The depth, stiffness and viscosity maps, represented as a 2D array of the same size as the image, are compressed before encoding them into the image. During the encoding process, a header is constructed such that the images can be decoded without any external information. At the time of interaction, a 3D polygon mesh is reconstructed after extracting the haptic maps from the coded image using the header information. Force rendered to the haptic device is calculated based on position and velocity of the HIP, and physical property information in the extracted maps. The authors of (Seung-Chan et al., 2010) have also demonstrated the use of LSB manipulation technique to embed object descriptors into 2D images as metadata. Rigid body objects can be reconstructed by decoding this information during rendering and the user can interact with these objects while visualizing the 2D image.

2.4 Critical analysis

The advantage of methods for force feedback computation from images directly lies in the fact that force feedback can be calculated without the need of any supplementary information related to the scene. Moreover, being computationally less expensive, these techniques can usually perform haptic rendering on-the-fly. However, there are several disadvantages of these techniques. Firstly, most of these techniques limit the types of images that can be used for haptic rendering. This is due to the strict requirements of the algorithms used for extraction of shape from 2D images. Secondly, these techniques mostly offer haptic texture perception. While this is useful in image-based textures, geometric perception is an important feature that needs to be haptically accessible for immersive experience in a scene represented by images. Stability is another issue with most of these techniques when the force magnitude changes abruptly following sharp contrasts in 2D images. Sometimes these techniques also have other limitations such as haptic feedback limited in the number of DOF.

Existing techniques, which use supplementary information along with the 2D image for force feedback generation, also have certain disadvantages. Firstly, acquiring this additional information needs additional effort. Whether done through sensors, manual assignment of physical properties or assignment based on user tests, it is still challenging to obtain accurate information about physical properties or geometric structure of the scene. Secondly, authoring and editing such content also requires extensive interactive preprocessing. Sophisticated authoring tools must be developed for synchronization of the obtained supplementary information with different parts of the scene in multiple frames. Some techniques also use 3D reconstruction at the client computer for haptic rendering.

However, 3D reconstruction followed by haptic rendering requires significant computational resources and often fails to provide high precision as well as overall consistency of the reconstructed scenes.

We therefore conclude that there is a room for research in direct haptic interaction with images without full 3D reconstruction. We understand though that it may not be possible to come up with a single method applicable to all images universally for producing feedback forces for perception of visible geometry, texture, edges and material properties. Instead, a set of methods with user adjustable parameters is needed where the user can tune the parameters best suited to the desired application. Moreover, we do not have an ambition to replace the traditional haptic modeling and 3D visualization but we foresee that some applications can be implemented much faster and with a believable visual and haptic immersion when image-based approach is used.

2.5 Summary

In this chapter, first we reviewed principles of haptic interaction in virtual environments along with haptic interface hardware and software. We then presented a survey of existing work related to haptic rendering of scenes displayed in two-dimensional images and video. It has been categorized into two classes: using information derived from the image and using external information presented alongside the 2D image. The techniques presented in the first category usually provide for haptic perception of texture only and fail to render a sense of haptic structure, i.e. geometry, or physical properties. These techniques often have limitations such as haptic feedback limited in the number of DOF, unstable forces at abrupt force transitions, computationally intensive reconstruction processes, etc. Methods

that involve external information in addition to 2D images often require lots of pre-processing in authoring/editing stage. Moreover, 3D polygon meshes are usually generated from the augmented information which requires powerful client computers. Physical property information is either added based on sensors in the scene or is subjectively assigned, e.g., by using user tests. Use of such sensors is not always feasible due to dynamics of the scene in different applications such as sports broadcasting or minimally invasive surgery.

Chapter 3 Tangible Images – An Approach to Image-Driven Haptic Interaction

In the previous chapter, we surveyed methods which can be used in haptic interaction with images and concluded that there is a room for research. In this chapter, in Section 3.1 we come up with our hypothesis, and in Section 3.2 we elaborate the proposed approach.

3.1 Hypothesis

Through vision we obtain the majority of the information about the environment. It is a passive information collection process where we perceive the picture of the scene all at once, analyze it mentally, deduce its spatial properties, and anticipate its physical properties based on our experience and common sense. Therefore, with a computer the image has to be rendered with as high quality (level of detail) as possible to make the user believe it. Vision guides us to explore the objects in the scene haptically. Haptic interaction is an active and bi-directional process of collecting information by touching the objects, exerting forces to them, as well as sampling forces exerted by the objects or various phenomena which have no visible appearance (e.g., air flow, pressure, etc.). Therefore, we are prepared to feel what we see and to collect through touch the missing information complementing our vision experience. Given that most desktop haptic devices have only one haptic interaction point, it is essential to see the image to direct the haptic cursor to the desired part of the scene. Since it is always a 2D image in the end, whether it was rendered from 3D models at the time of haptic interaction or obtained by any other ways, the user may see no difference provided the observer position does not

change much. In this research, we make maximum use of the 2D images or actual photographs to calculate the feedback forces as if the haptic device interacted with the 3D scene displayed in the image.

We hypothesize that images of real scenes can be turned into “tangible images” that can be felt with haptic devices as if they were real 3D scenes following the principle “What You See Is What You Touch”. This can be achieved by mostly deriving haptic forces for visible surfaces of the objects straight from the images with pixel precision, as well as by segmenting the images and assigning physical properties to their parts, and by augmenting the images with invisible haptic objects. The scenes displayed in the images can also change following the haptic interaction either by image warping or by augmenting them with editable parts.

Ideally haptic interaction with 2D images should provide for perception of various features. Since the overall geometrical structure is one of the most important haptic cues in a scene, therefore tangible images should include an algorithm for haptic rendering of the scene geometry. Next, the fine texture details on the surface of each object must be rendered for a realistic tactile feel of different objects. Also, tangible images should render forces calculated based on physical properties of each object while the difference in physical properties of various objects should be clearly perceptible. For example, the user should be able to feel the softness, elasticity of objects, and various effects in the scene such as wind, liquid flow, etc., while being able to see the visual changes in the scene, such as movement or visual deformations of the objects. In some applications, the user may be interested in exploring edges or object contours in an image. Therefore, a tangible image should have a mechanism for emphasizing the edges or silhouettes in the

scene. Moreover, based on the requirements to the application under consideration, a tangible image system should allow for haptic rendering of individual features such as geometry, texture, edges, etc., along with a combination of any of these features. This will additionally ease the workload on the computer as the visual rendering of 3D scenes will be replaced by merely displaying their images while simulating haptic interaction with the displayed scenes.

Therefore, the **research goal** of this thesis is to develop an image-driven haptic rendering paradigm that

1. Allows for haptic exploration of the visible object's geometry, texture and physical properties including surface properties (e.g., friction and elasticity), as well as inner properties (e.g., viscosity). Achieving this goal is described in Chapter 4.
2. Includes a mechanism for modelling changes to the scene in response to the user manipulation. Algorithms proposed to achieve this goal are described in Chapters 4 and 5.
3. Allows for feeling natural phenomena that do not have clear visual appearance, such as wind, liquid flow, etc. Chapter 5 details the algorithms proposed to achieve this goal.
4. Simulates realistic interaction by supporting multiple contact point interaction. This goal is achieved in Chapter 6.

3.2 Approach, research and development novelty

To fulfil the research goals and hence verify the hypothesis, we propose a novel image-driven haptic rendering framework for generation of force feedback based on

analysis of image shading, calculating the visible depth, and using additional information that can be associated with the image.

Depth estimation and haptic rendering: In our approach, the core algorithm is based on the idea of obtaining the depth coordinate for each pixel by interpreting the pixel colour, as well as the colours of the pixels in its vicinity, in terms of visible depth. Haptic feedback forces are then calculated at each pixel, which involves calculation of the force direction and force magnitude. Both are calculated in such a way that there is no abrupt change in their values resulting in jitter or sudden kick back motions of the haptic device handle.

There are also three important cases that have to be processed separately, or rather as important application modes. First is the interpretation of colour patterns on the visible surface of the objects as changes of colour rather than changes of their geometry. This requires us to filter out sudden changes of colour while performing the depth calculation task. The idea of the method can be explained as setting up a threshold value for the changes of dominant colours while analysing pixels visited by the HIP. If the colour change exceeds the threshold, it is assumed that the HIP continues to move along the surface of the displayed object with the same depth as that for the previous pixel and rather relative changes of the next dominant colour are used for further depth calculations. For example, while performing haptic rendering of an image of a face with fair complexion, the HIP will not be “diving” into dark eyes or hair. Since real photographic images usually have blurred transitions at object boundaries, the contrast of the input image is stretched and the image with emphasized contrast is used to detect the colour patterns.

The second case to be processed separately is based on emphasizing contours of the visible objects. This case is essential for haptic interaction with displayed user interfaces or moving images. In contrast to the existing approaches, we do not detect or store edges of the objects but rather further extend the previously described approach with the colour threshold while using it as a trigger for detecting the fact that the sudden change of colour is a result of touching the contour of the object. Various force profiles can be used to reliably keep the HIP outside the visible contour. The core algorithm of the force feedback calculation is modified to produce a repelling force that affects the HIP while it is located within the same dominant colour. This mode also permits processing of videos to reliably detect the boundaries of the moving objects.

The third case requires emphasizing tangible textures. This mode is useful when the scene in the image has a small change geometrically but the texture details have to be explored. It is achieved by stretching the ‘usable’ range of the image histogram. The range is defined by two parameters chosen by the user. Any pixels that lie outside this range are clamped to the respective extreme of the chosen histogram range.

Interaction with hybrid scenes: Depth estimated from illumination and shading characteristics of the 2D image may not be always accurate as the bas-relief ambiguity states that neither shading nor shadowing of an object seen from a single viewpoint reveals its exact 3D structure. Although an approximation using some automatic or interactive image correction algorithms may suffice for certain applications, e.g., minimally invasive surgery images, but this approach still restricts the images to follow the strict criterion. To eliminate this constraint, we use, where it is feasible, depth data obtained from other sources such as RGBD depth-sensing cameras, e.g., Microsoft Kinect.

We can also use depth information obtained by other sources, such as stereo-images, subsequent frames of videos, etc. For scenarios where this is also not possible, additional haptic information that can be associated with the image is used. This supplementary information can be in the form of invisible haptic models collocated with objects displayed in the 2D image. Consider, for example, a region of the image showing a river stream. A function-based haptic 3D model with invisible geometry but physical properties defined by mathematical functions can be collocated with the image. As the haptic cursor will be moved into the stream, and the collocated haptic model, forces representing water flow will be applied to the haptic device.

Interactive modifications to the scene displayed in a 2D image: Active haptic interaction is an important aspect of achieving immersive experience as the elements of the scene are expected to change in response to user input. In other words, when a user manipulates the objects in the scene their position, orientation or shape may change. Therefore, a tangible image system should provide this functionality of scene response to user triggered changes. This is achieved by adding editable parts to the image either in the form of visible 3D objects or 2D texture patches. In case of visible 3D objects, the 3D foreground and 2D image background merge into one image that is processed by the uniform image-driven haptic rendering algorithm and hence makes tangible images editable. On the other hand, small 2D texture patches of modified scene elements are used to superimpose the 2D image at the point of interaction. Thus, apart from the haptic sensation received from the scene, visual changes, such as deformation, can also be seen. Another method that has been used in applications based on this research is to model the deformation in a scene by distorting the image through image warping. Selective regions

of the image are made available for such distortion by using existing image segmentation techniques, and the amount of distortion within a region is defined by analytic functions.

Multiple contact point interaction: Typically, desktop haptic devices offer single point of contact interaction. However, to provide a better interaction experience we extend the image-driven haptic rendering algorithm to provide multipoint interaction with single contact point desktop haptic devices. Different points are predefined on the 3D cursor based on their relative distances to the HIP of the device and the haptic rendering algorithm computes forces for each of these points. A resultant force computed based on these forces is applied to the HIP. This is particularly useful in some applications such as minimally invasive surgery where the view is magnified manifold and haptic cursor is no longer a single point but rather a significantly magnified 3D object.

Usefulness of the tangible images framework: The proposed tangible images can be used in many practical applications ranging from virtual museums and cultural heritage applications where images of the exhibits can be haptically explored without the need for their detailed 3D reconstruction, to training and simulation where images and video of real or virtual scenes can be used as a source of haptic interaction with the displayed objects. For example, many surgical operations are performed using minimally invasive techniques where the surgeon sees the surgical field as an image on a video monitor while operating remotely controlled surgical instruments. Given that the cost of 3D reconstruction of each surgical case can be very high, using examination videos for creating tangible panoramic images leads to much faster and efficient simulations. Individual modes can be suitable for different applications, e.g., contour or silhouette emphasis can help assist visually impaired users when interacting with graphical user

interfaces. Last but not least, we anticipate networked applications where the participants will be able to not only see each other in streaming videos, but also touch each other's image and get a feeling of real life interaction.

Based on this approach, this thesis produces the following **research novelties**:

1. We propose a set of image-driven haptic rendering algorithms for deriving three degrees of freedom haptic forces from images without performing 3D reconstruction of the scene but by calculating the force-feedback for each pixel of the image "touched" by the haptic interaction point on the fly. Different features are made available for haptic exploration along with methods for interactive changes to the scene displayed in the image.
2. We enrich the proposed tangible images by augmenting them with additional information in form of invisible haptic models of the objects displayed as well as 3D foreground objects which serve a purpose of simulating editable parts of the scene. We use mathematical functions and procedures to augment images ensuring a small size of the final tangible image.
3. We propose an extension to our haptic rendering algorithm that enables us to use common desktop haptic devices with one haptic interaction point for solving haptic rendering problems that require multiple haptic interaction points.

To prove the practical values of the project, we present the following **innovative implementations**:

1. Training software has been developed for simulating arthroscopic surgery procedures where the actual surgical videos are converted to tangible panoramic images. The

software is developed to teach the trainees how to move and apply the surgical instruments while observing the surgical area on video monitor.

2. A desktop software tool is developed for haptic perception of common visual interfaces as well as any visual information displayed on the computer monitor.
3. A haptic-enabled video communication tool is developed. It makes a Skype video tangible so that the involved parties are able to not only see each other but also to touch the video stream.

3.3 Summary

In this chapter, we analysed the work of human perceptions and defined the goal of developing a framework for image-driven haptic rendering which provides a force feedback calculation mechanism for various haptic features of a scene. We presented an overview of the algorithms proposed in this study. These include a core algorithm for geometry rendering, and its three application oriented modes for emphasizing texture and contours, and for neglecting colour discontinuities that do not represent geometric changes. Supplementary information in the form of function-defined 3D models is added to the image for rendering editable parts of the scene or invisible effects such as wind, flow, etc. This additional information can also be in the form of 2D texture patches that blend into the image background to simulate interactive changes in the scene. Different applications have been developed to illustrate the usability of the proposed approach. These include an arthroscopy simulator, a haptic-enabled video communication tool and software for haptic perception of common visual interfaces.

Chapter 4 A Novel Algorithm for Depth-Based Haptic Rendering

In this chapter, we present a novel algorithm for haptic rendering of scene geometry based on depth information that may be either estimated directly from an image or acquired in some other way. The hypothesis is validated by calculating haptic forces based on the depth values in the local neighbourhood of the haptic interface point. Three different application modes for emphasizing different features of the scene are discussed for the scenarios when the depth is estimated from pixel intensities in the image.

4.1 Depth-based haptic geometry rendering

An image is an array of pixel values representing intensity of the point as seen by the camera. In case of a colour image, each pixel is represented by three values representing the intensity of each of Red, Green and Blue in the RGB colour space. Other colour spaces such as YCbCr, CMY, etc. can also be used to represent the image. Similarly, a depth map is an array of values representing the distance of each point from the viewer. A depth map can be acquired in a number of ways. It can be captured directly from a scene using depth sensing cameras, it can be computed by exploiting correspondence between stereo image pairs and triangulation, or, in some cases, it can be derived from the image intensity. An approximation of depth of the scene can also be made by other techniques such as shape-from-motion.

Once the depth map has been estimated, the input image is used as a textured plane in 3D space, and the haptic device workspace is mapped to the size of the image in

xy -dimensions and to the number of different depth levels in the z -dimension. For example, in case of a 24-bit RGB image, it is first converted into a grayscale with $[0, 2^8 - 1]$ depth levels mapping to the 256 levels of the depth map. The resulting mapping assumes that the brightest pixels, i.e. with the brightness value of $2^8 - 1$, corresponds to the points closest to the viewer, and those with intensity value of 0 are the points located farthest from the viewer. Note that such a mapping is not necessarily accurate as the pixel brightness will depend on colour of the objects in the scene and the location of the light source relative to the viewer. As a result, the algorithm for haptic rendering of geometry does not work well for any arbitrary image but rather it is applicable to images having frontal illumination characteristics and Lambertian reflectance properties. Hence, the assumption of brightest pixels being closer to the viewer is an approximation which holds true in certain cases such as endoscopic images where the light source is collocated with the viewer position. The effect of this approximation on performance of the haptic rendering algorithm is discussed in Section 4.3.

Haptic feedback forces are calculated based on the relative position of the HIP and the mapped depth values. Collision detection is merely a test to check if the HIP lies above or below the height profile. The depth value at the pixel corresponding to the xy -coordinates of the HIP is compared against the z -coordinate of the HIP. For a haptic frame, if the z -coordinate of haptic interface point is $z_{HIP}(x, y)$ then the collisions are detected according to the following equation:

$$z_{HIP}(x, y) \geq d(x, y) \quad (4.1)$$

where $d(x, y)$ is the height profile of the scene. Once a collision is detected, the

magnitude of the feedback force applied to the haptic device is calculated according to:

$$\|\mathbf{F}\| = kp \quad (4.2)$$

where k is a spring constant that defines the stiffness of the object and the penetration depth p is calculated as the difference between the z -coordinate of the haptic interface point and the depth value of the pixel at (x, y) :

$$p = z_{HIP} - d(x, y) \quad (4.3)$$

The resulting force is proportional to the penetration into the depth map, i.e. greater penetration implies larger force magnitude. The feedback force is applied to the haptic device in the direction of the surface normal at the xy -coordinates of HIP, as shown in Fig. 4.1(c).

Figure 4.1 (a) A 2D image with brightness reflecting depth of the scene (b) corresponding heightmap (c) surface felt by the haptic device (d) content of the mask at the position indicated in (a).

<http://www.youtube.com/watch?v=8SMbvIdmuTk>

One case, where such dependence of force on penetration is not desirable, requires special attention for providing stable haptic interaction. Let us consider an object located close to the viewer location while the surrounding environment is relatively farther away.

In such a case, if the HIP moving in free space at the rear of the virtual environment penetrates into an object that lies much closer to the front of the scene, the depth of penetration is large and the direction of surface normal is outward from the object, i.e. opposite to the direction of motion. The resulting force calculated is thus also large. However, a sudden change from a zero force to a large force causes force kicking. To fix it, a threshold value p_{th} is used to clamp the penetration beyond which it is deemed to be constant. This reduces the magnitude of the sudden change in force. The smaller the value of the penetration threshold p_{th} , the smaller will be the sudden change in force magnitude. However, choosing a very small value for p_{th} would limit the hardness of the objects that can be perceived in the virtual environment. Moreover, since haptic devices differ in the maximum exertable force, so they are bound to produce different levels of stiffness. As a result the choice of threshold value p_{th} is device dependent. Our experiments with Geomagic Touch have shown that a p_{th} value of approximately 50 depth units produces acceptable force feedback.

The direction of force $\mathbf{F}$ is calculated by computing the cross product of two vectors $\vec{v}_1$ and $\vec{v}_2$ that share the current pixel (i.e. x, y coordinates at which HIP is located) as a common vertex, as shown in Fig. 4.2 (a).

Figure 4.2 (a) Cross product of two vectors sharing the current HIP gives direction of surface normal, (b) Calculation of surface normal by averaging a number of calculated surface normals.

The obtained depth map will always have some noise irrespective of the method used for depth acquisition. Since the haptic force is calculated based on depth data at a few neighbouring pixels, therefore the haptic geometry rendering algorithm is very sensitive to noise in the depth map. Even a small amount of noise in one of the depth values involved in surface normal calculation at any point results in unrealistic surface normal perturbations. Therefore, the surface normal $\vec{n}$ calculated in this manner is not sufficient for producing smooth forces. In order to counter the effects of noise, some smoothing needs to be performed on the depth values. A simple way of doing this is to use an average of a number of surface normals calculated based on depth values in a small neighbourhood of the HIP as shown in Fig. 4.2 (b). The 3×3 window, or a mask, represents a segment of the image plane. Depth values at these pixels are used for calculation of surface normals when the shaded pixel is the location of the HIP. A single pixel neighbourhood is shown in the figure. Larger neighbourhood sizes result in smoother surfaces, however using a too large neighbourhood window would also smooth out surface details. In order to avoid degenerate cases of surface normals, further modifications are required. For example, vectors $\vec{v}_1$ and $\vec{v}_2$ in Fig. 4.2 are made to have different magnitudes.

The complete algorithm for force direction calculation is as follows:

```

for  $i$ =rows of neighborhood window do
     $\vec{v}_1 = [x, y, depth(x,y)]$ 
     $\vec{v}_2 = [x + i, y + i, depth(x + i, y + i)]$ 
     $\vec{v}_3 = [x + i, y - i, depth(x + i, y - i)]$ 
     $\vec{v} = \vec{v}_2 - \vec{v}_1$ 
     $\vec{w} = \vec{v}_3 - \vec{v}_1$ 
     $\vec{n} = \vec{n} + crossproduct(\vec{w}, \vec{v})$ 
end for
for  $j$ =columns of neighborhood window do
     $\vec{v}_1 = [x, y, depth(x,y)]$ 

```

```

 $\vec{v}_2 = [x - j, y + j, \text{depth}(x - j, y + j)]$ 
 $\vec{v}_3 = [x + i, y + j, \text{depth}(x + j, y + j)]$ 
 $\vec{v} = \vec{v}_2 - \vec{v}_1$ 
 $\vec{w} = \vec{v}_3 - \vec{v}_1$ 
 $\vec{n} = \vec{n} + \text{crossproduct}(\vec{w}, \vec{v})$ 
end for
normalize  $\vec{n}$ 

```

In the above algorithm, x and y are the coordinates of the location of the current pixel, i.e. the xy -coordinates of the haptic interface point, $\vec{v}_1$, $\vec{v}_2$ and $\vec{v}_3$ are three-dimensional vectors. Function $\text{depth}(x,y)$ implies depth value at (x,y) location in the image. Function $\text{crossproduct}(\vec{a}, \vec{b})$ returns the cross product of $\vec{a}$ and $\vec{b}$. Further smoothing modifications can also be applied depending on the level of noise in the estimated depth.

4.1.1 Smooth transitions at object boundaries

The estimated depth map contains sharp discontinuities, such as those at the edges of the object that is much closer to the viewer than the rest of the environment. In such a case, as the haptic interface point tries to penetrate the object from any side other than the front, the magnitude of the force and the direction of the surface normal will change abruptly resulting in buzzing of the device and/or force kicking. This is because the calculation of the force direction depends on the current position of HIP only and it is independent of the force direction in previous haptic frames.

We have previously discussed the use of a threshold value p_{th} on penetration into the depth map to counter sudden changes in force magnitude. In order to avoid device buzzing, we also need to provide a mechanism to avoid abrupt changes in force direction at all points in the scene. We propose to include some history information of force direction calculated at previous frames to provide smooth transition of forces at object

boundaries. Let us consider Fig. 4.3, where a surface with a discontinuity in depth is shown as a bold step function along with directions of surface normals.

Figure 4.3 Surface normal adjustment at depth discontinuity (a) surface normal calculated without using previous normal (b) surface normal after the smoothing adjustment (c) angles between previous and current normals with and without using history information.

The vector $\vec{n}_0$ is the surface normal at the previous haptic frame. As the HIP moves along the direction marked in the figure (i.e. right to left), the direction of the feedback force changes abruptly (i.e. becomes vector $\vec{n}_1$). Our algorithm tests if the current force direction varies significantly from the previous frame. This can be done by computing the dot product of the currently calculated surface normal with the direction of force applied at the previous frame. If the angle between the current surface normal and force direction at the previous frame is greater than some threshold angle δ , then an average of the current normal and previous normals is used as the surface normal (vector $\vec{n}_2$) in the current frame. This means that the system needs to keep track of forces applied at previous frames. The smaller the acceptable change in force direction is, the smaller is the value of δ that can be chosen. However, a very small value will smooth out the details and edges or object contours will not be distinctly perceivable. On the other hand, if a very large value is chosen then the sudden changes in force direction will not be avoided and vibrations will be produced. Therefore, a trade-off is required to be able to haptically

perceive object boundaries without producing sudden changes in feedback forces. Our experiments have shown that a change in force direction greater than 25° is perceptible as a force discontinuity and causes vibrations. Therefore, the threshold angle δ is chosen less than or equal to this value.

Figure 4.4 Depth-based haptic geometry rendering workflow.

Force directions from a few previous frames are sufficient to provide believable force transition. Our experiments have shown that a value of 5-10 for h , i.e. using an average of 5-10 previous surface normals, is sufficient to provide stable feedback at object boundaries. A smaller value of h will cause the current surface normal to have a larger contribution in the average. As a result, the force vector for the current frame will still be significantly different from the force at the previous frame and hence cause vibration. On the other hand, if a very large value for h is chosen then the contribution of the current normal to the average will be very small and the change in force direction will be too slow as it will take too many frames until force direction changes to the new direction. Consequently, the edge or object contour will be smoothed out and will be perceived only as a tiny bump. In Fig. 4.3, an average of three previous normals and the current normal is used.

$$\delta_1 > \delta > \delta_2 \quad (4.4)$$

The workflow for the entire haptic geometry rendering algorithm is shown in Fig. 4.4. The depth and the colour image may have slightly different viewpoints in some cases such as when using depth sensing devices or some stereoscopic algorithms, hence the projection of depth image viewpoint to colour image viewpoint is required to align depth image pixels to the corresponding colour image pixels. Furthermore, $\vec{n}_t$ implies surface normal at frame t and h is the number of haptic frames used as force direction history. This algorithm contributes to the first research goal as it allows us to haptically explore the visible object's geometry.

Fig. 4.5 shows a colour image and the corresponding depth map obtained from a low cost depth sensor, Microsoft Kinect. The forces produced by the depth-based haptic

rendering algorithm are shown as arrows overlaid on both images. Notice the smooth transition of force direction at object boundaries.

Figure 4.5 Depth-based haptic rendering algorithm applied to depth stream obtained by Microsoft Kinect. Arrows show the force direction at different parts of the scene.

<http://www.youtube.com/watch?v=LwJjPuXUDhY>

When depth information about a dynamic scene is obtained in real-time, the depth-based haptic rendering algorithm can be used to perceive movement of objects in a scene. For example, in Fig. 4.6 the depth stream is captured in real-time using Microsoft Kinect. As the user hand moves upwards as if carrying the haptic cursor, the haptic device handle follows the cursor movement and moves upwards.

Figure 4.6 Haptic rendering of moving objects with depth stream obtained using Microsoft

Kinect. <http://www.youtube.com/watch?v=C2WeVLbYDbs>

4.1.2 Rendering haptic textures

The depth map obtained from an image or depth sensors may have low precision which may result in fine details on object surfaces not being incorporated into the depth map. Moreover, smoothing filters used by the depth-based haptic geometry rendering algorithm can also sometimes smooth out textures. In such cases, other image-based haptic texture rendering algorithms can be combined with the depth-based geometry rendering algorithm. The idea is to perturb the force direction calculated from depth values using forces calculated by the shading-based method. A simple scaling and addition of shading-based forces to the depth-based forces produces the desired perturbation. Several existing algorithms use the shading and illumination characteristics of an image to generate haptic texture forces. We use Vasudevan's algorithm (Vasudevan & Manivannan, 2008) which generates a force, for any point in the image, in the direction of the maximum gradient and proportional to the gradient magnitude. The force is thus directed from brighter to darker areas. The image is first converted to grayscale, and the image gradient is calculated using a two-dimensional Prewitt filter. Since common haptic devices have a single point of contact and we touch one point at a time, we need to calculate the gradient only in a small region around the haptic interface point. This allows the algorithm to work in real-time. The z -coordinate of the texture force is set to zero since the texture force vector is generated in xy -directions. The shading-based texture forces are scaled to a small value to ensure that the perturbation in the surface normal is bounded. The research goal of haptically exploring the visible object's texture is achieved in this way.

4.1.3 Rendering material properties of objects

In a virtual scene, different objects may be composed of different materials and thus are expected to feel haptically distinct. The stiffness of an object surface is defined by the spring constant for rendering the object geometry using depth information. However, there are other physical properties, such as friction, density, viscosity, etc., that can be haptically rendered to produce believable haptic perception. When using scenes composed of 3D models, each model can be defined as having specific physical properties. However, such a separation between objects is absent in a 2D image. Therefore, the image has to be segmented into regions of uniform physical properties. Moreover, material properties must be assigned to each region. The simplest way to do it is by assigning a different constant k in Eq. 4.2 for each segment, however, more complicated methods of assigning physical properties can also be employed, e.g., surface or inner properties defined by mathematical functions. Thus far, in this research an empirical assignment of k based on prior knowledge about the content of the scene has been employed. Accelerating or decelerating force components in the direction of motion of the HIP can be added to the force vector to simulate slippery or rough surfaces. This approach contributes to the first research goal as it allows us to haptically explore the visible object's physical properties including surface properties as well as inner properties.

The concept is illustrated in Fig. 4.7(a), which shows an outdoor scene. The image is manually segmented so that the region containing water is extracted as a segmentation map shown in Fig. 4.7(b). A very small stiffness constant is used for this segment to ensure that the water surface is not felt as a hard surface but rather can be easily

penetrated without much resistance. Moreover, when the HIP is in contact with the water segment, additional force components for flow direction represented by mathematical functions can be added to the final feedback force producing a realistic sensation. Although the rest of the image is not very suitable for haptic rendering because we seldom touch such a scene in real life, however, the example has been presented for illustration of the concept.

Figure 4.7 Haptic rendering of physical properties by assigning different properties to different regions, (a) an outdoor image, (b) manually created segmentation map.

4.1.4 Interactive scene modifications

Haptic interaction is a bidirectional process where a user exerts forces on different objects and received force feedback. Sometimes the scene is expected to change in response to the user actions. These changes may be in the form of deformations of objects or movement of objects from one position to another. In order to model these changes in image-driven haptic rendering, different methods can be used.

Firstly, the region of the image that will be modified is superimposed by small patches of the modified image. These are made to blend into the ‘background’ image. The superimposed patches can be overlaid in different ways, e.g. the superimposed patch can

be continuously displayed but modified only upon interaction, or the superimposed patch representing the modified region can be displayed only after interaction. Let us again consider the image in Fig. 4.7(a). The segmented water is considered to be the deformable part of the image and is therefore overlaid on the image. Fig. 4.8(a) shows the overlaid image as a highlighted region for illustration. When the HIP interacts with the scene, a procedural simulation of water flow is used to animate the overlaid water image. The result is an illusion that the static water has been disturbed by user interaction as shown in Fig. 4.8(b). Note that the background image remains static and the scene modification only happens in the desired region. This mechanism for modelling scene changes contributes to the second research goal.

Figure 4.8 (a) An image superimposed by a segmented image patch, (b) animation produced in the segmented patch by interaction using a haptic device.

Another method of modelling scene modification, particularly deformations in the scene, is by image warping in the vicinity of the interaction point. The idea is to divide the image plane into a regular grid and displace the grid points based on their distance from the point of interaction. The amount of displacement in each point can be defined mathematically to allow for larger displacement in points closer to the interaction point and smaller deformation in points farther away. The idea was illustrated in an

image-based virtual palpation system (Yasmin & Sourin, 2012) and a virtual venepuncture simulation (Xia & Sourin, 2012) developed as part of the externally funded project.

Figure 4.9 Deformation through (a) image warping in a virtual palpation system, (b) superimposition of the image with small texture patches.

Visual response of a scene to user manipulation can also be modelled using a hybrid approach of 3D models augmented with 2D images. The 3D models are collocated with the 2D image so that they overlap the corresponding objects in the image. These 3D models have associated material properties used for haptic rendering and can be interacted with as in fully three-dimensional virtual environments. They can be edited or deformed to simulate the visual changes in the scene when a user interacts with it. This approach of interaction with a hybrid scene is presented in the next chapter where it will be discussed in detail.

4.2 Special modes of the core algorithm

In the previous section, we have presented a novel depth-based haptic rendering algorithm and discussed how it can be combined with haptic rendering of textures and physical properties. While this approach produces believable perception of geometry, there may be images for which image intensity is not proportional to the height-map of the scene.

Therefore, it is applicable to a restricted set of images. Though we understand that it is not possible to come up with a single method applicable to all images universally for producing feedback forces for perception of visible geometry, texture, edges and material properties, we propose three special modes of the core algorithm that can make the image-driven rendering algorithm suitable for a broader range of image categories. These application modes with user adjustable parameters are processed separately. The user can switch between these modes based on the type of application and input image, and also tune the parameters best suited for the application. In the rest of this section, handling of these three important cases is described in detail.

4.2.1 Handling non-geometric intensity discontinuities

Since the colour characteristics of an image do not always represent the geometric features correctly, therefore a method to handle non-geometric colour discontinuity is proposed. Let us consider, for example, an image of a frontally illuminated face. A mole on the cheek or dark facial hair, such as eyebrows or moustache, will be interpreted by the algorithm as a discontinuity in depth. Such a discontinuity will be perceived as a deep hole or a cavity onto the face (Fig. 4.10).

Figure 4.10 A facial image with areas marked that will be interpreted as deep cavities.

This problem is solved by switching the haptic rendering algorithm to the ‘*offset*’ mode, where the maximum difference between neighbouring pixels at object boundaries is limited by a threshold value. For a colour change exceeding the threshold value, a false discontinuity is assumed and the HIP continues to move along the surface of the displayed object with the same depth as that for the previous HIP position. The relative changes of the next dominant colour will be used for subsequent depth calculations. This is illustrated in Fig. 4.11 where a checkerboard image is used. It shows two input images overlaid with the position and direction of movement of the HIP while the respective height-maps perceived by the haptic device are displayed below along with the gray level scale on the right. Fig. 4.11 (a) illustrates the perceived force feedback from the checkerboard pattern augmented with radial gradient if the default mode is used. In Fig. 4.11 (b) the same pattern is used in the offset mode. As the HIP moves in the offset mode, the first discontinuity from white to black being greater than the threshold is ignored. As it enters the block with a radial gradient, a second discontinuity, larger than the threshold value, is encountered. It is also ignored and the height-map shown is perceived.

Figure 4.11 Handling non-geometric intensity discontinuities (a) test image in normal (default) mode, (b) test image in the offset mode. http://www.youtube.com/watch?v=0OWBFvTNP_I

To this end, three masks of the same size are used. These are *current mask* for storing the depth values in the neighbourhood of the HIP at the current frame, *cached mask* for maintaining one frame depth mask memory, and the *force calculation mask* that is used to calculate the feedback forces. Each of them is centred at the HIP and follows the HIP movement in the xy -domain. At every haptic frame, each entry of the cached mask is tested against the corresponding entry of the current mask. For the entries having an absolute difference greater than the threshold, the corresponding entry of the force calculation mask is copied from the cached mask. The rest of the entries are copied from the current mask which is summarized as:

$$\begin{aligned} \delta_t^i &= d_t^i - d_{t-1}^i \\ \forall \quad i &\in \{1, 2, 3, \dots, \text{maskSize}^2\} \\ D_t^i &= \begin{cases} d_t^i & \delta_t^i < \delta \\ d_{t-1}^i & \delta_t^i > \delta \end{cases} \end{aligned} \quad (4.6)$$

where d_t^i , d_{t-1}^i and D_{t-1}^i are the i^{th} entries in the current mask, cached mask and the force calculation mask, respectively. At the end of the haptic frame, the force calculation mask is copied to the cached mask to be used in the next frame. This approach assumes a sudden transition from one dominant colour to another, hence a sharp discontinuity in depth is assumed. However, real images may not necessarily have such properties. Therefore, we propose to preprocess the image interactively or automatically so that a binary image map having sharp discontinuities is produced by stretching the contrast of the image. A *binary mask*, which is an additional sampling matrix, carries the values read from the binary map. For each entry of the binary mask having black values, the threshold value is added to the corresponding current mask entry. Therefore, any change in depth greater than the threshold is ignored while smaller changes remain perceivable. This

binary map may include only selected segments or the entire image. Fig. 4.12 shows a binary map with a few regions of the image along with the feedback force rendered with and without the offset mode. In the default mode the forces are directed as if the IKEA letters are carved deep inside the surface, while in the offset mode the feedback forces remain orthogonal to the surface.

Figure 4.12 Offset mode using a binary map. (a) 2D input image with blurry transitions, (b) the corresponding binary map with sharp discontinuities, (c) feedback forces in the default mode are directed as if the IKEA letters are carved deep inside the surface, (d) feedback forces in the offset mode are orthogonal to the surface

4.2.2 Emphasizing contours

Contours or object boundaries, being important features of a scene, also need to be emphasized. Unlike most of the existing approaches that detect and store edges in a scene, we further extend the previously described colour threshold case. It is used as a trigger for detecting the event that the sudden change of colours is a result of touching the visible contour of the object. The event of crossing the contour can be then defined as an intensity discontinuity larger than a predefined or calculated threshold value. We propose

to use mathematical functions to define the contour profile that has a variable width as a function of the speed of the HIP motion. A contour profile function f_{cpf} that rises exponentially to the maximum value, vanishes at a certain distance and ensures C^1 continuity is chosen such that $f_{\text{cpf}}(0) = 1$, $f_{\text{cpf}}(1) = 0$ and $f'_{\text{cpf}}(0) = f'_{\text{cpf}}(1) = 0$ (Levinski, 2007). This approach does not require generation of a contour map or image gradient but rather requires calculation of forces from the pixels surrounding the HIP. Once a contour is detected, a two-dimensional force exerted parallel to the image plane is calculated using the contour profile function and applied to the haptic device.

Figure 4.13 An edge with width Δ , an associated normal n and a function defined edge profile.

<http://www.youtube.com/watch?v=BCzVIW4c9Os>

For example, suppose the width of the contour is defined by parameter Δ within which the feedback force is calculated by some contour profile function, e.g., $f = 1 - r^6 + 3r^4 - 3r^2$ where r is a distance (Fig. 4.13). Once the HIP encounters the intensity discontinuity, a distance is calculated from the previous HIP position $\vec{p}_{t-1}^{HIP}$ to the current HIP position $\vec{p}_t^{HIP}$. The magnitude of the force $\vec{F}$ will be then calculated as:

$$\begin{aligned}
\vec{v} &= \vec{p}_t^{HIP} - \vec{p}_{t-1}^{HIP} \\
\delta &= \|\vec{v}\| \\
r &= \left(\delta/\Delta\right) - 1 \\
\|\vec{F}\| &= \frac{\Delta}{2}(1 - r^6 + 3r^4 - 3r^2)
\end{aligned} \tag{4.7}$$

As the HIP moves over an edge, the distance r increases from 0 to 1 and the force magnitude increases according to the contour profile function. As mentioned previously, the edge is defined to have a variable width proportional to the velocity of motion of the HIP. Therefore, the $\Delta/2$ term in Eq. 4.7 ensures that the edge produces a larger force if it is encountered by an HIP moving at a higher speed than if it is collided by a slow moving cursor. Besides the force profile for deriving the force magnitude, the force direction vector also has to be calculated. The normalized surface normal vector calculated using the core algorithm is used with one small modification. To link the direction of the force to the direction of motion when crossing the object contour, the angle between the vector $\vec{v}$ in the direction of HIP movement and the contour normal $\vec{n}$ is checked by calculating the dot product $\cos \theta = \vec{v} \cdot \vec{n}$. If it is negative, i.e. the angle θ is greater than 90° , then the force direction is opposite to the contour normal, otherwise it is in the direction of the contour normal. Since two-dimensional forces are applied to the haptic device for contour perception, the third coordinate of the force is set to 0. In order to ensure a stable force, the distance δ is calculated in haptic device coordinates rather than in image pixel coordinates. This allows for sampling the function at a higher precision to avoid sudden increase or decrease of the force magnitude and to ensure stability. Moreover, it is also important to consider the speed of motion of the haptic device handle. For example, if the HIP moves fast across the contour, the distance δ will exceed the contour width Δ very

quickly and the force would be applied for a very short duration. Therefore, the contour width Δ is made proportional to the speed of motion of the HIP: $\Delta_{min} + \|velocity\|$.

While this algorithm works for static contours, it requires some modification for moving objects. When the contour or an edge is detected and the force is applied as mentioned above, an *edge flag* is set. At the next frame in the haptic loop, if the edge flag is found to be set, the maximum force is applied as long as both the current HIP pixel and the *anchor pixel* (the pixel at $\vec{p}_{t-1}^{HIP}$ when the edge was found) have the same intensity values. This can happen either when the edge has moved over the current HIP pixel or it has moved over the anchor pixel, as shown in Fig. 4.14.

Figure 4.14 Movement of an edge (a) the edge is detected at $t=0$, (b) the edge moves from left to right, (c) the edge moves from right to left. <http://www.youtube.com/watch?v=zMReWQmTX30>

Here, the edge is detected at $t = 0$. At $t = 1$, the edge can move either to the position shown in (b) or to the position shown in (c). In case (b), since the current HIP pixel had a change in intensity after the last frame, we conclude that it has seen another edge. This means that the previous HIP position is not the anchor pixel any more. The new anchor position is the current HIP position with the previous colour. The current HIP position with the new colour therefore becomes the current pixel. In other words, the pixel colour of the HIP position has changed. Therefore, the positions of the current and

anchor pixel are the same, while their colours are different. Thus, the applied force becomes zero as the current intensity value differs from the new anchor position. In case (c), the maximum force will continue to push the HIP toward the edge until it crosses the edge where the edge flag will be unset.

4.2.3 Texture emphasis

The third special mode allows for emphasizing surface micro-details for images which have a relatively small change in geometry. Under the core algorithm, these details would be perceived as small bumps on the surface but the texture emphasis mode stretches them to the entire depth range. Histograms of such images are concentrated within a small range instead of the entire $[0, 2^8 - 1]$ range. In the texture emphasis mode, based on the histogram of the image the user interactively selects the region (i.e. minimum and maximum grey levels) of the histogram to be stretched. The histogram is stretched so that the pixels having an intensity value lying outside the selected range are clipped to the respective extreme of the selected range. Then, each entry i of the force calculation mask is obtained as:

$$D_t^i = \frac{2^8 (d_t^i - (2^8 - g_{max}))}{(g_{max} - g_{min})} \quad (4.8)$$

where D_t^i is the i^{th} entry of the force calculation mask, d_t^i is the i^{th} entry of the current mask, and g_{max} and g_{min} are the values selected by the user as histogram range values. Recalling the mapping convention about location of the brightest parts and darkest parts of the scene, each mask entry is inverted, i.e. subtracted from 2^8 (hence the term $2^8 - g_{max}$ is used in Eq. 4.8). Collision detection and haptic rendering will then be performed as discussed previously. The effect of texture emphasis in this way is

illustrated in Fig. 4.15 where a 2D image with relatively small changes in geometry is shown along with the perceived surface before and after texture was emphasized. It can be seen from the histogram of the test image that it does not cover the entire range of gray level values and a significant portion of histogram is ‘wasted’ in a sense that those depth values do not occur in the scene. Therefore, the texture can be emphasized by stretching the histogram to the unused depth values.

Figure 4.15 Texture emphasis mode: (a) test image (b) the histogram generated by the developed application (c) the perceived surface before and after texture emphasis.

http://www.youtube.com/watch?v=EWSn7qn_FHg

4.3 Analysis and discussion

At every frame the algorithms operating on a tangible image run independently from each other and the force computed for each of them is combined to generate the feedback force applied to the haptic device. The movement of the device by the user can also modify the scene. These changes will be perceivable in the next frame and the modified frame will be used to generate force for that frame. As a result, objects deformed in one frame will be

perceived to have the deformed shape in the next frame. Any abrupt change in force will be compensated by the smoothing adjustments described in Section 4.1.1. Therefore, stable force feedback is guaranteed. The relationship between different algorithms is depicted in Fig. 4.16.

Figure 4.16 Flow chart showing relationship of different parts of the tangible images approach.

While working with a few pilot applications we have identified cases which require further research and development. There are images for which the proposed algorithm will produce unrealistic results. This is due to how human vision and touch works in real life. For example, one interesting case is the bas-relief ambiguity. Let us consider a concave hemisphere lit by frontal illumination by a source at infinity. The inside of a concave hemisphere may appear to be the outside of a convex hemisphere. Such visual illusions occur in real life too and sometimes there is no way to tell the geometric structure or physical properties without manipulating the object. In such a case, other sources of depth information can be used to get a reference point of depth on the object surface, e.g., by using multiple views of the scene or depth sensors such as Microsoft Kinect. Such reference to the ground truth of scene depth can help reliably handle

confusing situations such as the bas-relief ambiguity. However, sometimes even the use of depth sensors may not be feasible e.g. in minimally invasive surgery.

We have mentioned a few methods of interactive modifications of the scene. However, there can also be other methods developed for this purpose such as pixel level editing of the image from neighbourhood of the interaction point, or seamless transition between layers of images such as those used in Google Street View. These methods need further research. We also propose the augmentation of the 2D scene with editable 3D models in the next chapter.

In order to make different objects feel distinct, each object or region in the image must be made to feel different. We proposed segmenting the image – manually or automatically – and assigning each region or segment a set of physical properties. Automatic assignment of physical properties to each region is a challenging task. Developing algorithms to automatically detect and/or assign different physical properties to different segments of an image is a subject of future research. An alternative way of haptic rendering of different physical properties to use additional 3D objects augmented to the image. While these 3D objects can be used as editable parts of the scene, they can also have distinct predefined physical properties which can be rendered upon contact with them as in 3D haptic rendering.

4.4 Summary

In this chapter, we presented a novel algorithm based on obtained depth information about a scene. It can generate reliable feedback forces when a sufficient approximation of the depth map is available. We also presented the usage of depth-based rendering

algorithm in combination with rendering of fine surface details and physical properties. We then identified and proposed solutions to three important special cases to increase the range of different image categories that can be made tangible using the proposed algorithm. These application modes provide tuneable parameters that can be adjusted by the user based on the application at hand. We also discussed that a single algorithm cannot be used to make every image tangible reliably and there may be cases such as the bas-relief ambiguity where the proposed algorithm would not be able to produce believable haptic perception. We proposed the superimposition of small texture patches onto the image and image warping to simulate interactive scene modification.

Chapter 5 Hybrid Approach – Augmenting 2D Images with 3D Object Models

In the previous chapter we discussed modelling of interactive scene changes using superposition of 2D texture patches and image warping. We also mentioned that a hybrid scene composed of 2D images and 3D object models can be used to model scene changes in response to user manipulation. Changes in the geometry of these objects, such as deformation, can be modelled as in haptic interaction with fully 3D virtual environments. Since a clear separation exists between the way the image and these models are defined, these models can additionally have physical properties associated with them. When interacting with the 3D objects these physical properties are rendered along with the geometric forces. These objects can be composed of polygonal meshes or have their geometry and physical properties defined by mathematical functions. The augmented 3D models can also serve the purpose of simulating invisible physical effects such as wind. In this case, they act as invisible haptic containers collocated with the objects in the image and having predefined physical properties.

In this chapter we present algorithms for haptic rendering of hybrid scenes composed of 2D images augmented with 3D objects. These objects can be visible editable models located such that they overlap the objects in the image they correspond to. Alternately, these can be mathematically defined invisible haptic objects with no visible appearance simulating different force field effects.

5.1 2D images and augmented editable 3D models

Not all types of interactive modifications in a scene can be modeled using the methods described in the previous chapter such as pixel level editing or superimposition of texture patches. In some cases, 3D models can be used for modelling these scene modifications together with the 2D image. We, therefore, propose mechanisms for haptic interaction with a hybrid scene composed of 3D models augmented to 2D images as editable objects with associated physical properties. The 3D models are placed in the 3D environment so that they merge seamlessly into the image background. The resulting hybrid scene can be haptically rendered in two ways. Firstly, 3D haptic rendering algorithms can be combined with the depth-based haptic rendering algorithm presented in Chapter 4. On the other hand, a depth map of the whole hybrid scene can be obtained and the depth-based haptic rendering algorithm can be used to compute force feedback.

5.1.1 3D haptic rendering with depth-based interaction

The augmented 3D objects can be composed of polygonal meshes or can be defined by implicit functions. The 3D objects make the foreground of the scene and blend into the 2D image background located at the far plane of the orthographically projected space. The HIP interacts with the 3D foreground objects using existing algorithms, such as the god-object algorithm, for haptic rendering of fully three dimensional virtual environments. This 3D interaction continues until the HIP penetrates the depth map of the 2D image background. When this event happens, the force calculation is switched to the depth-based haptic rendering algorithm described in the previous chapter. Using a 3D model augmented to the image allows for deformations to be modeled more realistically

than image warping.

As an example, let us consider an image-driven minimally invasive surgery simulation where certain types of tissues need to be either manipulated to diagnose any problems or to be partially removed. Fig. 5.1 shows instances from a real arthroscopic surgery where a meniscus tissue is probed (a), (b) and a part of it is removed (c). When only a two-dimensional image is used to represent the joint instead of a full 3D model, any interactive modifications require the image content to be edited, which is a challenging task. On the other hand, the augmented 3D objects, which can be used to model deformable structures (e.g., menisci in the knee joint), can be used for providing interactive modifications to the scene.

Figure 5.1 Interaction with soft tissues in arthroscopic surgery (a,b) probing of meniscus, (c) removal of damaged part of meniscus

A tangible image can be created by augmenting a 3D model of the meniscus tissue to an arthroscopic image of the knee joint. Since bones are solid structures and do not undergo any deformations, 3D models of only soft tissues need to be modeled. Fig. 5.2 shows a 3D meniscus, defined by a polygonal mesh created from MRI data, augmented to the arthroscopic image. Using this approach, various unhealthy meniscus models can be generated from different MRI data in this way and augmented to the 2D image to create several different surgical scenarios. Existing physics engines and libraries can then be

used to model deformation in the augmented 3D model of the tissue. Efficient visual and haptic models of human tissues for virtual simulation of minimally invasive surgery are under development as part of the externally funded project. These models will be used to replace the polygonal model in the tangible arthroscopic image for simulation of arthroscopic procedures.

Figure 5.2 (a) 3D polygonal model of meniscus augmented to an arthroscopic image (b) a grasper tool operating on the 3D meniscus

The 3D models augmented to 2D images can have different physical properties associated with them. If an image-based scene representation is used alone, then the image has to be segmented into regions of uniform physical properties. With a 3D object augmented to the image, the separation between the object and the image is already there so the need for image segmentation is avoided. This approach also contributes towards achieving the first research goal, i.e. exploration of physical properties of visible objects.

5.1.2 Depth-based interaction with the hybrid scene

The second method of interacting with hybrid scenes composed of a combination of visible 3D models and 2D images works by obtaining the depth map of the entire scene. A slightly modified haptic rendering algorithm is then applied to the hybrid scene.

Firstly, the depth information for the hybrid scene needs to be obtained. Since the scene is comprised of two parts – a 2D image and the augmented 3D objects – the depth data has to be calculated for both of them. For the depth reconstruction from the image, a correspondence between the image intensity and the scene depth is established as explained in the previous chapter. Another depth map is constructed by first visually rendering the 3D models and then reading the depth values from the z -buffer at each xy -coordinate pair. In this second depth map, for xy -coordinates that lie outside the 3D object, the depth map then contains the z -coordinate of the panoramic image plane.

Figure 5.3 Mask filling algorithm for hybrid scene composed of 3D model and 2D image.

The force calculation mask for the haptic geometry rendering algorithm is filled from the two depth maps based on the location of the haptic device in the scene. If the mask lies completely on the image, then it is filled with values from the image depth map as in case ‘A’ in Fig. 5.3, which shows the frontal view of a hybrid scene. When the mask lies fully on the 3D object, i.e. case ‘C’, z -buffer values are copied to the mask. When the

mask lies on the boundary of the 3D object, i.e. it lies partially on the image (case ‘B’), then depth values from both maps are combined to fill the mask. This is done by checking for all pixels in the small neighborhood around the HIP whether the value in the z -buffer is less than the z -coordinate of the panoramic image plane. If so, the z -buffer value is chosen to fill the corresponding mask entry, otherwise the mask value is obtained from the image depth map. This is formulated below where $depth_{buffer}$ represents the depth map corresponding to the 3D objects, $depth_{image}$ is the depth map extracted from image intensities and z_{image} is the z -coordinate at which the panoramic image plane is located in the 3D space.

```

 $a = (\text{size of neighborhood window}) / 2$ 
for  $i = -a$  to  $a$  do
    for  $j = -a$  to  $a$  do
        if  $depth_{buffer}(x + i, y + j) < z_{image}$ 
             $mask(i + a, j + a) = depth_{buffer}(x + i, y + j)$ 
        else
             $mask(i + a, j + a) = depth_{image}(x + i, y + j)$ 
        end if
    end for
end for

```

The mask is then used to compute forces to be rendered as described in the previous chapter.

As mentioned earlier, the augmented 3D objects can be represented by polygon meshes or mathematical definitions. When using, for example, blobby objects represented by implicit functions, interactive object removal procedures can be simulated. Set theoretic operations can be used to modify the implicitly defined blobby surfaces as it was previously done in, for example, (Qi & Sourin, 2006a; Qi & Sourin, 2006b; Wei et al., 2008) in case of shared virtual environments on the Web. We modeled the meniscus in the knee joint by using solid objects defined by implicit functions as inverse coordinate distance squares, as

shown in Fig. 5.4 (a), where a number of the functions are combined to form the resultant function. The entire meniscus needs not to be modeled, but rather only its limited part visible and haptically accessible in the virtual surgical field. In Fig. 5.4(b) the part of the meniscus that is modeled with the function in Eq. 5.1 is highlighted.

$$f(x, y, z) = \sum_{i=1}^n \frac{a_i}{b_i(x - x_i)^2 + c_i(y - y_i)^2 + d_i(z - z_i)^2} - \delta \geq 0 \quad (5.1)$$

where n is the number of blobs, a_i , b_i , c_i , d_i are parameters associated with i^{th} blob centered at coordinates (x_i, y_i, z_i) . Different views of the modeled part of the meniscus are then interactively modified by performing set theoretic operations to simulate surgical tasks, such as cutting away damaged pieces of the meniscus (Fig. 5.4 (c)). The 3D objects are visually rendered in real-time to update the surgical view. This approach achieves the research goal of modelling interactive scene changes in response to the user manipulation.

Figure 5.4 Meniscus modeling using implicit functions. (a) Summation of inverse square functions, (b) Relation of the modeled part of the meniscus to the complete meniscus along with different views of the modeled part. (c) Meniscus removal using set theoretic operations.

<http://www.youtube.com/watch?v=A0EYKC1yb8Q>

The proposed method of haptic interaction with a hybrid scene requires the hybrid scene to be manually authored. Manually adjusting the position and orientation of the 3D object in space, so that it matches exactly the object in the image it corresponds to, is a challenging task. Moreover, making the 3D object blend into the 2D image background poses further challenges. Therefore, automatic matching algorithms need to be established that can match the object pose to the image and produce a hybrid scene where the intersection between the image and 3D objects is visually realistic.

5.2 Invisible haptic containers

In 3D cyberworlds, as well as in various interactive and simulation environments, it is often crucial to make photo-realistic appearance of the virtual objects and provide haptic immersion into the scene where the objects will act like real objects rather than images of the objects. However, the price paid for visual realism often ends up in megabytes of data and non-interactive rendering rates. We have thus far presented novel algorithms for image-driven haptic rendering. Here we propose that in many cases real images of the objects can be successfully used in virtual environments if they are augmented with haptic effects defined as invisible tangible geometric shapes with transparent appearance and various haptic properties associated with them (Fig 5.5).

When mathematical functions and procedures are used for defining geometry of such shapes, as well as the physical properties associated with them, as in (Sourin et al., 2009; Sourin & Wei, 2009; Wei et al., 2009), the size of these models can be significantly reduced while the precision of representation can be provided with any required level of detail. Such simple shapes can be easily rendered for collision detection with haptic

devices compared to the collision with the geometry having photo-realistic level of detail. When positioned and oriented consistently, the image and the associated haptic objects produce an adequate visual and haptic immersion.

Figure 5.5 Illustration of using invisible haptic containers (a) a 3D sphere, (b) an image of the sphere on a plane and a surrounding invisible 3D sphere, (c) a front view on the image and the matching geometric sphere, (d) Superposition of the simplified 3D geometry with an actual image.

The haptic objects can either have their geometry based on that of the objects in the scene or they can be simplified geometric containers for force fields representing the region in which a force can be felt. Often merely exploring the geometrical structure may not provide sufficient information about the object. So, different kinds of physical effects can be defined for the haptic objects, e.g., using function representations. These include (1) surface properties such as elasticity, static and dynamic friction, (2) internal properties such as viscosity, density etc., and (3) force fields such as winds, flows, gravity.

Several methods of authoring such tangible images can be used. 3D models of objects

can be constructed separately and manually associated with images at appropriate position and orientation. Alternately the use of 3D reconstruction techniques for automatic extraction of geometry from the original image can be explored, e.g., using the robotic vision technique of 3D model selection from an internet database (Klank et al., 2009) or using the recently proposed 3-Sweep algorithm (Chen et al., 2013).

Vision has the highest bandwidth (i.e. the capacity to carry information) among the primary senses followed by the sense of hearing. Touch, on the other hand, has an even lower bandwidth, around one fourth of that of vision (Way & Barner, 1997). Due to this fact we can visually identify an object, say a car, even in a blurry image, but an approximate model of an object in a scene cannot be identified using the sense of touch alone. However, when touch is used together with vision, it becomes a complementary sense. Therefore, the invisible haptic containers need not be very detailed accurate 3D models of the objects in the image rather lower resolution approximations can be used without loss of immersion. For example, consider Fig. 5.6, which shows a photo of the Stanford Bunny. The original polygon mesh reconstructed from the Stanford Bunny volume data is quite large ~3.2 MB and nearly 70,000 triangles. Therefore we used a simplified 200 KB polygon model which was sufficient to provide an adequate haptic sensation. This transparent 3D model is then co-located with the photo image to make it appear tangible when looking at the image. The position and orientation of this haptic object is adjusted to match the orientation of the bunny in the image. When looking at the image and using a haptic device, the user feels the surface of the invisible 3D model and it appears as the user touches the surface of what is displayed on the screen. The surface properties are defined for such a haptic object while internal properties are omitted.

Figure 5.6 A photo of Stanford Bunny and its 3D VRML model used to augment the image.

The idea of making images tangible by augmenting invisible haptic containers can also be extended to haptic interaction with videos. Transparent shapes representing physical properties of objects in the scene can be made to follow movement of objects within the scene. The use of computer vision techniques like object tracking can be explored to reduce the complexity of the augmentation process. As an example, consider Fig. 5.7, where a moving helical force field is placed exactly behind the plane on which a video of a moving whirlpool is projected. The helical force field is defined inside a transparent cylindrical container, the centre of which follows the centre of the whirlpool. In this example, a VRML position interpolator has been used to make the haptic container movement follow the whirlpool in the video. The cylindrical container is also rotated to make it oriented such that the surface of the whirlpool is parallel to the top of the container. The force field itself is defined by parametric functions. The invisible cylinder is defined by implicit functions. No additional polygon-based objects are used in this scene. Thus, the third research goal of haptically exploring physical effects that have no visual appearance is achieved.

Figure 5.7 (a) A snapshot from a whirlpool video which is superimposed by the container in (b) at appropriate orientation, and the clockwise force effect on the haptic device, (b) a helical force field inside a transparent cylindrical container. <http://www.youtube.com/watch?v=GUvWH8F1KAw>

5.3 Summary

In this chapter, we have proposed haptic interaction with a hybrid scene composed of 3D object models and 2D images. The 3D part of the scene can comprise editable visible 3D models with associated physical properties for representing modifiable parts of the scene. Interaction with such a scene is either based on conversion of the entire scene into a composite depth map or using a combination of 3D haptic rendering algorithms with the depth-based image-driven haptic rendering algorithm. Interactive modifications to the scene can be reliably simulated with them. Alternatively, the 3D part can be made up of invisible 3D models representing objects or regions of force effects. The 3D models, whether editable or invisible, can be defined by mathematical function definitions for smaller size and application of set theoretic operations for editing them. We have used manual authoring of such hybrid scenes. Automatic adjustment of the pose of 3D objects is a subject of future research.

Chapter 6 Constraint-Based Interaction with 2D Images

Different haptic devices are available to the research community. They have different capabilities and different limitations. During the course of the project we have come across scenarios where the haptic devices limit the amount of realism that can be achieved in haptic interaction in general and image-driven haptic rendering in particular. In this chapter, we propose methods to achieve realism either by creating constraints or by eliminating them. The created constraints may be physical or virtual and the eliminated constraints are software solutions to relax the inherent constraints in haptic devices.

6.1 Multiple contact point interaction

As mentioned previously, haptic interaction provides an increased sense of presence in 3D virtual environments by using various force feedback devices. Usually, the most common desktop devices with 3 or 6 DOFs have a single haptic interaction point. However, the real interaction tool is not always a point but often is a rather complex geometrical entity, therefore multiple contacts between the scene elements and the avatar must be modeled. While tactile feedback devices provide multipoint interaction, a few kinesthetic feedback devices that support such interaction are also commercially available. These include the 7-DOF *sigma.7* and 4-DOF *omega.7* devices from Force Dimension (Force-Dimension, 2013) and the Freedom 7S from MPB Technologies (MPB-Technologies-Inc., 2013). These devices have an active grasping degree of freedom and deliver feedback forces at two different points. Typically, multipoint haptic

interaction is used in scenarios where tasks, such as grasping, need to be simulated (Ferre et al., 2012). Multipoint haptic interaction is also very important in applications where the virtual haptic tool is a complex 3D object rather than merely a point. Since single contact point devices do not adequately represent such objects, the haptic interaction produced is not realistic as it does not prohibit unrealistic scenarios such as passing through small holes with a large object. Moreover, parts of the virtual haptic tool that are away from the actual HIP may penetrate the objects in the scene.

A few approaches exist that tend to avoid such unrealistic scenarios in simulating multipoint interaction with single contact point devices. Most of these techniques use penalty-based methods for force calculation but differ in the type of scene representation, method of collision detection, or positions of contact points relative to the HIP. For example, a volumetric scene representation is used in (Osorio, 2008; Petersik et al., 2003). In the former, a set of contact points at the center of a sphere and the axis of the cylindrical cursor are tested for collision detection while in the latter contact points are distributed on the surface of the spherical cursor. The force calculated at each contact point is always directed towards the HIP. These algorithms are computationally expensive as a large number of surface points are often needed to be checked for collision and sometimes a ray casting algorithm is used at sub-voxel resolution to calculate the surface of the object in collision. A haptic tool composed of line segments from the HIP to points at a spherical distance from the HIP was used in (Srivastava, 2007) to interact with a scene represented by polygons. It attempts to simulate contact with finger tips for virtual palpation by incorporating the algorithm into the Virtual Haptic Back project. It uses hashing and spatial decomposition to speed up the algorithm. In (Vembar et al., 2010), a

physics-based model of a deformable haptic tool is used. The tool is composed of serially attached virtual nodes with contact points defined at their centers. Each node center is tested for collisions, and a linear combination of the forces for each point is fed back to the device. Penetration of the tool into objects in contact cannot be avoided as the contact points are located inside the haptic tool rather than at the surface. As opposed to these penalty-based methods that compute forces based on penetration depth, the work in (Ruspini & Khatib, 2000) presents a multi-contact model, where constraints are applied to the motion of a virtual object when a collision is detected between the tool and the polygonal scene. Singapogu et al. (Singapogu et al., 2008), use a 5 DOF device with the haptic interface point at the center of a wand with its end points used for collision detection and force calculation. A ‘points generation subsystem’ updates the locations of the end points based on HIP location. Although it also offers torque feedback about two angles, but a pair of points is insufficient to simulate accurate tool-scene interaction.

We propose a novel multi-point interaction extension to the image-driven haptic rendering algorithm described in Chapter 4. Since collision detection for each of the contact points is a simple test based only on its current position, therefore the computational complexity of the algorithm is low. Moreover, note that the multiple point of contact extension to the depth-based haptic rendering algorithm proposed here is not an attempt to generate 6 DOF force feedback. Instead, the proposed extension makes the interaction experience more realistic by approximating the graphical avatar to a set of points as opposed to a single haptic interface point.

A number of *haptic contact points* (HCPs) are defined by an offset value relative to the HIP. These points are located at salient features of the haptic cursor such as corners

and edges. These HCPs can be simply distributed such that they are equidistant from the HIP in all directions. A composite depth mask is associated with each of them and filled from the two depth maps in case of a hybrid scene. Collision detection and force computation for each of these points takes place exactly in the same manner as for a single contact point described in the Section 4.1. The resulting forces are averaged to form the force vector applied to the haptic device at the actual HIP. The direction of the force vector is thus the resultant of these forces.

In other words, if the total number of HCPs is n while c HCPs are in collision at a given haptic frame, then the resultant force is given by:

$$\mathbf{F} = \frac{1}{c} \sum_{i=1}^n \mathbf{F}_i \quad (6.1)$$

where $\mathbf{F}_i$ is the force generated at the i^{th} HCP. As an example, let us consider Fig. 6.1, where a cubic haptic cursor with the HIP at its center is shown. There are 8 HCPs located at each of the corners. The cubic cursor interacts with two planes placed such that one end of each lies close to each other in 3D space. If a single contact point is used, the HIP can pass through the gap between the far ends of the planes. However, with a cubic avatar the haptic cursor does not pass through the opening if its size is greater than the gap.

Fig. 6.1 (a) shows one HCP in collision with one plane. The resultant force at HIP equals the force at the colliding HCP since forces for all other HCPs are zero. In Fig. 6.1 (b), two of the eight HCPs collide with the same plane and an average of the forces is applied to the HIP. In case 3 shown in Fig. 6.1 (c), two HCPs are in collision, one with each plane, and the resultant force exerts in a direction that prohibits movement in the negative z -direction, i.e. away from the user.

Figure 6.1 A cubic haptic cursor with eight haptic contact points (HCPs) interacting with two planes. The feedback force is applied at the HIP which lies at the center of the cubic avatar. (a) A single HCP is in contact with one of the planes. (b) Two HCPs in contact with one plane. (c) Two HCPs in collision with different planes.

Fig. 6.2 shows the 2D illustration of the concept with forces at each HCP and HIP indicated. The HCPs can also be arranged in other constellations based on the geometry of the haptic cursor and they need not necessarily be equidistant from the HIP while the force summation equation (Eq. 6.1) can be modified accordingly for each constellation, e.g., each force component may be scaled based on the distance of the respective HCP from the HIP.

Figure 6.2 2D representation of the cubic haptic cursor with HCPs at its corners interacting with two closely placed planes.

One of the key features of using multi-point interaction is to avoid inter-penetration between the haptic cursor and the virtual scene. Since contact points are distributed at salient features of the cursor, as in Fig. 6.1, such penetrations are avoided. There can still be cases of interpenetration of the tool and the scene at the edges of the cube. Defining more points on the edges will avoid such penetrations. The higher the number of points, the lesser is the possibility of interpenetrations. A comparison of using multi-point interaction with single-point interaction is shown in Fig. 6.3. It can be seen that in the case of multipoint interaction the haptic tool is unable to penetrate into the planes while in case of single point interaction, where the HIP lies at the center of the cube, a part of the cube has already penetrated when the HIP collides with the plane. Hence, this method fulfills the fourth research goal of providing for interaction with multiple contact points.

Figure 6.3 Tool-scene penetration. A comparison between (a) single point interaction and (b) multi-point interaction.

6.2 Dynamic haptic workspace mapping

Another limitation that the desktop haptic devices have is their limited workspace. It is sometimes smaller than the scene depicted in two dimensional images. To illustrate this limitation and its proposed solution let us consider the case of the z -dimension.

A linear mapping of real world coordinates to haptic workspace coordinates in the z -direction would result in loss of detail. This is due to the fact that a single unit movement of the haptic device results in a much larger movement of the haptic cursor in the virtual environment. If the movement in the scene due to a single unit movement of the haptic device is larger than the resolution of the depth map, then some scene detail will be lost. For example, Geomagic Touch devices have a usable workspace that is 160mm wide, 120 mm high and 70mm deep, approximately. The resolution of this device is about 450 dpi or 0.055mm, approximately (3DSystems, 2013). Therefore, the number of distinct values that can be read from the device along the z -direction is around $70/0.055 \sim 1275$. If the resolution of the obtained depth map is greater than this value and the haptic workspace is linearly mapped onto the obtained depth map, then a single unit movement of the haptic device along the z -direction will result in more than one unit movement of the haptic cursor with reference to the depth map. For example, Microsoft Kinect produces an 11-bit depth map which is converted to a 16-bit floating point value by the software depth generator of OpenNI library. Therefore, $2^{11} = 2048$ different levels of depth are generated (Ryden et al., 2011). Thus, a linear mapping of usable workspace of Geomagic Touch device having ~ 1275 levels in z -dimension onto 2048 levels of depth obtained from Microsoft Kinect will result in a loss of detail about a scene. Certain other haptic devices may have a higher z -resolution and a larger workspace. Similarly, other depth sensors may be able to retrieve depth of a scene at a more precise and higher resolution. Nevertheless, in order to achieve better immersion, a solution to this limitation must be proposed to provide for haptic perception of as much detail of the scene as possible.

We propose the dynamic mapping of haptic workspace to the virtual environment. Let us consider a fixed sized virtual environment larger in size than the haptic workspace, and the colour image displayed as a texture on the far plane of an orthographically projected virtual environment. The haptic workspace is initially mapped to the entire virtual environment. This mapping is made adjustable so that the near and far planes of the haptic workspace can be moved within the virtual environment in real-time. This allows the user to sense objects only in a part of the haptic virtual environment resulting in haptic zoom effect. The z -coordinate of the haptic cursor z_c is computed based on the z -coordinate of the haptic interface point z_{HIP} according to the following equation:

$$z_c = z_{HIP} \left(\frac{W_f - W_n}{r_z} \right) + W_n \quad (6.2)$$

where W_n and W_f are the near and far planes of the haptic workspace respectively and r_z is the size of the depth range the workspace is to be mapped to.

Fig. 6.4 illustrates the idea of dynamic workspace mapping for the z -dimension. The dashed black lines indicate top view of the virtual environment which has four objects, and the shaded area in the dotted red lines shows the haptic workspace. Fig. 6.4 (a) shows the entire virtual environment mapped onto the haptic workspace. All four objects can be touched using the haptic device in this case. In Fig. 6.4 (b) the near plane of the haptic workspace is shifted back so that only rear part of the scene can be haptically explored. As the green cube **A** lies outside the haptic workspace so it cannot be touched. Similarly, Fig. 6.4 (c) shows the far plane of the haptic workspace shifted forward which implies that the blue cylindrical object **B** cannot be felt haptically. In Fig. 6.4 (d) the near plane of the haptic workspace is shifted back, and also the far plane shifted forward allowing a

smaller region of interest to be explored using the haptic device. In this case the region of interest includes the blue rectangle *C* and lower part of the triangle *D*.

Figure 6.4 Dynamic haptic workspace mapping. Top view of a virtual environment is shown with depth values increasing from bottom to top. Near and far planes of the scene are marked.

Fig. 6.5 shows the mapping of the haptic workspace to a small region of interest in the z -dimension of the scene when using Microsoft Kinect. The haptic workspace is mapped to the nearer flower pot in the first part and the farther pot in the second part of the figure. This allows haptic sensation of fine details otherwise imperceptible due to loss of depth detail when mapping the entire scene to the haptic workspace. The near and far planes of haptic workspace can be adjusted (moved forward or backward) in real time by pressing arrow keys on the keyboard.

Figure 6.5 Haptic workspace is mapped to different regions of interest to allow for exploration of fine details.

As another example, let us consider a 16-bit grayscale linear image shown in Fig. 6.6(a). A 16-bit image implies the possibility of having $2^{16} = 65536$ different depth values. For such an image with a high resolution the depth values increase from 0 on the left towards 65536 on the right with single unit increments. An inclined plane should be perceived haptically when the haptic cursor is moved from left to right along the path shown by the arrow in Fig. 6.6(a). Since the Phantom Touch haptic device has a resolution of 0.055mm and a maximum workspace range of $[-130, 85]$ in the z dimension, the number of discrete z-coordinate values of the device that can be read are $215/0.055 \sim 3910$. In order to illustrate the effect of dynamic haptic workspace mapping, consider a linear mapping between 3910 values of the device workspace and the entire 65536 depth range, and the haptic device being moved along the arrow in Fig. 6.6(a). Fig. 6.6(b) shows the surface felt by the haptic device, i.e. the red line, and the actual surface, i.e. the blue line, for the first 100 pixels. Notice that the surface is discretized and the haptic device is not able to feel a smooth feedback. Using dynamic haptic workspace mapping, the near plane can be moved farther from the viewer to map the device workspace onto a subset of the total possible depth values such as in the range $[0, 5000]$. Fig. 6.6(c) shows that the same pixels are perceived much smoother and closely matching to the actual surface as compared to when the device workspace was mapped to the entire depth range.

(a)

(b)

(c)

Figure 6.6 Dynamic mapping of haptic workspace onto a 16-bit grayscale image.

6.3 Alteration of DOF using physical constraints

As discussed in Section 2.1, single contact-point desktop haptic devices have both input and output degrees of freedom (DOF). The number of DOF that a user can manipulate a device is referred to as the input DOF whereas the number of DOF in which the haptic device provides feedback to the user is indicated by output DOF. The number of input and output DOF in any desktop haptic device need not necessarily be equal. The low-end desktop haptic devices usually have 3 DOF input (i.e. translations in x , y , z directions) and 3 DOF output in the form of a single 3D force vector. Then there are devices which provide more input flexibility by allowing roll, pitch and yaw movements in addition to the translation motions. These are 6 DOF input and 3 DOF output devices. The high-end desktop haptic devices are capable of providing 6 DOF output in the form of a 3D force vector and a 3D torque vector along with 6 DOF user input. Based on the application requirements, these capabilities may or may not be sufficient for simulation of haptic interaction tasks. For example, a desktop haptic device with a stylus such as a Geomagic Touch is not suitable for a needle insertion simulation in its original form. Instead, solutions such as the Bangor needle holder modification (Coles et al., 2011a) are used to make the device suitable for such applications. In our research, we also came across an application – minimally invasive surgery simulation – where the performance of said devices was restricted due to a limited number of DOF or rather the required input DOFs were not the three translations and rotations. Instead the application required a different set of DOF.

We have modified the Geomagic Touch devices so that the input DOFs offered by them match the requirements of the application. We have extended the handles of the

device stylus using actual surgical instruments. Fig. 6.7 (a) shows the extension created using the plastic hub of a Vacuette blood sampling equipment and reusable arthroscopic shaver blades. Its connection to the Geomagic Touch device is displayed in Fig. 6.7 (b). In order to match the use of the modified device to the operating room (OR), a thin blue plastic sheet is used to cover the table base on which the device lies. The extended device handle is then inserted into a dummy object below the blue sheet (Fig. 6.7 (c-d)). Note that the blue cover sheet can also be placed as per requirement of the medical users such that the base of the device is hidden underneath and only the stylus is visible.

Figure 6.7 Application of physical constraints on the device for simulation of minimally invasive surgery. (a-b) creation of device handle extension using medical instruments, (c-d) simulation of OR environment using the extended device.

The modified device offers a different DOF than the original device. The modified device allows the user to move in and out of the scene and it pivots about the point of insertion into the dummy objects. Therefore, it offers only restricted motion in x and y

directions depending upon how far it has been moved in or out. The range of motions allowed are conical as shown in Fig. 6.8 where it can be seen that the range of motion varies as the stylus is moved in and out. The location of the pseudo end effector is then calculated based on actual distance of the pivot point from the HIP. Mathematical formulation for such stylus motion is presented in the Chapter 7 with details of implementation of the application.

(a)

(b)

Figure 6.8 Change in effective motion range when the device stylus is moved (a) inwards, (b) outwards.

6.4 Summary

In this chapter, we have extended the capabilities of the desktop haptic devices by proposing novel solutions to limitations of these devices. Virtual and physical constraints

have been proposed to provide for a more realistic interaction experience. The novel depth-based haptic rendering algorithm has been extended for multiple contact point interaction by tracking the motion of several points defined with reference to the HIP. These haptic contact points are pseudo-HIPs, distributed around the actual HIP. Collision detection and force computation for each point take place as if each were the actual HIP. The resulting forces are combined based on the relative locations of these points in 3D space. We have also proposed dynamic mapping of haptic workspace when the obtained depth has a greater resolution than the haptic device. This can be easily extended to x and y dimensions as well. Physical constraints have been applied to Geomagic Touch devices to simulate minimally invasive surgical procedures. The modification to the device results in different set of DOFs that matches the DOF required for simulation of such medical procedures. In the next chapter we will describe how such physical or virtual constraints can be used in different applications.

Chapter 7 Application Examples

In this chapter, we present different applications developed using the algorithms proposed in the previous chapters. Several applications have been developed to demonstrate the usefulness of the proposed haptic rendering algorithms. An arthroscopic surgery training simulation system has been developed along with a few training exercises. We have also simulated acupuncture therapy – a traditional Chinese medicine technique, virtual palpation and virtual venepuncture using the tangible images approach. Lastly, a ‘touch desktop’ application for touching GUIs and other interfaces displayed on the screen in a Windows operating system has also been developed.

7.1 Arthroscopy simulation

The first example that we present here is from the area of simulation and training in medical practice. We use the image-driven haptic rendering approach to simulate arthroscopic surgery of the knee. Like many orthopedic surgery procedures, arthroscopic surgery is done on joints in a minimally invasive way by making small incisions on the skin through which a miniature endoscopic camera is inserted along with pencil-sized surgical instruments. It demands from the surgeons to acquire special motor-skills while learning complex stepwise tasks including positioning (triangulation) of the instrument in front of the surgical camera (arthroscope), examination of the tissues in the surgical area, and using different instruments for cutting away and removing damaged parts of the tissues. The major challenge for the trainee surgeons here is that the surgical area is observed as a 2D image on the video monitor which is physically displaced from the

remotely controlled instruments.

Also, the arthroscope is not a common forward-looking but a wide-angle oblique-viewing camera. This means the view direction is not along the axis of the arthroscope cylinder but rather at a certain tilt from it, e.g., at the angles of 30°, 70° or 90° (Fig. 7.1). Consequently, the view direction changes as the arthroscope is rotated about its axis. Since the movement of the arthroscopic camera is very restricted due to the fixed incision point and limited space in the joint cavity (about 8-9 cm³), therefore an oblique-viewing camera has significant advantages over a forward-looking camera as the former provides a wider range of viewing directions. Thus, it provides greater freedom in desired directions to view target objects. Moreover, the arthroscope has a light source attached to its end, which is actually a fiber-optic cable transmitting light to the surgical area. Finally, due to a short focal length, there are some non-linear peripheral distortions of the image.

Figure 7.1 Arthroscope motions and rotations changing the camera view.

These properties of the camera create further challenges for the trainees in manipulating the arthroscope to be able to see the surgical instruments inserted through

the incision made a few centimeters away from the camera incision point. Given that the actual motions of the working pieces of the arthroscope and the instruments are done in a very limited space, it requires very high precision of motor skills to be achieved by the trainees. To achieve these skills, the surgeons have to go through special training programs. Currently, arthroscopic training is provided by commercial, vendor-driven courses which combine proprietary hardware based on real arthroscopic cameras and instruments with physical or simulated models of the surgical areas. There are also cadaveric courses; however they are uncommon and expensive.

On the other hand, simulation of arthroscopic surgery can be done entirely in virtual reality environments using common personal computers and haptic devices. It is very feasible because the surgeons are detached from the actual three-dimensional surgical scene, seeing it as a two-dimensional image on a video monitor, which can be replaced by a computer monitor, and operating within the scene with remotely controlled instruments, which can be quite realistically simulated with desktop haptic devices.

7.1.1 Related work

In recent years, arthroscopic simulation systems have received significant amount of research effort. Most existing techniques are based on reconstruction of the scene in 3D using MRI or CT scan data of the knee joint. For example, in (Mabrey et al., 2010), Marbey et al. discuss a knee arthroscopic surgical trainer system, where virtual anatomical 3D models are constructed from high resolution cryo-sectioned data sets, such as those available in the Visible Human Project (Ackerman, 1997). The system consists of a pair of high-fidelity commercial haptic devices that simulate the feel of the arthroscope

and the probe inside the knee. Two monitors are used: one to render the view as seen by the surgeon and the other one to display an instructional guide for the procedure being taught. In (Heng et al., 2004), Heng presents a virtual reality training system for knee arthroscopic surgery consisting of custom-made haptic devices interacting with 3D models created from the Visible Human Project (Ackerman, 1997) database. Tasks that can be performed with this system include standard inspection of the knee joint for recognizing major landmarks and soft tissue cutting simulated by editing the tetrahedral mesh. The system presented in (Megali et al., 2002) constructs 3D models from a CT scan of a mock-up leg. Although it includes learning modules for anatomical understanding, the structural deformation is not modeled and only limited haptic feedback is provided. The techniques developed by Touch of Life Technologies Inc. enhance the system by producing a number of variations in the virtual patients by altering morphology and pathologies displayed from the data. Bayona (Bayona et al., 2003) presents a low-cost arthroscopic simulation system where a commercial virtual laparoscopic interface is employed as the surgical instrument for delivering haptic feedback. 3D models of the joints are extracted from the Visible Human dataset and represented in VRML format. The system is suitable only for acquiring hand-eye coordination and general joint inspection. Another simulator for arthroscopy was presented by Muller and Bockholt in (Muller & Bockholt, 1998). It allows for reconstruction of 3D models from MRI data and manages interaction with virtual instruments, and a 2D graphical user interface is provided for visual feedback. However, it lacks any haptic feedback. Gibson et al. (Gibson et al., 1996), used volumetric object representations derived from MRI data for simulation of knee arthroscopy. It relies mostly on high-end workstations for

computationally intensive operations of calculating deformations and real-time visual rendering. The object models produced are smoothed before calculating surface normals, so that stable haptic feedback is delivered to the user via commercial haptic devices. Another work using volumetric representation for anatomical structures and employing commercial haptic devices for delivering force feedback is presented in (Mabrey et al., 2002). Here, the visual and mechanical aspects of the knee are simulated through a combination of control, modeling, and content programs. A control module is responsible for collision detection and communication with the modeling part of the system. The haptic devices interact with the underlying volumetric representation of the knee, and the content programs use volume rendering to display the actual appearance of the knee on the monitor. Sherman et al. (Sherman et al., 1999) developed a Virtual Environment Knee Arthroscopy Training System, where a voxel-based collision detection system is used for collisions with solid objects such as bones. Behavior of deformable structures is modeled using the linear tetrahedral finite element method. Real-time rendering of only simple models can be guaranteed with this technique due to its computationally expensive nature. A custom force feedback device is attached to the mock instruments to provide haptic feedback to the user.

A few virtual knee arthroscopy simulators have also been commercialized. GMV Innovating Solutions developed insightArthroVR (insightArthroVR, 2010)—a virtual arthroscopy trainer for knee and shoulder surgery, which was later acquired by Simbionix Corporation in early 2012 and remarketed as Arthro Mentor training system (Simbionix, 2012). It is a multi-purpose tool, which includes various virtual anatomical models, an adapted arthroscopic instrument set, and a high fidelity haptic feedback system. It

simulates diagnostic and therapeutic procedures as well as presents statistic data for training assessment. The complete system has a didactic sequence of training cases along with other hardware features such as an adjustable platform. VirtaMed AG, a Switzerland based ETH spin-off, has developed a virtual arthroscopy training platform ArthroS (VirtaMed, 2012), which provides training for diagnostic and therapeutic arthroscopy, uses original endoscopic tools, and provides objective performance feedback. The accompanying software includes a number of guided procedures for basic skill training and a few different virtual patients for diagnostic sweeps and meniscus operations.

The virtual arthroscopic camera and the virtual surgical instruments can be controlled either by proprietary made devices, or by common desktop haptic devices. In either case, for an immersive simulation the motions of the haptic device handles must be constrained by the virtual incision point and should exactly mimic the motions of the actual surgical camera and instruments while the respective simulated view on the computer monitor should look as close as possible to the actual surgical view. A few camera models have been proposed to incorporate oblique-viewing property of endoscopic cameras. Yamaguchi (Yamaguchi et al., 2003) was the first to formulate a camera model and calibration method for such cameras to be used in an augmented reality system. Based on Tsai's camera model (Tsai, 1987), it establishes the extrinsic parameters of the camera as a function of rotation about the arthroscope axis. Calibration is performed by attaching optical markers to the camera and tracking them while rotation parameters are measured using a rotary encoder. In (Chenyu & Jaramaz, 2008), the calibration was simplified by tracking the arthroscope rod instead of the camera head hence reducing the number of parameters to be estimated. However, two additional optical trackers were used in an

attempt to eliminate the need for the rotary encoder. Buck et al. (Buck et al., 2007) generalized Yamaghuchi's method by incorporating the changes to camera intrinsic parameters and adding a radial distortion component. Optical trackers are used for calibration but the calibration result is slightly less accurate. A real-time method for calibration and removal of radial distortion is proposed in (Melo et al., 2012), along with a formulation of relative rotation between endoscope axis and camera head. These methods provide calibration procedures for an endoscopic camera so that a mapping between the 3D scene and the resulting 2D image can be estimated.

7.1.2 Image-based approach

We have proposed a hybrid image-based approach for arthroscopy simulation, where real arthroscopy images are mostly used for visual and haptic rendering rather than 3D models of the surgical area. At the preprocessing stage, a panoramic image of the entire knee cavity is created by stitching a few representative frames of a real arthroscopy video. This approach can produce believable interaction with much shorter turnaround time than that of the full 3D modeling approach provided there is a reliable and quick pipeline for making panoramic images from the surgical videos taken during the operation. A straight-forward but tedious way of doing it is to manually make an image by adding the matching parts from the consecutive images (Fig. 7.2).

Alternatively, methods can be devised for automatically creating such panoramic images. Within the scope of the externally funded project, a method to automate this process is currently being developed, an initial study of which was presented in (Pestrikov & Sourin, 2013). It involves increasing the contrast of arthroscopic images to increase the

gradient of brightness. Analysis of the gradient is used to identify feature points in individual images. The precision of the algorithm is improved by using the Single Scale Retinex (SSR) algorithm, which eliminates lighting distortions. Speeded-Up Robust Features (SURF) were used to extract feature points which were compared using Fast Library for Approximate Nearest Neighbors (FLANN). A homographic transformation is determined to match the feature points in different frames of the video and using this transformation the individual frames are stitched into a single panoramic image.

Figure 7.2 Manual stitching of the individual frames into a panoramic image of the knee meniscus. Gaps are left for illustration purposes.

This image file then is visualized through a moving circular window that follows the motion of the virtual arthroscopic camera (Fig. 7.3). The image displayed should correspond to what can be seen for any given location and orientation of the arthroscope. The panoramic backdrop image is then augmented with a few 3D models of deformable tissues as well as models of the surgical instruments which will be seen by the camera.

Figure 7.3 Panoramic image created by stitching images obtained from the actual surgical video and the hybrid simulated view.

To haptically render the backdrop image as if it were the actual 3D scene, a depth map is then extracted from the intensity values of its pixels and some noise removal filters are applied to improve the approximation of the depth map from pixel intensities. Arthroscopic images are fairly suitable for depth acquisition as they are always illuminated from the front by the light source attached to the arthroscopic camera. Moreover, usually the captured objects do not have irregular colour patterns that can be confused with surface variations. Therefore, a direct correspondence between the image intensity and the scene depth can be assumed. The image is first filtered by a carefully selected colour filter based on the prior knowledge about the colour of the objects in the scene. For example, in case of the knee arthroscopy, the joint cavity usually has bone structures, menisci and ligaments, which have somewhat uniform colour, whereas some image areas representing muscle have a higher red component, as compared to green and blue components. When image intensity is considered to be directly proportional to the scene depth these areas will be interpreted as positioned further away in the scene due to lower overall intensity of these pixels. However, partial filtering of blue and green channels produces an image with a uniform intensity across the high red channel muscle areas of the image as well as the brightly coloured bone, menisci and ligaments in the

image. The colour-filtered image is converted to grayscale, and the intensity values are used as a depth map, which is normalized to occupy the entire depth range of the scene. A second filtering pass is applied to remove noise in the extracted depth map.

Figure 7.4 Preprocessing of arthroscopic images for depth estimation. (a) RGB image, (b) partially filtered GB channels, (c) grayscale image without GB filtering, and (d) grayscale image with GB filtering.

The preprocessing of the RGB image for depth extraction from an arthroscopic image is illustrated in Fig. 7.4. The brightness of the white areas remains nearly the same while the red muscular regions are illuminated brighter in the colour-filtered image. Note the difference in the highlighted area in Fig. 7.4 (c-d).

As mentioned in Chapter 4, in order to produce render material properties of the objects in the scene, image segmentation can be used and different stiffness can be assigned to each segment. Consider the image in Fig. 7.5(a), which shows an image captured during minimally invasive arthroscopic surgery. The image is segmented automatically, using a brightness threshold, to produce a segmentation map shown in Fig. 7.5(b). While this is sufficient for illustration purposes, sophisticated interactive image

segmentation techniques can be used to extract a better approximation of human tissue. Different haptic properties are interactively assigned to different parts of the image. Here, the choice of physical properties to be assigned to different segments has been made based on subjective feedback from a practicing surgeon about the tangible properties of different tissues. Moreover, the perceived haptic properties of different tissues in the simulation has also been subjectively validated by the surgeon. In this case, the physical properties are set so that the haptic cursor is allowed to penetrate into the tissue once it is in the white region on the segmentation map. A highly viscous effect is activated to give an illusion of cutting through the tissue. The original image is still used as a visual guide for haptic interaction.

Figure 7.5 Haptic rendering of different physical properties in arthroscopy simulation, (a) actual image with arrows showing direction of rendered force, (b) segmentation map.

<http://www.youtube.com/watch?v=deqQ8ggqhDk>

Two devices, the *camera haptic device* and the *instrument haptic device* are used to control the virtual arthroscopic camera and the virtual surgical tool respectively. Two Geomagic Touch haptic devices are physically modified as described in Section 6.3 and arranged in a setup shown in Fig. 7.6. As a result, the constrained motion of the haptic devices reflects the insertion of the camera and the tool through the incision points.

Figure 7.6 Geomagic Touch haptic devices used for simulation of minimally invasive surgery.

<http://www.youtube.com/watch?v=UQbcFOhn6hA>

A special method has to be used to model visualization with the oblique-viewing arthroscopic camera. Here, we need to display the relevant part of the existing panoramic image following the motion of the virtual camera controlled by the haptic device mimicking the actual surgical camera. Hence, we have to solve a problem of displaying various parts of the image as if they were seen by the actual surgical camera. The virtual arthroscopic camera is specified by three parameters: its position $\vec{P}_{c-p}$, view direction $\vec{P}_{c-f}$ and the up vector $\vec{P}_{c-u}$ in the modeling coordinate system XYZ that has its origin at the camera incision point $\vec{P}_{pvt}$ in the virtual environment. Each of these parameters is calculated after the virtual camera is initialized by the user by pressing the primary stylus button on the camera haptic device. At this instant, the virtual camera is considered to be just inserted into the knee therefore it is located at $\vec{P}_{c-p}^i$ a short distance d_i on the negative z -axis, looking at $\vec{d}_f$ which is at 30° tilt angle from the negative z -axis (i.e., towards the positive x -axis) in case of the virtual camera being manipulated with the left hand. Then, the up vector of the camera is $\vec{d}_{up}$ which is at 120° from the negative z -axis towards the positive x -axis.

At initialization, the HIP position $\vec{\mathbf{P}}_{HIP}^i$ and the transformation matrix $\mathbf{H}_{HIP}^i$ are captured. They are then used to compute the coordinates of the physically fixed incision point $\vec{\mathbf{P}}_{pvt}$.

$$\begin{aligned}\vec{\mathbf{t}} &= [0 \quad 0 \quad -d_{h \rightarrow pvt}^i]^T \\ \mathbf{M} &= \begin{bmatrix} \mathbf{I} & \vec{\mathbf{t}} \\ \vec{\mathbf{0}}^T & 1 \end{bmatrix} \mathbf{H}_{HIP}^i \\ \vec{\mathbf{P}}_{pvt} &= [m_{12} \quad m_{13} \quad m_{14}]^T\end{aligned}\tag{7.1}$$

where $\mathbf{I}$ is a 3×3 identity matrix, $d_{h \rightarrow pvt}^i$ is the predefined initial distance between the HIP and the pivot point as in Fig. 7.7 and m_{12} , m_{13} and m_{14} are entries of $\mathbf{M}$ in column major order. Note that point $\vec{\mathbf{P}}_{pvt}$ can be located anywhere in the virtual space depending on where the HIP was located when the virtual camera was initialized. Therefore, it is transformed to coincide with the origin of the virtual space as $\vec{\mathbf{P}}'_{pvt} = \mathbf{M}^{-1} \mathbf{H}_{HIP}^i{}^{-1} \vec{\mathbf{P}}_{pvt}$. Now that the virtual camera is initialized, we can proceed to compute the coordinates of the camera in subsequent frames. To this end, the distance between the current HIP $\vec{\mathbf{P}}_{HIP}$ and the incision point is calculated as

$$d_{h \rightarrow pvt} = \left\| \mathbf{M}^{-1} \mathbf{H}_{HIP}^i{}^{-1} \vec{\mathbf{P}}_{HIP} - \vec{\mathbf{P}}'_{pvt} \right\| \tag{7.2}$$

Figure 7.7 Location of HIP, proxy point and incision point and the relative distances between them with reference to the modified haptic device.

The position of the camera $\vec{P}_{c-p}$ in the virtual modeling space is then calculated as:

$$d_{proxy \rightarrow pvt} = d_i + d_{h \rightarrow pvt} - d_{h \rightarrow pvt}^i$$

$$\vec{v} = \mathbf{M}^{-1} \mathbf{H}_{HIP}^i{}^{-1} \vec{P}_{HIP} - \vec{P}'_{pvt} \quad (7.3)$$

$$\vec{P}_{c-p} = -d_{proxy \rightarrow pvt} [\alpha \quad \beta \quad \gamma],$$

where α , β and γ are the cosines of the angles between the vector $\vec{v}$ and the x , y and z axes respectively.

The point $\vec{P}_{HIP}$ changes with the camera HIP movement while $\vec{P}'_{pvt}$ remains constant. Hence, the changes in α , β and γ result in the new camera position being calculated for every frame. As the modified haptic device handle moves in and out, the distance $d_{h \rightarrow proxy} = d_{h \rightarrow pvt} + d_{proxy \rightarrow pvt}$ remains constant while the distances $d_{proxy \rightarrow pvt}$ and $d_{h \rightarrow pvt}$ change. This simulates the effect of the camera being moved closer to or away from the surgical area. Moreover, in order to ensure that the virtual camera remains within the surgical field, the distance $d_{proxy \rightarrow pvt}$ is bounded by $[d_i, r]$ where r is the maximum depth of the knee cavity. Fig. 7.8 illustrates the virtual camera initialization and the range in which the camera moves.

Figure 7.8 Parameters defining the virtual arthroscopic camera in the virtual environment. (a) at initialization, (b) at subsequent frames.

The arthroscopic camera may have different tilt angles ϕ of 30°, 70° or 90°. The orientation of such a camera is a function of the rotation about the virtual arthroscope axis. This is formulated by obtaining the gimbal angles $\vec{\mathbf{G}}_{HIP} = [g_x \ g_y \ g_z]^T$ of the camera device at every frame. The view direction $\vec{\mathbf{P}}_{c-f}$ and the up vector $\vec{\mathbf{P}}_{c-u}$ of the camera are then calculated as:

$$\begin{aligned}\psi_x &= -\cos^{-1}(\vec{\mathbf{v}}_{yz} \cdot \vec{\mathbf{z}}_{axis} / \|\vec{\mathbf{v}}_{yz}\|) \\ \psi_y &= -\cos^{-1}(\vec{\mathbf{v}}_{xz} \cdot \vec{\mathbf{z}}_{axis} / \|\vec{\mathbf{v}}_{xz}\|) \\ (\theta_x, \theta_y, \theta_z) &= \begin{cases} (\psi_x, \psi_y, g_z) & \text{if } \vec{\mathbf{P}}_{c-p}^x > 0 \text{ and } \vec{\mathbf{P}}_{c-p}^y > 0 \\ (\psi_x, -\psi_y, g_z) & \text{if } \vec{\mathbf{P}}_{c-p}^x < 0 \text{ and } \vec{\mathbf{P}}_{c-p}^y > 0 \\ (-\psi_x, \psi_y, g_z) & \text{if } \vec{\mathbf{P}}_{c-p}^x > 0 \text{ and } \vec{\mathbf{P}}_{c-p}^y < 0 \\ (-\psi_x, -\psi_y, g_z) & \text{if } \vec{\mathbf{P}}_{c-p}^x < 0 \text{ and } \vec{\mathbf{P}}_{c-p}^y < 0 \end{cases} \quad (7.4) \\ \vec{\mathbf{P}}_{c-f} &= \mathbf{M}_x \mathbf{M}_y \mathbf{M}_z \vec{\mathbf{d}}_f \\ \vec{\mathbf{P}}_{c-u} &= \mathbf{M}_x \mathbf{M}_y \mathbf{M}_z \vec{\mathbf{d}}_{up}\end{aligned}$$

where $\vec{\mathbf{v}}_{yz}$ is the projection of $\vec{\mathbf{P}}_{c-p}$ on the yz -plane and $\vec{\mathbf{v}}_{xz}$ on the xz -plane, $\mathbf{M}_x$, $\mathbf{M}_y$ and $\mathbf{M}_z$ are matrices for rotation about x , y and z axes by angles θ_x , θ_y and θ_z respectively, and $\vec{\mathbf{P}}_{c-p}^x$, $\vec{\mathbf{P}}_{c-p}^y$ are x and y coordinates of $\vec{\mathbf{P}}_{c-p}$ respectively.

A similar mathematical formulation is then used for the instrument haptic device to compute virtual tool position. However, the reference point then is not located at the origin of the coordinate system but at another point that lies on the x -axis. The distance between the two insertion points is user tunable and corresponds to the distance between the two incision points on the real joint. Since the virtual surgical tool is defined by its position in the modeling space and the reference point, so an equation similar to Eq. 7.3 is sufficient for representing the tool. Thus, the viewing direction and the up vector are computed as in Eq. 7.4 only for the oblique-viewing arthroscopic camera. The surgical tool is then drawn between the calculated tool position and the tool insertion point. The

angle of rotation of the tool about the tool axis corresponds to the z -rotation of the instrument haptic device, i.e. the third coordinate of the gimbal angle vector of the instrument haptic device.

Since the camera moves in and out of the knee cavity following the motion of the haptic device handle, a planar panoramic image will produce distorted views with the oblique-viewing camera as it moves very close to the plane. Therefore, the panoramic image is displayed as a texture on the inside surface of a moving hemisphere with a radius corresponding to the actual surgical area (e.g., 20 mm in case of the knee surgery). As the camera moves in the virtual surgical space, the orientation and center of the hemisphere are modified to ensure that the camera view direction is always orthogonal to the hemisphere.

As mentioned previously, the incision point coincides with the origin of the modeling coordinate system. Therefore, in order for the viewing direction to be orthogonal to the hemisphere (Fig. 7.9), the hemisphere is centered at the origin at initialization. Let the intersection between the viewing direction $\vec{P}_{c-f}$ and the hemisphere be the point $\vec{P}_{v-h}$ and the normal to the hemisphere at the intersection point be $\vec{P}_{n-v}$. Then, the transformation matrix applied to the hemisphere is given by $\mathbf{R}_v$ that rotates the hemisphere by an angle $-\theta_r$ about the axis $\vec{P}_r = -\vec{P}_{c-f} \times \vec{P}_{n-v}$. Hence, the hemisphere center $\vec{P}_{h-c}$ moves away from the origin and ensures that the viewing direction remains orthogonal to the hemisphere. The moving hemisphere is illustrated in Fig. 7.9.

Figure 7.9 Positioning of the textured hemisphere (a) at initialization, (b) at subsequent frames. Here, 0 is the origin and the dotted hemisphere in (b) shows the previous hemisphere location.

The images produced in this virtual environment are nearly identical to those in the actual surgical video, as it can be seen in Fig. 7.10. It shows how the inside surface of the whole textured hemisphere looks like along with the views through the camera at several instances when the camera haptic device is moved. Some small distortions can be seen in the last image where the camera is moved to the peripheral areas of the knee which are seldom navigated and almost never operated upon.

Figure 7.10 (a) Hemisphere with panoramic image as a texture mapped on its inside surface, (b) Views generated by the camera looking at various parts of the hemisphere.

3D objects can be used to model deformable or editable parts of the scene. Represented by polygon meshes or defined by mathematical functions they can be carefully placed to match the 2D image background and edited upon interaction with the virtual tool as described in Section 5.1 and Fig. 5.4.

7.1.3 Training exercises

Initial common arthroscopy training demands from the trainees to acquire motor-skills while learning complex stepwise tasks including incision of the side-looking arthroscope, positioning of the instrument in front of the arthroscope, and working with scissors and graspers. The major challenge for the trainees is that the observed 2D image is not physically collocated with the remotely controlled instruments. As for the instruments, such as scissor and graspers, the additional challenge is to learn how to open and close their jaws with adequate opening angles so that they will not damage the surrounding tissues. Therefore, a few exercises for acquisition of the required basic skillset through practice have been developed.

Positioning of the instrument in front of the arthroscope is the first exercise which the trainees will learn with the developed virtual arthroscopic surgery simulation system. It is not a trivial task since the camera is side-looking and the incision of the instrument has to be done in a rather blind way before it appears in the arthroscope view. The finding of relative positions of the instrument and the arthroscope along with their position with reference to the anatomical structures in the knee joint is known as triangulation. The arthroscope and the instrument are said to be triangulated once the instrument can be seen in the arthroscopic view on the screen. Sometimes the surgeons also strike the arthroscope

with the tool to better judge their relative locations. A point-plane collision is used to simulate such a collision of the instrument with the arthroscope. The arthroscope is represented by a point whereas the instrument is represented by a plane defined by three points as shown in Fig. 7.11. The instrument position at initialization is set to $\vec{P}_{t-i}$. The three points $\vec{q}_1$, $\vec{q}_2$ and $\vec{q}_3$ are defined by an offset from the initial instrument position $\vec{P}_{t-i}$. All subsequent transformations on $\vec{P}_{t-i}$ are also applied to these points, hence they follow the instrument position in the same configuration as they are initialized. The transformed points are computed as

$$\vec{q}'_i = \mathbf{N}_x \mathbf{N}_y \mathbf{N}_z \vec{q}_i, \quad i \in \{1,2,3\} \quad (7.5)$$

where $\mathbf{N}_x$, $\mathbf{N}_y$ and $\mathbf{N}_z$ are rotation matrices for instrument haptic device analogous to $\mathbf{M}_x$, $\mathbf{M}_y$ and $\mathbf{M}_z$ in Eq. 7.4 for camera haptic device.

Figure 7.11 Triangulation: Collision of instrument with the virtual arthroscope.

A collision between the arthroscope and the instrument is detected when the projection of the arthroscope $\vec{P}_{c-proj}$ along the arthroscope axis lies inside the triangle represented by points $\vec{q}_1$, $\vec{q}_2$ and $\vec{q}_3$ (Fig. 7.11), the point $\vec{P}_{c-p}$ crosses to the side of the normal $\vec{n}_t$ of the plane and the point $\vec{P}_{c-p}$ is within a short distance ε from the plane. First, vectors from each of the triangle vertices to the arthroscope projection point $\vec{P}_{c-proj}$ are calculated. Then, the angles formed by these vectors are computed. If the

sum of these angles equals 2π then the projection point lies inside the triangle. The signed distance of the point $\vec{\mathbf{P}}_{c-p}$ from the plane is then computed (see Appendix B). More specifically, it is expressed as

$$\begin{aligned}
\vec{\mathbf{a}}_i &= \vec{\mathbf{q}}_i - \vec{\mathbf{P}}_{c-proj} \quad i \in \{1,2,3\} \\
\varphi_1 &= \cos^{-1}(\vec{\mathbf{a}}_1 \cdot \vec{\mathbf{a}}_2 / \|\vec{\mathbf{a}}_1\| \|\vec{\mathbf{a}}_2\|) \\
\varphi_2 &= \cos^{-1}(\vec{\mathbf{a}}_2 \cdot \vec{\mathbf{a}}_3 / \|\vec{\mathbf{a}}_2\| \|\vec{\mathbf{a}}_3\|) \\
\varphi_3 &= \cos^{-1}(\vec{\mathbf{a}}_3 \cdot \vec{\mathbf{a}}_1 / \|\vec{\mathbf{a}}_3\| \|\vec{\mathbf{a}}_1\|) \\
d &= \vec{\mathbf{n}}_t(\vec{\mathbf{P}}_{c-p} - \vec{\mathbf{q}}_1).
\end{aligned} \tag{7.6}$$

A collision is detected between the instrument and the arthroscope if $\sum_{i=1}^3 \varphi_i = 2\pi$ and $0 < d < \varepsilon$. A force is then applied on the arthroscope haptic device in the direction $-\vec{\mathbf{n}}_t$ until $\vec{\mathbf{P}}_{c-p}$ crosses the plane back to the opposite side.

In arthroscopic surgery, the surgeon needs to grab and remove floating pieces of tissue that may have been detached from the meniscus or ligaments during some procedures. A grasper tool is used for such a task and its jaws may collide with one or more anatomical structures such as bones. Repeated collisions with the bones or other structures may damage them and hence must be avoided. Therefore, opening and closing of grasper jaws without colliding with different anatomical structures is the next training exercise we simulate. We use multiple contact point interaction described in Section 6.1 to simulate this procedure. A number of haptic contact points are defined at corners and edges of the virtual instrument (Fig. 7.12 (a)). These points are defined in terms of the offset distance from the initialized instrument position $\vec{\mathbf{P}}_{t-i}$. All subsequent transformations are applied to these points as in Eq. 7.5. Collisions for each point are then detected with the depth map of the panoramic image and feedback force calculated based

on a pixel mask associated with each point. The resulting forces are averaged to form the force vector applied to the haptic device at the instrument position $\vec{P}_{t-p}$. Force feedback calculated in this way is illustrated in Fig. 7.12 (b), where two HCPs are chosen; one for each jaw and $\vec{P}_{t-p}$ is located in the center of the grasper instrument.

(a)

(b)

Figure 7.12 Opening and closing of grasper jaws using multiple contact interaction (a) location of HCPs with respect to the instrument (b) forces calculated with 2-point instrument.

Floating pieces of tissue are also simulated using the image-based approach. A floating object is defined by a textured polygon at a given point. This point is initialized at a random location while moving in a randomly selected direction and rotating about its center in clockwise or counter-clockwise direction at a random speed. A polygon is drawn at this point and is textured with a GIF image taken from floating objects in a real surgery video. The polygon follows the movement of the point which gets reflected from the pre-defined boundaries of the modeling space. Moreover, it is also reflected when it

collides with the depth map of the tangible image. Surface normals required for updating the direction of the moving point are defined at initialization for boundaries of the graphic space while normals for the depth map are obtained using the depth-based haptic rendering algorithm presented in Section 4.1. In order to simulate grasping, six different points are defined at initialization, each located at an offset from $\vec{P}_{t-i}$. All subsequent transformations are applied to these points as defined in Eq. 7.5. The result is a six-point pyramid that continuously follows the instrument position $\vec{P}_{t-i}$. If at any time, the point defining the floating object lies within this pyramid (as shown in Fig. 7.13) and the grasper instrument is closed (e.g., by pressing of the secondary stylus button) the floating object is considered grasped. The test for checking whether the point lies inside the pyramid is performed by checking on which side of each of the five planes making up the pyramid, the point lies.

Figure 7.13 Six point pyramid used for grasping. <http://www.youtube.com/watch?v=Ksa2RjXk1Zw>

Simulation of removal of damaged parts of a tissue such as meniscus was discussed in Section 5.1 to illustrate haptic rendering of hybrid scenes. The challenges involved in such simulation were also highlighted. Note that there still will be complicated tasks that cannot be modelled using the tangible image approach. For example, anterior cruciate ligament (ACL) reconstruction or replacement is a complex task that cannot be reliably simulated with the image-driven approach. Therefore, as part of the externally funded

project a three-dimensional simulation is also under development. 2D images of the exterior of the knee are still used to guide the insertion of the camera and the tool at the appropriate positions as illustrated in (Rasool et al., 2013).

Through this example we have illustrated various aspects of the tangible images paradigm. These include haptic perception of geometry of the visible scene and multiple contact point interaction. It also illustrates haptic rendering of hybrid scenes where 3D collisions between tool and camera are used in combination with image-based haptic rendering. This example also demonstrated how interactive scene modifications can be modeled in tangible images.

7.2 Other image-driven medical training simulations

7.2.1 Simulation of acupuncture therapy

A simulation tool that simulates acupuncture therapy was developed as another illustrative example of the tangible images approach. Acupuncture therapy involves insertion of very thin needles into the skin at various points on of the patient's body (Kaptchuk, 2002; Mann, 2000; National-Institutes-of-Health, 1997). These needles are inserted into various depths depending upon the point being stimulated. In the developed simulator, the image is displayed on a plane and a 3D haptic cursor is represented by a 3D geometric model of the acupuncture needle. While looking at the image, the 3D needles can be inserted into what appears to be a 3D hand. After insertion, a part of the needle becomes invisible since it is behind the image, while the visible part remains on top of the image. The scene is expected to be visually modified upon insertion of the needle. In Chapter 4, we proposed the used of image patches to superimpose the background image.

Since the modification in the scene is restricted to a small area in the vicinity of the interaction point, a small circular texture patch of the deformed surface centred at the point of haptic interaction is used as shown in Fig. 7.14. It can be seen that the texture patch reliably produces the desired visual skin deformation as the texture patch is made to blend into the background image.

Figure 7.14 Simulation of skin deformation by superimposition of small texture patch onto the background image at the point of interaction

Since the needles have to be inserted into the body, the internal properties like density have also been defined for the haptic object. When the user hits an acupuncture point two different types of sensation – either a slight squeezing sensation or a slight vibration – are produced on the haptic device based on the selection by the surgeon. This choice of vibration or pull sensation was made based on the discussions with a practicing doctor of Chinese medicine. The coordinates of points on three meridians are predefined and stored. An acupuncture point is then a tiny circular area defined by a predefined radius around each of these coordinates. Once the needle is inserted into the hand its movement is restricted to a path in the direction it is moving at the time of contact with the image plane. The net effect on the device is a combination of high viscosity effect simulating density of the hand and a magnetic force pulling the device to the restricted path. A snapshot of the simulator is shown in Fig.7.15.

Figure 7.15 Screenshots of acupuncture simulation (a) insertion of needle at incorrect point (b) insertion of needle at correct point. <http://www.youtube.com/watch?v=GCs3sHUyqPk>

Alternatively, a hybrid scene representation can also be used for simulation of acupuncture therapy. A photograph of an actual object is superimposed with a 3D haptic container. Fig. 7.16 shows an image of the patient head displayed on a plane with a transparent ellipsoid behind it, which is a very simplified approximation of the human face. Since the application does not require the user to feel the surface by moving along the surface but rather the interaction is limited to one point at a time, an ellipsoid with predefined internal properties is a sufficient approximation. The immersion can be increased by using haptic containers that fit the objects in the scene more closely.

Figure 7.16 Screenshot of a simulation of acupuncture therapy by super imposing a 3D haptic container onto a 2D image.

7.2.2 Simulation of palpation

An image-driven virtual palpation system (Yasmin & Sourin, 2013) for medical training was also developed as part of the externally funded project. The image of a patient is augmented with invisible haptic models of the internal organs. These 3D models are defined by implicit functions resulting in simple collision detection and small size of the models. Thus, the system can be easily ported to any Web-enabled or collaborative environment. The function-based approach is very scalable as different sized organs (healthy or deformed) can be created by adjusting a few parameters. In order to simulate visual deformation, the image is divided into a regular grid for image warping. The grid points local to the HIP are displaced based on their relative distances from the point of interaction. The visual deformation produced in this way is shown in Fig. 7.17. The image-driven approach allows easier patient-specific modelling than full 3D modelling of the patient's body.

Figure 7.17 A virtual palm palpating a malignant and enlarged liver. Notice the deformation near the belly button in the left image (Yasmin & Sourin, 2013).

<http://www.youtube.com/watch?v=tn5rwhw73PE>

7.2.3 Simulation of venipuncture

Another medical procedure that was simulated using the image-based approach as a part of the externally funded project is virtual venipuncture (Xia & Sourin, 2012). It is the

procedure of drawing blood from veins for testing or transfusion purposes. Actual photos of patient arms are used to provide quick simulation of different patients. Function-defined models of the veins provide for fast collision detection. Stable and realistic force feedback is provided by the haptic model for multi-layer soft tissues. The implemented system is shown in. These applications illustrate the ability of the tangible image approach to simulate user triggered visual deformations through addition of texture patches and image warping.

Figure 7.18 Image-driven venipuncture simulation (a) image and the function-defined 3D model of the veins, (b) screenshot of the implemented system. <http://www.youtube.com/watch?v=-ISlcU0vKcw>

7.3 Touch desktop application

We have implemented a pilot application for the Microsoft Windows operating system which makes ‘touchable’ any part of the desktop screen by allowing the user to select any window or region on the screen to be touch-enabled. The control of the windows cursor is transferred to the haptic device. Cursor control can then be toggled between mouse and the haptic device by pressing the primary stylus button on the haptic device. Once a window is selected, its handle is used to retrieve information about its position and size in pixel coordinates. Using this information, the haptic workspace is mapped to the graphic space, i.e. the window dimensions, and a screen capture is performed for the contents of

that window to obtain colour values for each pixel. These pixel values are used to generate force feedback as described in Section 4.1. Various adjustable parameters such as the mask size, threshold value, stiffness value, etc. can be tuned to suitable values, as shown by the GUI of the application in Fig. 7.19. Note that each of the individual modes can be enabled other than the default mode. When a window or region is selected in the texture mode, a histogram of the image is computed and displayed. The user is then allowed to choose two values defining the range of the histogram which are used to stretch the spectrum of the image (see Fig. 4.12 (b)). Touch interaction is quite believable for frontally illuminated scenes with slowly moving objects.

Figure 7.19 Snapshots of the touch screen application.

An example of the desktop tangible image application is shown in Fig. 7.20, where the contour perception mode is enabled to perceive boundaries of the buttons in a dialog box. When combined with auditory output, this can be a useful tool for visually impaired

users to interact with computers using graphical user interfaces. This example highlights the usefulness of the contour perception mode of the tangible images framework.

Figure 7.20 Use of desktop tangible image application for haptic interaction with graphical user interfaces in contour perception mode. The direction of the force at the current location of the HIP is shown as an arrow. <http://www.youtube.com/watch?v=5-JY-eE-Jw0>

The algorithm has also been applied to haptic video communication on the Web as proof of concept. The developed application is used to select the chat window of a video conversation over Skype™, as shown in Fig. 7.21, where the arrow displayed in the image shows the direction of the force applied to the haptic device at the current location of the HIP. Since colour information does not always reflect the depth of the scene accurately, therefore the perceived haptic sensation may not always be realistic. If the ground truth of the depth map (obtained using Microsoft Kinect, for example) can be transmitted with the colour video stream a better interaction experience can be ensured.

Figure 7.21 Haptic video communication: Proposed algorithm applied to video communication on the web. The direction of the haptic force at the current location of the HIP (at the nose) is shown as a red arrow.

Alternatively, in order to avoid the cursor diving into the depth discontinuity arising in facial images due to hair such as eyebrows, mustaches, etc., the offset mode of the proposed system can be used, as it is illustrated in Fig. 7.22. The binary map generated using contrast enhancement and image brightness manipulation is used to trigger the offset mode. There can be methods developed to generate such a binary map automatically based on image and histogram analysis. This will allow the offset mode with binary map to work with videos in real-time.

Figure 7.22 Offset mode applied to facial images. (a) 2D input image with blurry transitions, (b) the corresponding binary map with sharp discontinuities, (c) force feedback in the default mode, (d) force feedback in the offset mode.

As mentioned previously, the algorithm is suitable for interaction with moving objects as it works well in real-time. This is illustrated in Fig. 7.23, where the haptic device moves up and down following the hand movement of the other party in a Skype video conversation. The arrow shows the direction of force applied to the haptic device.

Figure 7.23 Interaction with the moving object in haptic video communication. The handle of the haptic device moves up and down following the motion of the hand in the Skype window.

<http://www.youtube.com/watch?v=zMReWQmTX30>

7.4 Other applications

There can be other applications developed using the algorithms proposed in this thesis as part of tangible images. Virtual museums can be created using the hybrid approach with invisible haptic models proposed in Section 5.2. These e-museums can be made available online with invisible haptic models having simplified geometry. This simplification will guarantee smaller size of the haptic models and hence reduce storage and bandwidth requirements. For example, a PDE based simplification (Ming-Yong et al., 2010; Sheng et al., 2010) was used for compression of 3D haptic models in Fig. 5.6.

We have already mentioned the use of depth sensors in video communication systems. A mechanism is required for compression of the captured depth map to be communicated with the colour stream. The feedback force can then be generated at the client computer.

7.5 Summary

In this chapter, we have presented a few applications as examples of tangible images approach. They illustrate the usefulness and feasibility of the proposed algorithms. An

image-driven arthroscopy training simulator was presented with a few basic training exercises. This application proves that the research goal of force computation from 2D images without 3D reconstruction has been achieved. Moreover, the application demonstrated the interactivity of 3D objects augmented to images in a hybrid scene. Other medical applications illustrated scene editing using image warping and superposition of images with small texture patches to simulate deformation. Moreover, a touch desktop application was also presented which highlights different application modes and interaction with dynamic scene content such as in Skype. Other prospective application areas were also highlighted.

Chapter 8 Conclusion and Future Work

This chapter summarizes the work presented in this thesis. In Section 8.1 the conclusion is made, and future work is discussed in Section 8.2.

8.1 Conclusion

In this thesis we first presented a survey of the relevant literature as well as a study of the current research in the area of image-based haptic interaction. The research niche for the development of tangible images – a set of methods for image-driven haptic interaction – was identified and motivated the subsequent R&D work. We then hypothesized that images of real scenes can be turned into “tangible images” which can be felt with haptic devices as if they were real 3D scenes following the principle “What You See Is What You Touch”. This can be achieved by mostly deriving haptic forces for visible surfaces of the objects straight from the images with pixel precision, as well as by segmenting the images and assigning physical properties to their parts, and by augmenting the images with invisible haptic objects. The scenes displayed in the images can also change following the haptic interaction either by image warping or by augmented editable parts.

In order to prove this hypothesis we proposed algorithms and methods for image-driven haptic interaction that have resulted in the following **research novelties**:

1. The novel depth-based image-driven haptic rendering algorithm has been proposed. It uses depth information either approximated from the image or obtained by depth sensing cameras. Mechanisms for combining geometric forces with force components for rendering surface texture and physical properties of objects have been proposed.

We also presented novel techniques for handling of special cases such as contours and non-geometric discontinuities in image intensity. This has been published in papers C2, C3, C4, C6 and C7 of the list of publications that follows this chapter.

2. The innovative method of haptic interaction with hybrid scene representation is presented. In this method, the 2D image is augmented with visible or transparent 3D objects. The visible 3D objects, used as editable parts of the scene, are interacted with either based on generation of a composite depth map or using 3D haptic interaction in conjunction with image-driven haptic rendering algorithm. Invisible 3D haptic containers with simplified geometry are used for passive interaction as well as for simulating haptic effects which do not have a clear visual manifestation. The results of this research novelty have been published in papers C1 and C8 of the list of publications that follows this chapter.
3. The set of novel methods enhancing the capabilities of the haptic device through physical and virtual constraints is proposed. Virtual constraints are used to simulate force feedback at multiple haptic interaction points using a single contact point haptic device. The degrees of freedom of the haptic device are altered by physical constraints to produce a set of DOFs that match the requirements of minimally invasive surgery. This has been published in papers C9 and C10 of the list of publications that follows this chapter.

In order to prove that the research goals have been achieved, several application examples were presented which have resulted in the following **development novelties**:

1. An image-driven arthroscopic surgery training simulator has been developed using tangible panoramic images created from real surgical videos. The images are displayed on a moving hemisphere in a virtual scene matching the dimensions of the actual surgical field. A few training exercises for acquisition of basic skills such as triangulation and grasping have been developed in collaboration with a practicing surgeon at Alexandra Hospital, Singapore. This has been published in papers J2, C5, and C11 of the list of publications that follows this chapter.
2. A desktop software tool with different modes for perception of individual features, such as edges and textures, is developed for haptic exploration of the traditional visual interfaces as well as any visual information displayed on the computer monitor. This has been published in papers J1 of the list of publications that follows this chapter.
3. A few other representative implementations and application examples have been presented. A software tool for haptic perception of the traditional visual interfaces, as well as any visual information displayed on the computer monitor on a Windows-based system, has been developed. The results of the PhD project have also contributed to the development of a few more medical applications based on the idea of tangible images, specifically: virtual acupuncture, virtual venipuncture, and virtual palpation.

The research and development results achieved have been published in **2 journal and 11 conference papers**, which are listed in the ‘List of Publications’ that follows this chapter. The research results contributed to the **completed externally funded project** “Visual and Haptic Rendering in Co-space” (NRF-MDA) and to the **ongoing externally funded project** “Collaborative Haptic Modeling for Orthopaedic Surgery Training in

Cyberspace”(MOE Tier 2). The developed arthroscopic surgery training simulator will be deployed at Alexandra Hospital, Singapore in near future.

8.2 Future work

During the course of these projects, several areas of future research have been identified. They are detailed in the following.

1. Automatic assignment of physical properties

One issue that needs further research is the association of material properties to different parts of the scene as perception of physical (material) properties of the objects is very important in the overall human touch experience. When using scenes composed of 3D models, each model can be defined as having specific physical properties. However, such a separation between objects is absent in a 2D image. Therefore, the image has to be segmented into regions of uniform physical properties. Moreover, material properties must be assigned to each region. In the thesis it was done interactively, as it is described in Section 4.1.3. While automatic image segmentation is a well-researched topic, development of an algorithm that assigns properties automatically is a challenging task because even in the real world, it is often not possible to judge the physical properties of an object just by looking at it. Different approaches can be researched. For example, a database of objects with associated physical properties can be created and algorithms to detect and match objects in a 2D image can be developed to obtain physical property information about a scene (Klank et al., 2009). Alternatively, techniques used in content-based image retrieval (CBIR) can be used to obtain material property information of one image with reference to similar images with known physical properties. This work

is currently being done as part of another Ph.D. project in collaboration with the author of the thesis.

2. Automatic generation of hybrid scenes

In this thesis we have manually created hybrid scenes comprising 3D objects augmented to an image by carefully placing the 3D objects so that they blend seamlessly into the background image both visually and haptically. While this is much faster than 3D modelling of the scene, it is still a time consuming process. Methods for automatic estimation of position and orientation of an object need to be researched and a scene representation format needs to be developed to store and exchange hybrid scenes in shared environments. This scene representation can have an additional file embedded to the image file, and can contain the required supplementary information required to generate feedback force. A tangible image player can be developed for such a file format.

3. Automatic stitching of arthroscopic images into a panoramic image

The image-driven virtual arthroscopy simulation requires creation of a panoramic image by stitching different frames of a real surgical video. An algorithm that uses computer vision techniques to achieve this automatically will further reduce turnaround time. With the emphasis of recent research on image features with desirable properties such as scale invariance, algorithms for automatic correspondence between arthroscopic images can be researched. Furthermore, tools can be developed to remove radial distortion in surgical images based on surgical camera specifications. Stitching of undistorted images could be more suitable for automatic stitching algorithms. This work is currently being done as a part of a Masters project in collaboration with the author of the thesis.

4. Depth compression

Depth information obtained from depth sensors such as Microsoft Kinect provides low resolution ground truth of the scene. While this generates reliable haptic interaction experience on a single computer, to be able to use it in shared environments such as video communication systems, compression algorithms need to be developed. A compression codec analogous to video compression codecs will ensure low bandwidth requirements and real-time interaction on devices connected to a network. This work is currently being done as part of another Ph.D. project in collaboration with the author of the thesis.

List of Publications

Journal papers

- J1. **RASOOL, S. & SOURIN, A.** (2014). Image-Driven Haptic Rendering. In Transactions on Computational Science XXIII, vol. 8490 eds. M. Gavrilova, C. J. K. Tan, X. Mao and L. Hong), pp. 58-77: Springer Berlin Heidelberg.
- J2. **RASOOL, S. & SOURIN, A.** (2013b). Image-driven virtual simulation of arthroscopy. The Visual Computer, 29(5), 333-344.

Conference papers

- C1. **RASOOL, S. & SOURIN, A.** (2010). Towards Tangible Images and Video in Cyberworlds--Function-Based Approach. In International Conference on Cyberworlds (CW), 2010, pp. 92-96, Singapore.
- C2. **RASOOL, S. & SOURIN, A.** (2011a). Haptic interaction with 2D images. In Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry, pp. 13-22. ACM, Hong Kong, China.
- C3. **RASOOL, S. & SOURIN, A.** (2011b). Tangible images. In SIGGRAPH Asia 2011 Sketches. ACM, Hong Kong, China.
- C4. **RASOOL, S. & SOURIN, A.** (2011c). Towards haptic interaction with images. In Proceedings of MEDIAS2011, pp. 75-82. ISBN 978-5-88835-032-4, Limassol, Cyprus.
- C5. **RASOOL, S. & SOURIN, A.** (2012a). Tangible arthroscopic images. In Proceedings of Computer Graphics International CGI 12, pp. 1-12. ISBN

978-1-85899-283-9, Bournemouth, UK.

- C6. **RASOOL, S. & SOURIN, A. (2012b).** Image-driven haptic interaction. Asia-Oceania Top University League (AUTOLE) Proceedings of Postgraduate Student Conference. pp. 124. Kuala Lumpur, Malaysia.
- C7. **RASOOL, S. & SOURIN, A. (2013a).** Image-Driven Haptic Rendering in Virtual Environments. In Cyberworlds (CW), 2013 International Conference on, pp. 286-293.
- C8. **RASOOL, S., SOURIN, A. & KAGDA, F. (2013a).** Image-driven haptic simulation of arthroscopic surgery. Studies in Health Technology and Informatics, 184, 337-343.
- C9. **RASOOL, S., SOURIN, A., XIA, P., WENG, B. & KAGDA, F. (2013b).** Towards hand-eye coordination training in virtual knee arthroscopy. In Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology, pp. 17-26. ACM, Singapore.
- C10. **RASOOL, S., SOURIN, A., PESTRIKOV, V. & KAGDA, F. (2014a).** Modeling arthroscopic camera with haptic devices in image-based virtual environments. In Haptics Symposium (HAPTICS), 2014 IEEE, pp. 403-408.
- C11. **RASOOL, S., SOURIN, A., PESTRIKOV, V. & KAGDA, F. (2014b).** Virtual Knee Arthroscopy Using Haptic Devices and Real Surgical Images. In Digital Human Modeling. Applications in Health, Safety, Ergonomics and Risk Management, vol. 8529 (ed. V. Duffy), pp. 436-447: Springer International Publishing.

References

- 3DSYSTEMS. (2013). 3D systems - Geomagic solutions: Haptic Devices and Toolkits. In http://www.geomagic.com/files/4213/6864/7095/01_23_13_Phantom_4pg_A4_ENGLISH_hi.pdf, pp. 1-4. 3DS Systems, Morrisville, USA.
- ACKERMAN, M. J. (1997). The Visible Human Project: a resource for anatomical visualization. In Information Technology Applications in Biomedicine, 1997. ITAB '97., Proceedings of the IEEE Engineering in Medicine and Biology Society Region 8 International Conference, pp. 29-31.
- ADI, W. & SULAIMAN, S. (2009). Haptic Texture Rendering Based on Visual Texture Information: A Study to Achieve Realistic Haptic Texture Rendering. In Visual Informatics: Bridging Research and Practice, vol. 5857/2009, pp. 279-287. Kuala Lumpur, Malaysia: Springer.
- ALEXA, M. & MATUSIK, W. (2010). Reliefs as images. ACM Trans. Graph., 29(4), 1-7.
- AUBRY, S., THOUVENIN, I., LENNE, D. & OLIVE, J. (2007). A knowledge model to read 3D annotations on a virtual mock-up for collaborative design. In 11th International Conference on Computer Supported Cooperative Work in Design, 2007. CSCWD 2007, pp. 669-674, Melbourne, Australia.
- BASDOGAN, C. (2007). From 2D images to 3D tangible models: autostereoscopic and haptic visualization of Martian rocks in virtual environments. Presence: Teleoper. Virtual Environ., 16(1), 1-15.
- BASDOGAN, C., HO, C.-H. & SRINIVASAN, M. A. (1997). A ray based haptic rendering technique for displaying shape and texture of 3D objects in virtual environments. Proceedings of ASME Dynamic Systems and Control Division (ASME 1997), 61, 77-84.
- BASDOGAN, C. & SRINIVASAN, M. (2002). Haptic rendering in virtual environments. Handbook of Virtual Environments, 117-134.

- BAYAKOVSKI, Y., LEVKOVICH-MASLYUK, L., IGNATENKO, A., KONUSHIN, A., TIMASOV, D., ZHIRKOV, A., MAHNJIN, H. & IN KYU, P. (2002). Depth image-based representations for static and animated 3D objects. In Proceedings. International Conference on Image Processing 2002, vol. 3, pp. III-25-III-28 vol.23, Rochester, USA.
- BAYONA, S., ESPADERO, J., PASTOR, L. & FERNANDEZ-ARROYO, J. M. (2003). A low-cost arthroscopy surgery training system. In The IASTED International Conference on Visualizatoin, Imaging and Image Processing (VIIP '03) (ed. M. H. Hamza). ACTA Press, Benalmadena, Spain.
- BOOTH, S., ANGELIS, F. F. & SCHMIDT-TJARKSEN, T. (2003). The influence of changing haptic refresh-rate on subjective user experiences - lessons for effective touch-based applications. In Eurohaptics, pp. 374-383, Dublin, Ireland.
- BUCK, S., MAES, F., D'HOORE, A. & SUETENS, P. (2007). Evaluation of a Novel Calibration Technique for Optically Tracked Oblique Laparoscopes. In Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007, vol. 4791 eds. N. Ayache, S. Ourselin and A. Maeder), pp. 467-474: Springer Berlin Heidelberg.
- BURDEA, G. C. (1996). Force and touch feedback for virtual reality. New York, USA: John Wiley & Sons Inc.
- CHA, J., EID, M. & EL SADDIK, A. (2008). DIBHR: Depth Image-Based Haptic Rendering. In Haptics: Perception, Devices and Scenarios, vol. 5024/2008, pp. 640-650. Madrid, Spain: Springer.
- CHA, J., EID, M. & SADDIK, A. E. (2009). Touchable 3D video system. ACM Trans. Multimedia Comput. Commun. Appl., TOMMCAP, 5(4), 1-25.
- CHA, J., KIM, S.-M., OAKLEY, I., RYU, J. & LEE, K. (2005a). Haptic Interaction with Depth Video Media. In Advances in Mulitmedia Information Processing - PCM 2005, vol. 3767/2005, pp. 420-430. Jeju Island, Korea: Springer.

- CHA, J., KIM, S.-Y., HO, Y.-S. & RYU, J. (2006). 3D video player system with haptic interaction based on depth image-based representation. *IEEE Transactions on Consumer Electronics*, 52(2), 477-484.
- CHA, J., RYU, J., KIM, S., EOM, S. & AHN, B. (2005b). Haptic Interaction in Realistic Multimedia Broadcasting. In *Advances in Multimedia Information Processing - PCM 2004*, vol. 3333/2005, pp. 482-490. Tokyo, Japan: Springer.
- CHA, J., SEO, Y., KIM, Y. & RYU, J. (2007). An Authoring/Editing Framework for Haptic Broadcasting: Passive Haptic Interactions using MPEG-4 BIFS. In *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2007*, pp. 274-279, Tsukuba, Japan.
- CHAN, L. S. H. & KUP-SZE, C. (2009). Integrating PhysX and OpenHaptics: Efficient force feedback generation using physics engine and haptic devices. In *Joint Conferences on Pervasive Computing (JCPC)*, 2009, pp. 853-858, Taipei, Taiwan.
- CHEN, L., SONG, Z. & FENG, L. (2004). Internet-enabled real-time collaborative assembly modeling via an e-Assembly system: status and promise. *Computer-Aided Design*, 36(9), 835-847.
- CHEN, T., ZHU, Z., SHAMIR, A., HU, S.-M. & COHEN-OR, D. (2013). 3-Sweep: extracting editable objects from a single photo. *ACM Trans. Graph.*, 32(6), 1-10.
- CHENYU, W. & JARAMAZ, B. (2008). An easy calibration for oblique-viewing endoscopes. *Ieee International Conference On Robotics And Automation. Proceedings*, 1424-1429.
- COLES, T. R., JOHN, N. W., SOFIA, G., GOULD, D. A. & CALDWELL, D. G. (2011a). Modification of commercial force feedback hardware for needle insertion simulation. *Stud Health Technol Inform*, 163, 135-137.
- COLES, T. R., MEGLAN, D. & JOHN, N. (2011b). The Role of Haptics in Medical Training Simulators: A Survey of the State of the Art. *Haptics, IEEE Transactions on*, 4(1), 51-66.

- CONTI, F., BADBAGLI, F., MORRIS, D. & SEWELL, C. (2005). CHAI 3D: An open source library for the rapid development of haptic scenes. In First Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environmen and Teleoperator Systems. IEEE Computer Society, Pisa, Italy.
- CYBERGLOVE. (2013). CyberGlove Systems LLC <http://www.cyberglovesystems.com>, vol. 2013.
- DE PASCALE, M. & PRATTICHIZZO, D. (2007). The Haptik Library. Robotics & Automation Magazine, IEEE, 14(4), 64-75.
- DINDAR, N., TEK LAP, A. M. & BASDOGAN, C. (2010). Immersive haptic interaction with media. In Visual Communications and Image Processing, vol. 7744. SPIE, Huangshan, China.
- DUROU, J.-D., FALCONE, M. & SAGONA, M. (2008). Numerical methods for shape-from-shading: A new survey with benchmarks. Computer Vision and Image Understanding, 109(1), 22-43.
- ECKER, A. & JEPSON, A. D. (2010). Polynomial shape from shading. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 145-152, San Francisco, USA.
- EL-FAR, N. R., GEORGANAS, N. D. & EL-SADDIK, A. (2007). A Collision Detection Algorithm for Point-Like Haptic Interactions in Highly Detailed Virtual Environments. In IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2007. VECIMS 2007, pp. 25-30, Ostuni, Italy.
- FERRE, M., CERRADA, P., BARRIO, J. & WIRZ, R. (2012). Haptic Rendering Methods for Multiple Contact Points. In Immersive Multimodal Interactive Presence eds. A. Peer and C. D. Giachritsis), pp. 157-169: Springer London.
- FORCE-DIMENSION. (2013). Force Dimension <http://www.forcedimension.com>, vol. 2013, Nyon, Switzerland.
- GARY, S. (2004). Haptic Painting. IEEE Computer Graphics and Applications, 24(6), 4-5.

- GAW, D., MORRIS, D. & SALISBURY, K. (2006). Haptically Annotated Movies: Reaching Out and Touching the Silver Screen. In 14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2006, pp. 287-288, Alexandria, USA.
- GEOMAGIC. (2013). Haptic devices from Geomagic <http://www.geomagic.com/>, vol. 2013.
- GIBSON, S., SAMOSKY, J., MOR, A., FYOCK, C., GRIMSON, E., KANADE, T., KIKINIS, R., LAUER, H., MCKENZIE, N., NAKAJIMA, S., OHKAMI, H., OSBORNE, R. & SAWADA, A. (1996). Simulating Arthroscopic Knee Surgery using Volumetric Object Representations, Real-Time Volume Rendering and Haptic Feedback. Joint Conference on Computer Vision and Virtual Reality in Medicine and Medical Robotics and Computer Assisted Surgery.
- GREGORY, A., LIN, M. C., GOTTSCHALK, S. & TAYLOR, R. (2005). A framework for fast and accurate collision detection for haptic interaction. In ACM SIGGRAPH 2005 Courses, pp. 34. ACM, Los Angeles, California.
- HAYWARD, V. & MACLEAN, K. E. (2007). Do it yourself haptics: part I. Robotics & Automation Magazine, IEEE, 14(4), 88-104.
- HENG, P. A., CHUN-YIU, C., TIEN-TSIN, W., YANGSHENG, X., YIM-PAN, C., KAI-MING, C. & SHIU-KIT, T. (2004). A virtual-reality training system for knee arthroscopic surgery. IEEE Transactions on Information Technology in Biomedicine, 8(2), 217-227.
- HO, C.-H., BASDOGAN, C. & SRINIVASAN, M. (1999). An efficient haptic rendering technique for displaying 3D polyhedral objects and their surface details in virtual environments. Presence: Teleoper. Virtual Environ., 8(5), 477-491.
- HOIEM, D., EFROS, A. A. & HEBERT, M. (2005). Automatic photo pop-up. In ACM SIGGRAPH 2005 Papers, pp. 577-584. ACM, Los Angeles, California.
- HUA, J. & HONG, Q. (2001). Haptic sculpting of volumetric implicit functions. In Computer Graphics and

- Applications, 2001. Proceedings. Ninth Pacific Conference on, pp. 254-264.
- HUANG, G. Q., MAK, K. L., SHEN, J. Y. & YAN, J. Q. (2001). Web-based product design review: implementation perspective. In 6th International Conference on Computer Supported Cooperative Work in Design 2001, pp. 261-266, London, Canada.
- IGLESIAS, R., PRAFA, E., URIBE, A., GARCIA-ALONSO, A., CASADO, S. & GUTIERREZ, T. (2007). Assembly Simulation on Collaborative Haptic Virtual Environments. In 15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision pp. 241-247.
- IKEI, Y., WAKAMATSU, K. & FUKUDA, S. (1998). Image data transformation for tactile texture display. In Proceedings of IEEE Virtual Reality Annual International Symposium, 1998, pp. 51-58, Atlanta, USA
- IMMERSION-CORPORATION. (2014). Immersion Corporation - Haptics for the Digital User Experience <http://www.immersion.com>, vol. 2014.
- INSIGHTARTHROVR. (2010). Virtual Reality Arthroscopy Trainer, pp. 1-12. GMV Innovating Solutions.
- ITKOWITZ, B., HANDLEY, J. & ZHU, W. (2005). The OpenHaptics toolkit: a library for adding 3D Touch, navigation and haptics to graphics applications. In Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint, pp. 590-591.
- JANABI-SHARIFI, F., HAYWARD, V. & WANG, Q. Y. (2007). Design and implementation of a graphic-haptic display system. Displays, 28(3), 118-128.
- JING, H. & HONG, Q. (2002). Dynamic Implicit Solids with Constraints for Haptic Sculpting. In International Conference of Shape Modeling and Applications SMI'02, vol. 0 (ed. Q. Hong), pp. 119-119, Banff, Canada.

- JOHNSON, D., THOMPSON, T. V., II, KAPLAN, M., NELSON, D. & COHEN, E. (1999). Painting textures with a haptic interface. In Virtual Reality, 1999. Proceedings., IEEE, pp. 282-285, Houston, USA.
- JUAN, W., AIGUO, S. & CHUIGUO, Z. (2007). A novel haptic texture display based on image processing. In IEEE International Conference on Robotics and Biomimetics, 2007. ROBIO 2007, pp. 1315-1320, Sanya, China.
- JUNGSIK, K., JANABI-SHARIFI, F. & JUNG, K. (2010). A Haptic Interaction Method Using Visual Information and Physically Based Modeling. *Mechatronics, IEEE/ASME Transactions on*, 15(4), 636-645.
- JUNGSIK, K., YOUNGJIN, K. & JUNG, K. (2007). Real-time haptic rendering of slowly deformable bodies based on two dimensional visual information for telemanipulation. In International Conference on Control, Automation and Systems, 2007. ICCAS '07, pp. 2448-2451, Seoul.
- KADLECEK, P. (2011). Overview of current developments in haptic APIs. In Proceedings of the 15th Central European Seminar on Computer Graphics CESC'11, Vinivne, Slovakia.
- KAGAWA, T., SHIMAMOTO, T., NISHINO, H. & UTSUMIYA, K. (2009). A Method of Image Edition with Haptic Devices. In International Conference on Complex, Intelligent and Software Intensive Systems, 2009. CISIS '09, pp. 1042-1047, Fukuoka, Japan.
- KAGAWA, T., SHIMAMOTO, T., NISHINO, H. & UTSUMIYA, K. (2010). A Study of Haptic Interaction for Image Edition Tools. In International Conference on Complex, Intelligent and Software Intensive Systems, vol. 0, pp. 1135-1140, Krakow, Poland.
- KAGAWA, T., SHIMAMURA, N., NISHINO, H. & UTSUMIYA, K. (2007). A method of haptic interaction for photo-retouch tools. In ISIC. IEEE International Conference on Systems, Man and Cybernetics, pp. 2443-2448, Montreal, Canada.

- KAPTCHUK, T. J. (2002). Acupuncture: Theory, Efficacy, and Practice. *Annals of Internal Medicine*, 136(5), 374-383.
- KIM, S.-C. & KWON, D.-S. (2013). Haptic interaction with objects in a picture based on pose estimation. *Multimedia Tools and Applications*, 1-22.
- KIM, S.-M., CHA, J., RYU, J. & LEE, K. (2006). Depth video enhancement for haptic interaction using smooth surface reconstruction. *IEICE Transactions on Information and Systems*, E89-D(1), 37-44.
- KLANK, U., ZIA, M. Z. & BEETZ, M. (2009). 3D model selection from an internet database for robotic vision. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 2406-2411.
- KLATZKY, R. L. & LEDERMAN, S. J. (2003). Touch. In *Experimental Psychology*, vol. 4 (ed. I. B. Weiner), pp. 147-176. New York: Wiley.
- KRUFKA, S. E., BARNER, K. E. & AYSAL, T. C. (2007). Visual to Tactile Conversion of Vector Graphics. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 15(2), 310-321.
- LAEHYUN, K., KYRIKOU, A., SUKHATME, G. S. & DESBRUN, M. (2002). An implicit-based haptic rendering technique. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3, pp. 2943-2948 vol.2943, Lausanne, Switzerland.
- LAEHYUN, K., SUKHATME, G. S. & DESBRUN, M. (2004). A haptic-rendering technique based on hybrid surface representation. *IEEE Computer Graphics and Applications*, 24(2), 66-75.
- LAREAU, D. & JOCHEN, L. (2012). Haptic rendering of photographs. In *Haptic Audio Visual Environments and Games (HAVE), 2012 IEEE International Workshop on*, pp. 107-112.
- LAREAU, D. & LANG, J. (2013). Instrument for Haptic Image Exploration. *Instrumentation and Measurement*, *IEEE Transactions on*, PP(99), 1-1.
- LECUYER, A., BURKHARDT, J.-M. & ETIENNE, L. (2004). Feeling bumps and holes without a haptic

- interface: the perception of pseudo-haptic textures. In Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 239-246. ACM, Vienna, Austria.
- LEVINSKI, K. (2007). Interactive function based shape modelling. Doctoral Thesis thesis, Nanyang Technological University.
- LI, J., SONG, A. & ZHANG, X. (2010a). Haptic Texture Rendering Using Single Texture Image. In Computational Intelligence and Design (ISCID), 2010 International Symposium on, vol. 1, pp. 7-10.
- LI, J., SONG, A. & ZHANG, X. (2010b). Image-based haptic texture rendering. In Proceedings of the 9th ACM SIGGRAPH Conference on Virtual-Reality Continuum and its Applications in Industry, pp. 237-242. ACM, Seoul, South Korea.
- MABREY, J., REINIG, K. & CANNON, W. (2010). Virtual Reality in Orthopaedics: Is It a Reality? Clinical Orthopaedics and Related Research®, 468(10), 2586-2591.
- MABREY, J. D., GILLOGLY, S. D., KASSER, J. R., SWEENEY, H. J., ZARINS, B., MEVIS, H., GARRETT, W. E., POSS, R. & CANNON, W. D. (2002). Virtual reality simulation of arthroscopy of the knee. Arthroscopy : the journal of arthroscopic & related surgery : official publication of the Arthroscopy Association of North America and the International Arthroscopy Association, 18(6), 28e.
- MACHULIS, K. (2009). libNiFalcon - open source driver for the Novint Falcon.
- MANN, F. (2000). Chapter I - From tradition to the twentieth century. In Reinventing Acupuncture (Second Edition) (ed. F. Mann), pp. 3-11. Edinburgh: Butterworth-Heinemann.
- MASSIE, T. & SALISBURG, K. (1994). The PHANToM Haptic Interface: A Device for Probing Virtual Objects. In Proceedings of the 1994 ASME International Mechanical Engineering Congress and Exhibition, vol. DSC 55-1, pp. 295-302.

- MCNEELY, W. A., PUTERBAUGH, K. D. & TROY, J. J. (1999). Six degree-of-freedom haptic rendering using voxel sampling. In Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pp. 401-408. ACM Press/Addison-Wesley Publishing Co., Los Angeles, USA.
- MEE YOUNG, S., KYUNGKOO, J., DONGJU, J., HWANMUN, L. & KIKWON, K. (2009). Touchable Video and Tactile Audio. In 11th IEEE International Symposium on Multimedia. ISM '09. , pp. 425-431, San Diego, USA.
- MEGALI, G., TONET, O., MAZZONI, M., DARIO, P., VASCELLARI, A. & MARCACCI, M. (2002). A New Tool for Surgical Training in Knee Arthroscopy. In Medical Image Computing and Computer-Assisted Intervention — MICCAI 2002, vol. 2489 eds. T. Dohi and R. Kikinis), pp. 170-177: Springer Berlin / Heidelberg.
- MELO, R., BARRETO, J. P. & FALCAO, G. (2012). A New Solution for Camera Calibration and Real-Time Image Distortion Correction in Medical Endoscopy;Initial Technical Evaluation. Biomedical Engineering, IEEE Transactions on, 59(3), 634-644.
- MING-YONG, P., YUN, S., SOURIN, A., CASTRO, G. G. & UGAIL, H. (2010). Automatic Reconstruction and Web Visualization of Complex PDE Shapes. In Cyberworlds (CW), 2010 International Conference on, pp. 97-104.
- MINSKY, M., MING, O.-Y., STEELE, O., FREDERICK P. BROOKS, J. & BEHENSKY, M. (1990). Feeling and seeing: issues in force display. SIGGRAPH Comput. Graph., 24(2), 235-241.
- MIRTICH, B. (1998). V-Clip: fast and robust polyhedral collision detection. ACM Transactions on Graphics (TOG), 17(3), 177-208.
- MOUSTAKAS, K., TZOVARAS, D. & STRINTZIS, M. G. (2007). SQ-Map: Efficient Layered Collision Detection and Haptic Rendering. IEEE Transactions on Visualization and Computer Graphics,

13(1), 80-93.

MPB-TECHNOLOGIES-INC. (2013). MPB 6 Degree of Freedom Haptic Interface www.mpb-technologies.ca/, vol. 2013.

MULLER, W. & BOCKHOLT, U. (1998). The Virtual Reality Athroscopy Training Simulator. In Medicine Meets Virtual Reality (MMVR) - Art, Science, Technology: Healthcare (R)Evolution (ed. J. D. Westwood), pp. 13-19, San Diego, USA.

NATIONAL-INSTITUTES-OF-HEALTH. (1997). Acupuncture. NIH Consens Statement, 15(5), 1-34.

NOVINT-TECHNOLOGIES-INC. (2013). Novint Technologies Inc. <http://www.novint.com>, vol. 2013, Alburguerque, USA.

O'MODHRRAIN, S. & OAKLEY, I. (2003). Touch TV: Adding Feeling to Broadcast Media. In Proceedings. European Conference on Interactive Television: from Viewers to Actors?, pp. 41-47, Brighton, UK.

O'MODHRRAIN, S. & OAKLEY, I. (2004). Adding interactivity: active touch in broadcast media. In Proceedings. 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS '04. , pp. 293-294, Chicago, USA.

OSORIO, N. A. (2008). Multipoint Collision Detection Algorithm for Bone Milling and Drilling in a Haptic VR Surgery Simulator MS thesis, Royal Institute of Technology.

PAI, D. K. & REISSELL, L. M. (1997). Haptic interaction with multiresolution image curves. Computers & Graphics, 21(4), 405-411.

PANEELS, S. & ROBERTS, J. C. (2010). Review of Designs for Haptic Data Visualization. IEEE Transactions on Haptics, 3(2), 119-137.

PESTRIKOV, V. & SOURIN, A. (2013). Towards Making Panoramic Images in Virtual Arthroscopy. In International Conference on Cyberworlds, pp. 48-51. IEEE, Yokohama, Japan.

- PETERSIK, A., PFLESSER, B., TIEDE, U., H\, K.-H., \#246, HNE & LEUWER, R. (2003). Realistic haptic interaction in volume sculpting for surgery simulation. In Proceedings of the international conference on Surgery simulation and soft tissue modeling, pp. 194-202. Springer-Verlag, Juan-Les-Pins, France.
- PING-SING, T. & SHAH, M. (1994). Shape from shading using linear approximation. Image and Vision Computing, 12(8), 487-498.
- QI, L. & SOURIN, A. (2006a). Function-based shape modeling extension of the Virtual Reality Modeling Language. Computers & Graphics, 30(4), 629-645.
- QI, L. & SOURIN, A. (2006b). Function-defined shape metamorphoses in visual cyberworlds. The Visual Computer, 22(12), 977-990.
- QUANSER-INC. (2014). Quanser - Haptic Applications. <http://www.quanser.com/Products/haptics>, vol. 2014.
- RASOOL, S., SOURIN, A., XIA, P., WENG, B. & KAGDA, F. (2013). Towards hand-eye coordination training in virtual knee arthroscopy. In Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology, pp. 17-26. ACM, Singapore.
- RAYMAEKERS, C., BEETS, K. & VAN REETH, F. (2001). Fast Haptic Rendering of Complex Objects Using Subdivision Surfaces. In Proceedings of the sixth PHANToM Users Group Workshop, Aspen, USA.
- REACHIN-TECHNOLOGIES-AB. (2013). Reachin - The leading haptic software solution company <http://www.reachin.se>, vol. 2013, Stockholm, Sweden.
- REINKENSMEYER, D. J., PANG, C. T., NESSLER, J. A. & PAINTER, C. C. (2001). Java Therapy: Web-based Robotic Rehabilitation. In 7th International Conference on Rehabilitation Robotics, pp. 66-71, Evry, France.

- RUSPINI, D. & KHATIB, O. (2000). A framework for multi-contact multi-body dynamic simulation and haptic display. In Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on, vol. 2, pp. 1322-1327 vol.1322.
- RUSPINI, D. C., KOLAROV, K. & KHATIB, O. (1997). The haptic display of complex graphical environments. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp. 345-352. ACM Press/Addison-Wesley Publishing Co.
- RYDEN, F., CHEZECK, H. J., KOSARI, S. N., KING, H. & HANNAFORD, B. (2011). Using Kinect and a haptic interface for implementation of Real-time virtual fixtures. In 2nd Workshop on RGB-D: Advanced Reasoning with Depth Cameras (in conjunction with RSS 2011), Los Angeles, USA.
- SALISBURY, K., BROCK, D., MASSIE, T., SWARUP, N. & ZILLES, C. (1995). Haptic rendering: programming touch interaction with virtual objects. In Proceedings of the 1995 symposium on Interactive 3D graphics, pp. 123-130. ACM, Monterey, California, United States.
- SALISBURY, K., CONTI, F. & BARBAGLI, F. (2004). Haptic rendering: introductory concepts. Computer Graphics and Applications, IEEE, 24(2), 24-32.
- SALISBURY, K. & TARR, C. (1997). Haptic rendering of surfaces defined by implicit functions. In Proceedings of the ASME 6th Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 61-68, Dallas, TX.
- SCENEGRAPHICS-AB. (2013). Open Source Haptics <http://www.h3dapi.org/>, vol. 2013.
- SCHOEMER, E., REICHEL, J. & WARKEN, T. (2002). Efficient Collision Detection for Curved Solid Objects. In Proceedings of the seventh ACM symposium on Solid modeling and applications, pp. 321-328. ACM, Saarbruecken, Germany.
- SEUNG-CHAN, K., BYUNG-KIL, H., JEONG-YEAN, Y. & DONG-SOO, K. (2010). Interaction with objects inside a media space. In ACM SIGGRAPH ASIA 2010 Posters, pp. 1-1. ACM, Seoul,

Republic of Korea.

SEUNG-CHAN, K., KYUNG, K.-U. & KWON, D. S. (2011). Haptic annotation for an interactive image.

In 5th ACM International Conference on Ubiquitous Information Management and Communication ICUIMC '11. ACM, Seoul, Korea.

SHENG, Y., SOURIN, A., CASTRO, G. & UGAIL, H. (2010). A PDE method for patchwise approximation of large polygon meshes. *The Visual Computer*, 26(6), 975-984.

SHERMAN, K. P., WARD, J. W., WILLS, D. P. & MOHSEN, A. M. (1999). A portable virtual environment knee arthroscopy training system with objective scoring. *Studies in Health Technologies and Informatics*, 62, 335-336.

SHI, Y. & PAI, D. K. (1997). Haptic display of visual images. In *IEEE Virtual Reality Annual International Symposium*, 1997, pp. 188-191, Albuquerque, USA.

SIDHARTA, R., OLIVER, J. & SANNIER, A. (2006). Augmented Reality Tangible Interface for Distributed Design Review. In *Computer Graphics, Imaging and Visualisation, 2006 International Conference on*, pp. 464-470, Sydney, Australia.

SIIRA, J. & PAI, D. K. (1996). Haptic texturing-a stochastic approach. In *Proceedings of IEEE International Conference on Robotics and Automation*, 1996, vol. 1, pp. 557-562 vol.551, Minneapolis, USA.

SIMBIONIX. (2012). Arthro Mentor - VR training simulator for knee and shoulder arthroscopic procedures. Symbionix Corporation.

SINGAPOGU, R. B., SANDER, S. T., BURG, T. C. & DONGBIN, L. (2008). A Five DOF haptic rendering algorithm using multiple contact points. In *IEEE Southeastcon*, pp. 262-267.

SIU-YUENG, C. & CHOW, T. W. S. (2001). Neural computation approach for developing a 3D shape reconstruction model. *IEEE Transactions on Neural Networks*, 12(5), 1204-1214.

- SOURIN, A., SOURINA, O., WEI, L. & GAGNON, P. (2009). Visual Immersive Haptic Mathematics in Shared Virtual Spaces. In Transactions on Computational Science III. LNCS5300, vol. 5300/2009, pp. 1-19: Springer.
- SOURIN, A. & WEI, L. (2008). Visual immersive haptic rendering on the web. In Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry, pp. 1-6. ACM, Singapore.
- SOURIN, A. & WEI, L. (2009). Visual immersive haptic mathematics. *Virtual Reality*, 13(4), 221-234.
- SRINIVASAN, M. A. & BASDOGAN, C. (1997). Haptics in virtual environments: Taxonomy, research status, and challenges. *Computers & Graphics*, 21(4), 393-404.
- SRIVASTAVA, M. (2007). Implementation and Evaluation of a Multiple-Points Haptic Rendering Algorithm, Ohio University.
- THOMAS V. THOMPSON, I. & COHEN, E. (2005). Direct haptic rendering of complex trimmed NURBS models. In *ACM SIGGRAPH 2005 Courses*, pp. 89. ACM, Los Angeles, California.
- THOMAS V. THOMPSON, I., JOHNSON, D. E. & COHEN, E. (1997). Direct haptic rendering of sculptured models. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pp. 167-176. ACM, Providence, Rhode Island, USA.
- THOUVENIN, I. (2005). Knowledge integration in early design stages for collaboration on a virtual mock up. In *9th International Conference on Computer Supported Cooperative Work in Design*, vol. 2 (ed. D. Lenne, A. Guenand and S. Aubry), pp. 1141-1145 Vol. 1142, Coventry, UK.
- TSAI, R. Y. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4), 323-344.
- VAN DER LINDE, R. Q. (2003). HapticMaster - a generic force controlled robot for human interaction.

Industrial Robot, 30(6), 515-524.

VASUDEVAN, H. & MANIVANNAN, M. (2006). Recordable Haptic textures. In Haptic Audio Visual Environments and their Applications, 2006. HAVE 2006. IEEE International Workshop on, pp. 130-133.

VASUDEVAN, H. & MANIVANNAN, M. (2008). Tangible Images: Runtime Generation of Haptic Textures From Images. In Proceedings of the 2008 Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 357-360. IEEE Computer Society, Reno, USA.

VEMBAR, D., DUCHOWSKI, A., PAUL, M., GRAMOPADHYE, A. & WASHBURN, C. (2010). Visuohaptic borescope inspection simulation training: modeling multi-point collision detection/response and evaluating skills transfer. In Proceedings of Graphics Interface 2010, pp. 177-184. Canadian Information Processing Society, Ottawa, Ontario, Canada.

VIRTAMED. (2012). VirtaMed ArthroS - Virtual Reality training for arthroscopy. In VirtaMed AG.

WALKER, S. P. & SALISBURY, J. K. (2003). Large haptic topographic maps: marsview and the proxy graph algorithm. In Proceedings of the 2003 symposium on Interactive 3D graphics, pp. 83-92. ACM, Monterey, California.

WANG, Q. (1999). Translation of graphics to haptic boundary representation. Electronic Thesis or Dissertation thesis, McGill University.

WAY, T. P. & BARNER, K. E. (1997). Automatic visual to tactile translation. I. Human factors, access methods and image manipulation. IEEE Transactions on Rehabilitation Engineering, 5(1), 81-94.

WEI, L. (2011). Function-based visual and haptic rendering, interaction and collaboration in shared virtual spaces. PhD Thesis thesis, Nanyang Technological University.

WEI, L., SOURIN, A. & SOURINA, O. (2008). Function-based visualization and haptic rendering in shared virtual spaces. The Visual Computer, 24(10), 871-880.

- WEI, L., SOURIN, A. & STOCKER, H. (2009). Function-based haptic collaboration in X3D. In Proceedings of the 14th International Conference on 3D Web Technology (Web3D 09), pp. 15-23. ACM Press, Darmstadt, Germany.
- XIA, P. & SOURIN, A. (2012). Design and implementation of a haptics-based virtual venepuncture simulation and training system. In Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry, pp. 25-30. ACM, Singapore, Singapore.
- XIANGYU, W. (2006). Usability Evaluation of a Mixed Reality Collaborative Tool for Design Review. In International Conference of Computer Graphics, Imaging and Visualisation CGIV'06, vol. 0 (ed. S. D. Phillip), pp. 448-451, Sydney, Australia.
- XU, S., LI, C., HU, L., JIANG, S. & LIU, X. P. (2012). An improved switching vector median filter for image-based haptic texture generation. In Image and Signal Processing (CISP), 2012 5th International Congress on, pp. 1195-1199.
- YAMAGUCHI, T., AKABANE, A., MURAYAMA, J. & SATO, M. (2006). Automatic generation of haptic effect into published 2D graphics. In Proceedings of Eurohaptics Conference 2006, pp. 617-620, Paris, France.
- YAMAGUCHI, T., NAKAMOTO, M., SATO, Y., NAKAJIMA, Y., KONISHI, K., HASHIZUME, M., NISHII, T., SUGANO, N., YOSHIKAWA, H., YONENOBU, K. & TAMURA, S. (2003). Camera Model and Calibration Procedure for Oblique-Viewing Endoscope. In Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003, vol. 2879 eds. R. Ellis and T. Peters), pp. 373-381: Springer Berlin Heidelberg.
- YASMIN, S. & SOURIN, A. (2012). Virtual Palpation for Medical Training in Cyberworlds. In Cyberworlds (CW), 2012 International Conference on, pp. 207-214.

- YASMIN, S. & SOURIN, A. (2013). Image-Based Virtual Palpation. In Transactions on Computational Science XVIII, vol. 7848 eds. M. Gavrilova, C. J. K. Tan and A. Kuijper), pp. 61-80: Springer Berlin Heidelberg.
- ZHANG, R., TSAI, P.-S., CRYER, J. E. & SHAH, M. (1999). Shape from Shading: A Survey. IEEE Trans. Pattern Anal. Mach. Intell., 21(8), 690-706.
- ZHU, W. & LEE, Y.-S. (2004). Product prototyping and manufacturing planning with 5-DOF haptic sculpting and dixel volume updating. In Proceedings of the 12th international conference on Haptic interfaces for virtual environment and teleoperator systems, pp. 98-105. IEEE Computer Society, Chicago, USA.

APPENDIX A: URLs of Supplementary Videos

Figure 4.1 (a) A 2D image with brightness reflecting depth of the scene (b) corresponding heightmap (c) surface felt by the haptic device (d) content of the mask at the position indicated in (a).
<http://www.youtube.com/watch?v=8SMbvIdmuTk>

Figure 4.5 Depth-based haptic rendering algorithm applied to depth stream obtained by Microsoft Kinect. Arrows show the force direction at different parts of the scene.
<http://www.youtube.com/watch?v=LwJjPuXUDhY>

Figure 4.6 Haptic rendering of moving objects with depth stream obtained using Microsoft Kinect.
<http://www.youtube.com/watch?v=C2WeVLbYDbs>

Figure 4.11 Handling non-geometric intensity discontinuities (a) test image in normal (default) mode, (b) test image in the offset mode. http://www.youtube.com/watch?v=0OWBFvTNP_I

Figure 4.13 An edge with width Δ , an associated normal n and a function defined edge profile.
<http://www.youtube.com/watch?v=BCzVIW4c9Os>

Figure 4.14 Movement of an edge (a) the edge is detected at $t=0$, (b) the edge moves from left to right, (c) the edge moves from right to left. <http://www.youtube.com/watch?v=zMReWQmTX30>

Figure 4.15 Texture emphasis mode: (a) test image (b) the histogram generated by the developed application (c) the perceived surface before and after texture emphasis. http://www.youtube.com/watch?v=EWSn7qn_FHg

Figure 5.4 Meniscus modeling using implicit functions. (a) Summation of inverse square functions, (b) Relation of the modeled part of the meniscus to the complete meniscus along with different views of the modeled part. (c) Meniscus removal using set theoretic operations. <http://www.youtube.com/watch?v=A0EYKC1yb8Q>

Figure 5.7 (a) A snapshot from a whirlpool video which is superimposed by the container in (b) at appropriate orientation, and the clockwise force effect on the haptic device, (b) A helical force field inside a transparent cylindrical container. <http://www.youtube.com/watch?v=GUvWH8FIKAw>

Figure 7.5 Haptic rendering of physical properties, (a) actual image with arrows showing direction of rendered force, (b) segmentation map. <http://www.youtube.com/watch?v=deqO8ggqhDk>

Figure 7.6 Geomagic Touch haptic devices used for simulation of minimally invasive surgery. <http://www.youtube.com/watch?v=UQbcFOhn6hA>

Figure 7.13

Six point pyramid used for grasping.

<http://www.youtube.com/watch?v=Ksa2RjXk1Zw>

Figure 7.15

Screenshots of acupuncture simulation (a) insertion of needle at incorrect point (b) insertion of needle at correct point.

<http://www.youtube.com/watch?v=GCs3sHUyqPk>

Figure 7.17

A virtual palm palpating a malignant and enlarged liver.

Notice the deformation near the belly button in the left image (Yasmin & Sourin, 2013).

<http://www.youtube.com/watch?v=tn5rwhw73PE>

Figure 7.18

Image-driven venipuncture simulation (a) image and the function-defined 3D model of the veins, (b) screenshot of the implemented system.

<http://www.youtube.com/watch?v=-ISlcU0vKcw>

Figure 7.20

Use of desktop tangible image application for haptic interaction with graphical user interfaces in contour perception mode. The direction of the force at the current location of the HIP is shown as an arrow.

<http://www.youtube.com/watch?v=5-JY-eE-Jw0>

Figure 7.23

Interaction with the moving object in haptic video communication. The handle of the haptic device moves up and down following the motion of the hand in the Skype window. <http://www.youtube.com/watch?v=zMReWQmTX30>

APPENDIX B: Point-Plane Distance

Given a plane $ax + by + cz + d = 0$ with normal vector $\vec{n} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ and a point $\vec{p} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}$.

A vector from the plane to the point $\vec{p}$ is given by $\vec{v} = -\begin{bmatrix} x - p_1 \\ y - p_2 \\ z - p_3 \end{bmatrix}$.

The distance D between point $\vec{p}$ and the plane is obtained by projecting the vector $\vec{v}$ onto the normal $\vec{n}$ as

$$\begin{aligned} D &= \|\text{proj}_{\vec{n}} \vec{v}\| \\ &= \frac{\|\vec{n} \cdot \vec{v}\|}{\|\vec{n}\|} \\ &= \frac{\|a(x - p_1) + b(y - p_2) + c(z - p_3)\|}{\sqrt{a^2 + b^2 + c^2}} \\ &= \frac{\|ax + by + cz - ap_1 - bp_2 - cp_3\|}{\sqrt{a^2 + b^2 + c^2}} \\ &= \frac{\|-d - ap_1 - bp_2 - cp_3\|}{\sqrt{a^2 + b^2 + c^2}} \\ &= \frac{\|ap_1 + bp_2 + cp_3 + d\|}{\sqrt{a^2 + b^2 + c^2}} \end{aligned}$$

The signed distance is then given by

$$D = \frac{ap_1 + bp_2 + cp_3 + d}{\sqrt{a^2 + b^2 + c^2}}$$

In Hessian normal form, this can be written as

$$D = \hat{n} \cdot \vec{p} + p$$

where $\hat{n}$ is the unit normal.

Given three points in the plane $\vec{q}_i$ for $i = 1, 2, 3$ a unit normal is given by

$$\hat{n} = \frac{(\vec{q}_2 - \vec{q}_1)(\vec{q}_3 - \vec{q}_1)}{\|(\vec{q}_2 - \vec{q}_1)(\vec{q}_3 - \vec{q}_1)\|}$$

Then the signed distance from a point $\vec{p}$ to the plane containing the three points is

$$D = \hat{n} \cdot (\vec{p} - \vec{q}_i)$$

where $\vec{q}_i$ is any of the three points.