Image-inspired haptic interaction

Zhang, Xingzi

2018

Zhang, X. (2018). Image-inspired haptic interaction. Doctoral thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/75884>

<https://doi.org/10.32657/10356/75884>

(On the Spine)

IMAGE-INSPIRED HAPTIC INTERACTION

ZHANG XINGZI

2018

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

IMAGE-INSPIRED HAPTIC INTERACTION

ZHANG XINGZI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2018

ZHANG XINGZI

IMAGE-INSPIRED HAPTIC INTERACTION

ZHANG XINGZI

School of Computer Science and Engineering

A thesis submitted to
Nanyang Technological University & Technische Universität Darmstadt
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2018

ABSTRACT

In this research, an image-inspired haptic interaction approach is proposed where the problem of augmenting images with 3D haptic models is addressed in a systematical way. It is hypothesized that images of real scenes can be converted into a tangible form with the proposed approach so that one could interact with them using desktop haptic devices as if they were real 3D scenes.

To achieve this goal, two different scenarios are considered when augmenting images with haptic models. First, in the case when there is no available haptic model for the image, a method of sketching and augmenting the images with perspectively distorted haptic models is proposed. The research novelty of this method lies in the ways of creating distorted haptic models and aligning them with the images. In contrast to common 3D modelling, haptic models are defined in perspectively distorted modelling space, so that when projected with orthographic projection their contours would match the respective parts of the image. In the proposed method, function representations (FReps) are mostly used to define the augmented haptic models, which makes this method highly expandable – any type of function-based models, as long as it matches the image content, can be added to this method. Besides, the novelty is also in the haptic rendering algorithm proposed for rendering function-based models, which ensures smooth interaction with a combination of models defined in FReps. The use of FReps allows us to minimize the model size while still keeping the rendering precision, and makes the method specifically suitable for web-based real-time applications.

Next, the case where the image scene can be reconstructed is considered. Given multiple images of a scene, the 3D model of the image can be obtained with computer vision

techniques. Based on this scenario, a way of haptic interaction with real life scenes is proposed where multiple original images of the real scenes are augmented with the reconstructed polygon mesh. The research novelty here lies in the solution to the problem of interactive haptic rendering of large polygon meshes with reconstruction artifacts. In particular, the presented collision detection algorithm is robust enough to support smooth interaction with millions of polygons.

To prove the hypothesis and investigate the feasibility and acceptance of the proposed approach, a usability test was conducted with the tangible image interface. Besides, a potential application of image-inspired haptic interaction—tangible online shopping interface—was investigated. The results of these studies have shown that the users' attitude towards the proposed ideas is between neutral and largely positive, which leads to the conclusion that the hypothesis can be accepted under both two cases. Thus it can be concluded that the proposed image-inspired haptic interaction approach brings new possibilities to interaction with images and provides a way to enrich images with a touch interface and haptic feedback.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor Associate Professor Alexei Sourin and my co-supervisor Professor Michael Goesele for their invaluable guidance in the past five years. I am grateful for their patience in the supervision and this thesis is the result of joint efforts.

I also would like to express my gratitude to the School of Computer Science and Engineering and Nanyang Technological University for providing me the opportunity and financial support to further my study and research. Special thanks go to Joint PhD Degree Program NTU-TU Darmstadt as well as to Fraunhofer Singapore for supporting my research.

Last but not the least, I would like to thank all my present and past lab mates, especially Dr. Shahzad Rasool, Weng Bin and Cui Jian, for their inspiration and support during my postgraduate study. Without their help, my research could not have reached this point. Furthermore, I wish to thank my parents. Their selfless concern and constant encouragement has always been my biggest support.

TABLE OF CONTENTS

ABSTRACT.....	I
ACKNOWLEDGEMENTS.....	III
TABLE OF CONTENTS.....	IV
LIST OF FIGURES	VII
LIST OF TABLES.....	X
Chapter 1 Introduction	1
1.1 Motivation.....	1
1.2 Proposed approach	3
1.3 Report Organization.....	9
Chapter 2 Adding Haptic Modality to Images	10
2.1 Common haptic interaction.....	10
2.1.1 Haptic devices	11
2.1.2 Haptic rendering algorithms.....	15
2.2 Haptic interaction with images	22
2.2.1 Computing haptic response from images.....	23
2.2.2 Augmenting images with haptic data.....	26
2.2.3 Critical analysis and conclusion.....	31
Chapter 3 Interactive Augmenting Images with Haptic Models.....	34
3.1 Overview.....	35
3.2 Sketching haptic models	36
3.2.1 Sketching primitives	39
3.2.2 Sketching complex geometries	45
3.3 Projecting haptic models.....	53
3.4 Haptic rendering of the augmented scene	54
3.4.1 Collision detection and proxy calculation.....	54
3.4.2 Force rendering	56
3.5 Haptic file.....	59
3.6 Results.....	61
3.7 Summary.....	66
Chapter 4 Augmenting Images with Reconstructed Haptic Models	67
4.1 Overview.....	68

4.2	Coordinate transformation	70
4.3	Preprocessing of the reconstructed mesh	74
4.4	Collision detection with the preprocessed meshes	76
4.4.1	Intersection test.....	78
4.4.2	Tracking of the active primitive	80
4.5	Force rendering	82
4.6	Results.....	83
4.7	Comparison experiments.....	85
4.8	Summary	89
Chapter 5 Integrating Haptic Perception of Material Properties and Visual Feedback into Haptic Interaction		91
5.1	Adding haptic perception of material properties.....	92
5.1.1	Feeling the texture.....	92
5.1.2	Haptic simulation of elasticity and friction.....	95
5.1.3	Weight simulation	97
5.1.4	Application examples and discussion	100
5.2	Adding visual feedback based on haptic input.....	101
5.2.1	Visual rendering of the haptic cursor	102
5.2.2	Interactive image distortion.....	104
5.3	Summary	105
Chapter 6 User Studies.....		106
6.1	User study on tangible image.....	106
6.1.1	Capturing Test Photos	106
6.1.2	Experimental Setup	107
6.1.3	Experimental design.....	108
6.1.4	Results.....	110
6.1.5	Evaluation	111
6.1.6	Conclusion	114
6.2	Tangible online shopping application	115
6.2.1	Hypothesis.....	116
6.2.2	User interface design.....	117
6.2.3	Experimental environment setup.....	119
6.2.4	Experiment design.....	119

6.2.5	Results.....	121
6.2.6	Evaluation and discussion.....	123
6.2.7	Conclusion	126
Chapter 7 Conclusion and Future Work		128
7.1	Conclusion	128
7.2	Future Work	130
7.2.1	Automatic template matching for single-view reconstruction	130
7.2.2	Automatic estimation of camera parameters from single image	131
7.2.3	Physically-based haptic rendering of deformable objects.....	132
Appendices.....		134
Reference		136

LIST OF FIGURES

Figure 1-1	A visual-haptic interactive scene with the statue of the ram of Amun.....	1
Figure 1-2	Image-based virtual street walkthroughs in Nanyang Technological University through Google Earth	2
Figure 1-3	Illustration of the image-inspired haptic interaction approach.	5
Figure 1-4	The general pipeline of the proposed image-inspired haptic interaction approach.	6
Figure 2-1	The basic structure of a haptic interaction system.....	10
Figure 2-2	Geomagic haptic devices and Novint Falcon.	14
Figure 2-3	The simplified pipeline of haptic rendering for impedance controlled haptic devices.	16
Figure 2-4	Illustration of point-based haptic interaction.....	18
Figure 2-5	Illustration of the haptic rendering algorithm proposed by Salisbury and Tarr [32]..	21
Figure 3-1	The pipeline of the proposed image-inspired haptic interaction approach where the haptic models are interactively defined by user sketch.	36
Figure 3-2	Location of haptic objects with reference to the image plane.	38
Figure 3-3	The 2D sketch and the corresponding 3D model..	43
Figure 3-4	Projection of the circle on the image plane.	43
Figure 3-5	Illustration of shearing an infinite cone.....	45
Figure 3-6	Illustration of the stroke-based sketching of the truncated cone.	45
Figure 3-7	Rotation of a truncated cone towards the image plane with the rotation axis defined on the ground plane.....	47
Figure 3-8	(a) Stroke-based sketching of the plane-bounded solid object.	

	(b) Placement of the FRep solid in front of the image plane.	49
Figure 3-9	Images with sketch input and the corresponding haptic models displayed using the interactive FVRML software tool [96].	51
Figure 3-10	An example to illustrate the projection of the haptic models constructed from the strokes.	54
Figure 3-11	An ambiguous situation for proxy calculation.	55
Figure 3-12	Feedback force model where f_v is optional.	58
Figure 3-13	An example of a “haptic” file.	61
Figure 3-14	(a) Sketching on the image. (b) The haptic model built based on the sketch. (c) Aligning the model with the image. (d) A snapshot of the interactive scene.	62
Figure 3-15	Haptic interaction with images augmented with haptic objects.	63
Figure 3-16	The conference poster is converted to tangible image touched with Geomagic Touch.	65
Figure 4-1	The tangible image pipeline with models generated from MVE [98] as an example.	68
Figure 4-2	Flowchart of the mapping process.	72
Figure 4-3	Examples of alignment results.	73
Figure 4-4	An illustrative example for neighbor definitions.	75
Figure 4-5	Flowchart of collision detection process.	77
Figure 4-6	An example to illustrate how to find all the cells intersected with ray P_0P_1 in the following order: $a \rightarrow c \rightarrow b \rightarrow e \rightarrow d$	79
Figure 4-7	An example to illustrate difference between intersection and collision.	80
Figure 4-8	(a) The original reconstructed model. (b) The transformed model displayed in simulated camera frustum. (c) The alignment of the transformed model and the image. (d) A snapshot of the interactive scene.	84

Figure 4-9	Examples of interactions with the models reconstructed from images.	85
Figure 4-10	The image is displayed using OpenGL in the comparison experiment	86
Figure 4-11	(a) An image with which the proposed algorithm fails due to the undesired alignment. (b) The corresponding model.....	88
Figure 5-1	A haptic interaction comparison between the proposed image-inspired sketch-based method and the polygon-based interaction approach.	94
Figure 5-2	Directions of normal force, texture force and friction force.....	95
Figure 5-3	Scale position and handle placement setup during the measurement.....	98
Figure 5-4	Line charts of expected output, raw measurements and measurements after adjustment.	100
Figure 5-5	Illustration of occlusion.....	103
Figure 5-6	Interactively generate deformed images to simulate animation..	105
Figure 6-1	Top row: photos taken from high viewpoints. Middle row: photos taken from normal eye-level viewpoints. Bottom row: photos taken from low viewpoints.	107
Figure 6-2	A beta test participant interacts with the tangible photos.....	108
Figure 6-3	Example of sliding from far surface to near surface.	112
Figure 6-4	Screenshots of mouse-based and haptic GUIs.....	118
Figure 6-5	Environment setup for the experimental group.	119

LIST OF TABLES

Table 2-1	Highlighted specifications and sales price for Geomagic Touch, Geomagic Touch X and Novint Falcon.....	14
Table 3-1	Stroke patterns for defining primitives.....	40
Table 3-2	Parameters for defining haptic primitives.....	60
Table 4-1	Results of the performance comparison.	87
Table 6-1	Questions and corresponding factors.....	109
Table 6-2	Results of the questionnaire.....	111
Table 6-3	Questionnaire for both two groups..	120
Table 6-4	Average ratings of the two groups for the questionnaire and the t-test results.	121

Chapter 1 Introduction

1.1 Motivation

Haptic technology, or haptics, is an interaction feedback technology based on applying forces, vibrations, and/or motions to the user. Compared with visual rendering, which runs at 30-60 Hz, smooth haptic interaction requires 1000 Hz refresh rate. Usually, haptic interaction is considered with 3D objects defined by polygons. A simple haptic interaction pipeline can use the same polygon mesh for both visual and haptic interaction. However, when it comes to a simulated real-life scene which contains a large number of polygons, stable visual-haptic interaction becomes a problem.

Figure 1-1: A visual-haptic interactive scene with the statue of the ram of Amun. The users touch the statue by controlling the position of the virtual “hand” through the haptic interface. If the digital replica of the statue is directly used for visual display and haptic interaction, the graphics rendering time would be too long, which results in clumsiness in the displayed “hand” movement or unexpected pop-throughs. Statue of the ram of Amun by The British Museum is licensed under CC Attribution-NonCommercial-ShareAlike.

Given a simulated real-life model, e.g., the digital replica of the ram of Amun statue in Figure 1-1, even if this model could be viewed and “touched” through a haptic interface,

it cannot be simultaneously visualized and haptically rendered due to the big discrepancy between the visual (30-60 Hz) and haptic (1 kHz) refresh rates. While the virtual “hand” position is computed by the CPU (together with other haptic rendering tasks) at haptic refresh rate, samples of “hand” positions are read and displayed at graphics refresh rate. Therefore, if the average time for updating the whole interactive scene (including the model and the virtual “hand”) is too long, the movement of the virtual “hand” would be clumsy and there might be unexpected pop-throughs. To solve this problem, it is needed to reduce the graphics rendering time for visual models.

Figure 1-2: Image-based virtual street walkthroughs in Nanyang Technological University through Google Earth. By clicking in the image scene, we are able to mark our walking direction, change our viewpoint and zoom in/out.

To render a photorealistic scene in real time, an alternative is to replace the actual 3D scenes with their images. This approach is actively used in image-based real-time applications such as interactive panoramas, virtual street walkthroughs and image-based online shopping. The advantages of image-based visualization are that it naturally achieves photorealism and can effectively reduce the loading and rendering time. One of its most popular applications is Google Earth. With a large number of images used, one can make virtual street walkthroughs in the image scenes by choosing the walking path, zooming in/out and rotating the view. Figure 1-2 shows how this works. Based on this, it is assumed that replacement of the simulated visual-haptic scenes with their “tangible images” can be

an alternative solution to reduce the graphics rendering time and achieve real-time haptic interaction with complex scenes. In the scenario of “tangible images”, images of the real-life scenes are displayed for visual feedback and three-dimensional haptic feedback is added to the images based on their content so that they can be felt with haptic devices as if they were real 3D scenes.

1.2 Proposed approach

To achieve photorealistic visual-haptic interaction in real time, it was previously proposed to use images, mostly photos, for visual rendering while a new “touching” modality is added into the image scenes to complement the interaction experience. A brief introduction of these methods is presented below.

Digital images are usually made tangible in the following three ways. Firstly, the haptic forces can be derived directly from the image by analyzing pixel information [1, 2]. These techniques assume that the scene geometry information can be derived from the pixels, and forces are generated based on the assumptions. In this manner, little computation is required for the interaction, but these techniques are constraint by the imposed assumptions.

Secondly, the depth map of the image scene can be captured by external sensors and then superimposed on the image for haptic interaction [3, 4]. For images or videos captured by depth cameras such as the Microsoft Kinect™, 3D haptic forces can be generated on the fly.

Thirdly, pre-defined 2D haptic information can be augmented to the image, followed by haptic interaction [5]. It can be a depth image, a texture map or other material data which represents the haptic properties of the image scene. The captured or pre-defined

depth data allows for building topological or 2.5D geometric structure of the image scene so that the viewers can touch the parts of the objects in the image scene facing the camera. However, since no complete 3D haptic models are used in the interaction, the interaction is still in 2.5 dimensions. In order to make images tangible like real 3D scenes rather than embossed structures, several attempts have been made to involve 3D haptic models into interaction with images, nevertheless, most of them only work with low-resolution reconstructed models, failing to provide high-resolution haptic feedback. Moreover, the situations where reconstructed meshes are not available are neglected.

In this research, it is proposed to tackle the problem of augmenting images with 3D models by considering different situations with respect to the way of obtaining the 3D models of the image scene. To confirm the research direction, the relevant papers were surveyed and it was concluded that there are two main cases that need to be considered separately.

One is the situation where only one image is provided and there is no available 3D model for the image. In this case, a way of recovering 3D geometry from a single image is required so that the 3D haptic model can be estimated based on the given image and then augmented for interaction.

The other case is the situation where the reconstructed mesh of the image scene can be obtained through 3D scanning, multi-view reconstruction techniques, etc., so that it could be directly used for augmentation. However, since scene reconstruction using computer vision techniques often leads to millions of polygons, which normally cannot be haptically rendered directly with existing APIs, a haptic rendering method for interaction with large-

scale polygon meshes needs to be established to incorporate reconstructed meshes of original resolution.

Figure 1-3: Illustration of the image-inspired haptic interaction approach. The 3D haptic model is aligned with the pyramid displayed in the image, so that we can not only feel the edges and surfaces visible to us but also touch the back side of the object which is not visible in the image.

Based on this, a systematic way of augmenting images with 3D haptic models was proposed. *It was hypothesized that images of real scenes can be converted into a tangible form with this image-inspired haptic interaction approach so that one could interact with them using desktop haptic devices as if they were real 3D scenes.* This approach is called “image-inspired” in this thesis because the image here only provides inspiration for what models to be used for augmentation while the haptic response is generated from the augmented models and the physical properties assigned to the models. These models do not have to be visually displayed but rather will be haptically rendered complementing the visual experience. After aligning these invisible haptic models with the corresponding parts of the image, a three-dimensional haptic response is generated from the models, allowing the user to haptically explore the whole geometry of the tangible parts (i.e. both the visible

and the invisible surfaces) in the image (as in Figure 1-3) using desktop haptic devices. Haptic material properties such as texture force and friction are also assigned and added for a more realistic haptic perception of the tangible objects in the image.

Figure 1-4: The general pipeline of the proposed image-inspired haptic interaction approach.

The pipeline of the proposed approach is presented in Figure 1-4. In this approach, the augmented models can range from simple geometric models defined based on the image content to high-resolution polygon meshes reconstructed from computer vision techniques. Alignment and haptic rendering methods were designed to deal with each of two different situations with respect to the way of obtaining the 3D models of the image scene: (i) when there is no available haptic model, and (ii) when the corresponding mesh of the image can be obtained through 3D reconstruction. To achieve 3D immersion with images in both situations, the following methods were developed:

- For case (i), a method of retrieving haptic models from a single image with the help of the user input was proposed, and a way of aligning the models with the image was also put forward. *The research novelty of the proposed method lies in the ways of creating haptic models in a 3D haptic workspace distorted by perspective projection and aligning them with the images.* In contrast to existing approaches, the user input, instead of pre-defined prior information, is introduced

for the modelling so that there would be no constraints on image contours. This part is elaborated in Chapter 3.

- For case (ii), to perform real-time haptic interaction with images augmented with reconstructed meshes of original resolution, the problems of interactive haptic rendering of large polygon meshes with reconstruction artifacts and haptic model alignment with the images were solved. In particular, *a novel hybrid collision detection method was proposed to meet the real-time requirements for haptic rendering.* This method combines pre-computed connectivity information and spatial partitioning and can handle collision detection with large-scale dense meshes which normally cannot be directly haptically rendered with readily available APIs. Moreover, three main imperfections (holes, outliers and degenerated facets) inherent in reconstructed models are also handled to help provide a smooth and consistent force feedback. The proposed methods are presented in Chapter 4.

The augmentation methods applied to an image can be then selected based on the availability of a reconstructed mesh of the image scene. Besides these two main augmentation methods, the following tasks were also performed to further enrich the interaction experience with images:

- Haptic perception of material properties were added. *A novel method for simulating virtual weighing with Geomagic Touch was also proposed.* Methods of simulating material properties and virtual weighing are presented in Chapter 5.

- The occlusion effect and limited distortion of the images were created to enhance the visual immersion. The implementation details can be found in Chapter 5.

Chapter 5 presents several interaction techniques with incremental research and development novelties. These techniques were adopted in the haptic interaction with images. Here it is noted that although the elasticity of a haptic object is discussed in Chapter 5, the proposed haptic rendering algorithms for both two cases are focused on rigid-to-rigid collision simulation.

To prove the hypothesis and the feasibility of the proposed image-inspired haptic interaction approach, respective algorithms were implemented and user studies were performed. The results of the user studies and the evaluations are presented in Chapter 6.

The proposed methods and results have been published in the following journals and conference papers:

- J1. **X. Zhang**, M. Goesele, A. Sourin. “Tangible images of real life scenes.”
Computers & Graphics, Elsevier, v. 64, pp. 62-73, 2017.
- J2. **X. Zhang**, A. Sourin, “Image-inspired haptic interaction,” Computer
Animation and Virtual Worlds, vol. 26, pp. 311-319, 2015.
- C1. **X. Zhang**, N. Chen, A. Sourin, “Adding a Sense of Touch to Online Shopping:
Does It Really Help?” In Computer Graphics International, 2017.
- C2. **X. Zhang**, M. Goesele, A. Sourin, “Haptic interaction with a polygon mesh
reconstructed from images,” In International Conference on Cyberworlds
(CW), 2016, pp. 49-56.
- C3. K. Chen, **X. Zhang**, S. Rasool, H. Johan, A. Sourin, “Multisensory Experience

with Images,” In International Conference on Cyberworlds (CW), 2014,
pp. 55-62.

1.3 Report Organization

In Chapter 2, background knowledge of the research topic is reviewed, including the basics about haptic interaction and a detailed survey for haptic interaction with images. A method for interactively augmenting images with haptic models is presented in Chapter 3 while Chapter 4 elaborates on the methods for making real life images tangible with reconstructed meshes. Then, the ways to add simulated material properties into haptic interaction are discussed in Chapter 5, in which a novel method for simulating virtual weighing is also presented. Besides, Chapter 5 also includes the methods for blending the visual immersion with the haptic immersion. In Chapter 6, the user studies proving the hypothesis of turning images into a tangible form with the proposed image-inspired haptic interaction approach are presented, along with a potential application of image-inspired haptic interaction. Finally, a conclusion is drawn in Chapter 7, followed by a discussion of future work.

Chapter 2 Adding Haptic Modality to Images

Based on the research goal, i.e. to turn images into 3D tangible scenes, a brief introduction to haptic interaction is given in Section 2.1. Haptic devices and haptic rendering algorithms, as two main components of a haptic interaction system, are primarily discussed, and the reasons for using Geomagic Touch in this research are also stated. Then in Section 2.2, existing methods for adding haptic modality to images are presented and analyzed. In the end, it is concluded that there is room for research on achieving real 3D interaction with images and the problems that may be encountered in this process are presented, along with the potential solutions.

2.1 Common haptic interaction

The sense of touch, one of our five main senses, lets us perceive the real world through touching. Haptic technologies, also called haptics, allow us to incorporate the sense of touch into virtual reality systems using haptic devices, so that we can not only see but also feel the physical presence of objects in the virtual world and perform operations (e.g., grasping, cutting, trimming, etc.) on them like in the real world.

Figure 2-1: The basic structure of a haptic interaction system. The information and energy exchange in haptic interaction is bidirectional and the haptic device works as a bridge between the human operator and the virtual environment.

Figure 2-1 shows the structure of a virtual reality application for haptic interaction. As described by Salisbury *et al.* [6], there are three main elements in this application in terms of haptic interaction:

- Simulation engine: it computes the virtual environment behavior and updates the haptic display over time;
- Haptic rendering algorithm: the core of the haptic interaction process which computes the haptic response based on the operator input;
- Haptic device: it sends the operator input to the simulation engine and delivers the haptic response to the operator in a form that can be perceived by the operator.

2.1.1 Haptic devices

In real life, the feedback that one gets from exploring an object is basically a combination of tactile (i.e. cutaneous) and kinesthetic feedback. Due to different generation principles of these two types of feedback, most haptic devices provide only either tactile or kinesthetic feedback.

Tactile feedback, collected by the receptors in the skin, makes one feel the hardness (hard/soft), warmth (warm/cold), friction (moist/dry, sticky/slippery), fine roughness (rough/smooth) and macro roughness (uneven, relief) of a material/texture, as summarized by Okamoto *et al* [7]. Tactile haptic devices (i.e. tactile displays) aim at simulating tactile feedback to our skin, mostly to the skin of our fingers. A number of technologies have been proposed for simulating tactile feedback, which include the use of actuators for generating vibration, pressured air [8], ultrasonic waves [9], etc., and a few actuator-free methods, such as creating vertical indentation on the skin with pin arrays [10] and tactile display based on electrovibration [11]. A detailed survey of tactile display types can be found in [12].

Haptic touchscreens and touch buttons are two common tactile displays, of which the most familiar applications are the smartphones and tablets that can create vibrations on

demand with extrinsic actuators. Wearable haptics are also popular choices in simulating tactile feedback. There have been prototypes and commercial haptic wearable devices created for different parts of the body, such as Soundbrenner for the arm [13], Apple Watch for the wrist [14] and the Lechal Mach Insoles for the feet [15]. Despite the versatility of the tactile feedback, the tactile feedback provided by the commercial tactile displays is still limited to vibrotactile sensation [16].

Kinesthetic feedback (i.e. force feedback) refers to the sensation acquired by receptors embedded in muscles, tendons and joints, which reflects hardness, weight, inertia, etc., of an object. A kinesthetic haptic device simulates the kinesthetic feedback and delivers it to the users through tools. Based on the control scheme, there are two types of haptic devices—impedance and admittance controlled—and they have different forms of input and output. The impedance control paradigm is this: the user moves the haptic device and the device will react with a force if a virtual object is met. In admittance control the paradigm is this: the user exerts a force on the haptic device and the device will react with a proper displacement. Admittance controlled haptic devices, such as the HapticMaster [17], have position or velocity source actuators while haptic devices of impedance type work with force source actuators. Most of the kinesthetic haptic devices so far are impedance controlled devices. They receive the position of the haptic interface point (HIP) and respond with force feedback. These devices are normally back-drivable (i.e. one can interactively control the energy transmission between the input and the output of the devices.), have low friction and inertia, and are suitable for providing high-fidelity human computer interaction. In this research, the haptic device that was used (i.e. Geomagic Touch) is an impedance controlled device.

Based on the mechanical ground, kinesthetic haptic devices can be categorized into two groups: the grounded and the ungrounded. Grounded kinesthetic haptic devices occupy the majority of the available kinesthetic haptic devices. Most of them are desktop devices with fixed workspace. In contrast, the ungrounded kinesthetic haptic devices are mostly wearable gloves and exoskeletons specially designed for medical, telerobotics or military use. It is noted that, because of the popularity of mobile devices, research on handheld kinesthetic haptic devices has also become more and more popular in the last ten years. Light-weighted handheld interfaces such as the Haptic Snake and the other handheld prototype developed by Han's group [4, 18] allow the users to interact with virtual objects in air. Since the research goal is to haptically interact with the images displayed on the computer screen in three dimensions, our focus is on the simulation of kinesthetic feedback using grounded kinesthetic haptic devices.

Grounded kinesthetic haptic devices of different performance have been created for different usage, from the 1-DOF (i.e. degree-of-freedom) Haptic Paddle [19] for educational use to the 7-DOF Freedom 7S [20] for surgery training. Among them, the promising haptic devices for this research are 3-DOF or 6-DOF force feedback devices, such as Geomagic haptic devices [21] from Geomagic, 3-DOF Novint Falcon [22] which was designed to be a game controller, and the Omega.x series [23] which features high-end force feedback.

Geomagic haptic devices are commonly-used commercial grounded kinesthetic haptic devices. The Geomagic line of haptic devices range from the mid-range Geomagic Touch to the more premium Geomagic Touch X, and to the high-precise Geomagic Phantom Premium [21]. Normally haptic devices with high positioning precision are expensive. A Phantom Premium 1.5/6DOF can cost up to USD 50,000. Comparatively,

Geomagic Touch and Geomagic Touch X are more cost-effective. Their main device specifications are displayed in Table 2-1. With the use of the OpenHaptics® toolkit, these devices can be integrated into a range of commercial and scientific applications such as 3D modelling and robotic control. Figure 2-2 shows the Geomagic haptic devices and Novint Falcon.

Figure 2-2: Geomagic haptic devices and Novint Falcon.

Table 2-1: Highlighted specifications and sales price for Geomagic Touch, Geomagic Touch X and Novint Falcon.

	Novint Falcon	Geomagic Touch	Geomagic Touch X
Force feedback workspace (width $\times$ height $\times$ depth)	101.6 mm $\times$ 101.6 mm $\times$ 101.6 mm	160 mm $\times$ 120 mm $\times$ 70 mm	160 mm $\times$ 120 mm $\times$ 120 mm
Nominal position resolution	>400 dpi $\sim$ 0.048 mm	>450 dpi $\sim$ 0.055 mm	>1100 dpi $\sim$ 0.023 mm
Maximum exertable force at nominal (orthogonal arms) position	8.89 N	3.3 N	7.9 N
Input/output degree of freedom	3/3	6/3	6/3
Price	USD 239	USD 2,000	USD 4,900

In this research, Geomagic Touch was used as the main haptic device. This medium-priced device has been commonly used in research labs. It has 6-DOF position input and 3-

DOF force output. Compared with cheaper grounded kinesthetic haptic devices such as Novint Falcon (USD 239), which is already discontinued, and the TouchTM Haptic 3D stylus (USD 599), its position resolution is relatively high, which makes it more suitable for touching large-scale, dense polygon meshes. This is one of the reasons for using Geomagic Touch in this research.

Another reason for choosing Geomagic Touch is the release of the Penn Haptic Texture Toolkit (HaTT) [23]. This toolkit solves the problem of lack of programmable tactile feedback for kinesthetic haptic devices by transforming texture force into a kinesthetic force, which can be directly rendered by Geomagic Touch. Haptic texture and friction models of real objects are pre-recorded and then applied in force feedback generation based on the current normal force and speed. This toolkit enables delivering data-driven tactile feedback with Geomagic Touch, so that we can not only haptically render the shape of virtual objects but also present the haptic texture and friction.

Since Geomagic Touch only has 3-DOF force output, only translational force feedback is provided. It is true that if a haptic device with 6-DOF output could be used and the gripping force feedback could be added when touching the images, the interaction would become more immersive. However, such devices usually cost much more than Geomagic Touch.

2.1.2 Haptic rendering algorithms

The haptic rendering algorithm is responsible for computing the haptic response based on the operator input sent through the haptic device. Here only haptic rendering algorithms for impedance controlled haptic devices are described because the majority of haptic devices

implemented today, including the Geomagic Touch used in this research, are of the impedance type.

Figure 2-3: The simplified pipeline of haptic rendering for impedance controlled haptic devices. Here only impedance controlled haptic devices are considered because the majority of haptic devices implemented today are of the impedance type and the reasons for it can be found in [24].

Haptic simulation generally includes collision detection and force response rendering, as illustrated in Figure 2-3. For an impedance controlled haptic device, the operator input is the position of the device end-effector. After the device sends the end-effector position P_d to the simulation engine, the collision detection algorithm checks whether the end-effector is in contact with the virtual environment at position P_d and passes the collision state S and the position P_d to the force response rendering algorithm. Then, the force response F is computed accordingly and returned to the haptic device, thus completing a haptic loop. In the simulated virtual environment, the avatar P_a of the end-effector is calculated at the haptic refresh rate (1000 Hz) and sampled at graphics refresh rate (30-60 Hz) for visual rendering.

Normally, a commercial haptic device comes with an SDK that is only applicable to the devices from the same company, such as the OpenHaptics® toolkit for Geomagic haptic devices and the Haptic SDK for Force Dimension products. These SDKs provide access to specific haptic interfaces and enable programming and defining the low-level features of the haptic interaction. Besides the vendor-specific SDKs, there are also open-source, cross-platform, high-level APIs such as H3DAPI [25] and CHAI 3D [26], which

can be applied to several different haptic devices. These APIs not only allow for defining the interaction properties from scratch but also provide interfaces for existing haptic rendering algorithms. In this research, only Geomagic Touch is used, so a cross-platform API is not necessary. In addition, the research goal is to develop new haptic rendering algorithm and it requires access to defining low-level features of the haptic interaction. Based on these factors, the OpenHaptics® toolkit is selected for haptic simulation.

Based on the representation of the haptic avatar, haptic rendering algorithms could be divided into three types:

- Point-based interaction, where the avatar is modelled as a point – the Haptic Interface Point (HIP).
- Ray-based interaction, where the avatar is modelled as a line segment.
- Object-based interaction, where a 3D object is used to represent the avatar.

In this thesis, the avatar is also referred to as the haptic cursor.

Point-based haptic interaction only renders the interaction between the HIP and the virtual environment. It is relatively easy to implement compared to the other two interaction forms. It is able to simulate touching with the pen tip in real life and can provide haptic information about the shape and surface properties of the virtual objects. In contrast, ray-based and object-based interactions are computationally intensive and are mostly used in medical or industrial applications where a simulated tool is needed for specific operations. The goal of this research is to present photorealistic haptic interaction with images in real time. Considering the computational complexity of haptic rendering in a photorealistic scene, it was chosen to simulate point-based haptic interaction in this research with the

assumption that the simulation of touching with the tip of the pen is enough to provide useful haptic information of the image.

Figure 2-4: Illustration of point-based haptic interaction where the red spheres represent the trace of the HIP and the orange spheres stand for the corresponding proxy positions.

In point-based haptic interaction, the user is able to control the position and sometimes also the orientation of the HIP by moving the handle of a haptic device. In a virtual scene, a *proxy*, also known as the Ideal Haptic Interface Point (IHIP), god-object, etc., is calculated to indicate the position of the HIP. The surface (polygon) that the HIP is in contact with is called the active surface (polygon) in this thesis. As shown in Figure 2-4, when the HIP moves in the free space, the proxy position matches the position of the HIP. When the HIP collides with the haptic model in a simulation of rigid-to-rigid collision, the proxy lies on the active surface (polygon). Since impedance controlled haptic devices have low friction and inertia, they are not able to produce very large resistance force to keep the HIP from intruding the surface. Therefore, in the force response rendering phase, the proxy position needs to be computed and updated for visualization.

Next, the point-based methods for haptically rendering polygon meshes and implicit models will be discussed in details.

2.1.2.1 Point-based haptic rendering with polygon meshes

Polygon meshes have been widely used for modelling in virtual reality applications due to their efficient graphics rendering and simple construction. In order to use the same model for visual rendering and haptic interaction, many haptic rendering methods have been proposed for polygon meshes.

In a rigid-to-rigid collision simulation, when collision happens the proxy (i.e. the IHIP or God-object) always lies on the surface of the model while the real HIP is potentially located beneath the surface. Therefore, most of the existing methods, such as God-Object [27], Ruspini [28] and CHAI3D [26], utilize this constraint to locate the proxy by checking whether the line from the previous proxy to the current HIP intersects with any facet surfaces. If there is only one facet surface that meets the condition, then this surface is considered as active and the proxy lies on this surface, otherwise more computations are required. For example, in the God-object method [27], when a set of active surfaces is found, either the whole intersection test with the “new” God-object location is iterated until there exists only one active facet surface or Lagrange multipliers are used to form a set of equations of which the solution is the proxy position. These methods detect collisions with the whole polygon mesh in each haptic frame. Thus the haptic rendering time depends on the number of polygons.

OpenHaptics HLAPI utilizes the OpenGL Depth Buffer and Feedback Buffer to access shapes rendered in the graphics rendering loop and automatically detects the collision based on the geometry and depth information stored inside these two buffers. In this way, HLAPI’s performance is not influenced by the size of the polygons. However, the Feedback Buffer has a limited size (storing up to 65536 vertices) and using the Depth Buffer

results in discontinuities in the computed haptic force due to the fact that 3D geometry is saved as an image in the Depth Buffer.

There are a number of methods that have been proposed to reduce the computational time using spatial partitioning and hierarchical structures, such as H-COLLIDE [29] and ActivePolygon [30]. In the ActivePolygon algorithm, polygons are stored in an octree data structure. Only the polygons stored in the cells that the haptic cursor passes by between frames are used for collision detection. These methods could effectively reduce the haptic rendering time, however, they cannot handle the situation when the mesh is too dense, because the computation complexity of these algorithms depends on the number of polygons in the cells that the haptic cursor passes from frame-to-frame.

Geometric connectivity information was first used by Ho *et al.* in their “neighborhood watch” algorithm [31] to predict the next active primitive (an extension of the active polygon) based on the previous active primitive. Before haptic rendering, the connectivity among vertices, lines and polygons of the mesh is predefined and stored. After the first collision is detected, only the neighbors of the previous active primitive are checked. In this way, the haptic rendering time is independent of the number of polygons except for every first collision with the mesh.

Inspired by the “neighborhood watch” algorithm [31], a hybrid collision detection method was proposed in this research. It makes computational time fully independent of the polygon count by combining the pre-computed connectivity information and spatial partitioning. The proposed hybrid collision detection method is elaborated in Chapter 4.

2.1.2.2 Point-based haptic rendering with implicit models

Collision with implicit models can be effectively detected compared to models defined by other representations. By substituting the coordinates of the HIP into the implicit function and comparing the resulting value with zero, it can easily be determined whether the HIP is inside or outside the model.

Figure 2-5: Illustration of the haptic rendering algorithm proposed by Salisbury and Tarr [32]. $P_{0,last}$ is the previous HIP and P_0 is the HIP in the current frame. To find the current proxy P_{proxy} , we draw the tangent plane T_{last} of the surface at the previous proxy $P_{proxy,last}$ and project P_0 onto T_{last} . The projection, $P_{0,proj}$, is then used as the seed point. This figure is adapted from [32].

Salisbury and Tarr [32] developed one of the earliest haptic rendering algorithms for analytic functions. When the HIP has the first contact with the implicit-defined model, the proxy position is obtained based on the constraint that it is on the surface of the model. More specifically, the HIP is used as the seed point. Given a fixed step size, the seed point moves towards the surface in the direction of the gradient until the function value at the new seed point approximates zero, and then the new seed point is considered as the proxy. To find the subsequent proxy points, an iteration of the first step is performed based on the tangent plane of the previous proxy (Figure 2-5).

Implicit functions naturally support Constructive Solid Geometry (CSG) modelling, thus it is relatively easy to make topological changes to implicit-defined models. This advantage has inspired many researchers to incorporate implicit solid modelling into haptic virtual sculpting systems [33-35]. For example, Kim *et al.* [33] developed a virtual sculpting

toolkit (VST), in which they used a novel way to haptically render geometric models by pre-computing and storing their discrete distance field on a 3D regular grid. The collision is detected based on the sign of the distance value at the HIP. Hua and Qin [35] presented a haptic sculpting system where implicit B-spline functions are used as shape primitives for object representation. Similarly, the haptic sculpting system proposed by Gao and Gibson [34] is also based on B-spline surface models. Object-based haptic interaction is achieved with the provided multi-resolution sculpting tools and the interaction realism is improved with a novel force rendering algorithm, however, the number of mass points (control points) is rather limited (i.e. less than 1600 on a dual 2.0GHz CPU PC workstation) due to the high computational complexity.

2.2 Haptic interaction with images

With the popularity of smartphones and personal computers, a large number of images is shared online every day. These digital images enable seeing the objects and scenes, but in many situations, we are also expecting to touch the objects in the images. For example, when we are shopping online, despite seeing the images or videos of the products in online shops, we still want to feel the weight, texture, stiffness, etc., of a product with our hands. When we are viewing images of sculptures in digital museums, we also want to touch the sculptures. Not to mention that it would be fun if we would be able to touch family photos shared online and recall the happy moments through the sense of touch.

The existing methods of adding haptic modality to images can be divided into two groups:

- ***Computing haptic response from images***: Methods in this group compute the haptic response straight from the image. A correspondence between the image

features and the haptic geometry of the image is assumed and then the haptic response is computed based on the correspondence on-the-fly.

- ***Augmenting images with haptic data***: The image here only serves as a visual reference for the interaction while the haptic response is generated based on the augmented haptic information. Usually, captured or pre-defined 2D haptic information is superimposed on the image for interaction. There are also attempts to incorporate 3D haptic models of the image scene.

2.2.1 Computing haptic response from images

Methods which retrieve haptic response from images mostly rely on image processing techniques to generate the haptic response on-the-fly during run time. These methods derive a relationship between the haptic information and one or more visual features of the image, such as grey value, color, gradient and edges [2, 36-38]. The visual features can be extracted from the whole image or from some specific areas of the image.

One commonly-used feature is the grey value/intensity of the image. Haptic texture is usually generated based on the assumption that the brightness of a pixel conveys the height information at that point. Ikei *et al.* [39] summarized a few conditions that the image needs to meet to ensure that the brightness distribution of the image matches the height map of a surface, which include the lighting condition for photo shooting, the histogram distribution of the image and the size of the texture pattern. Various methods have been proposed for converting grey values into height maps following different techniques. For example, histogram transformations are adopted by the works [39, 40] where high frequency components of a grayscale texture image are considered to reflect the edge information, and the shape-from-shading techniques are used in the works [41, 42] where a

height map is generated based on the Tsai & Shah Algorithm [43]. Besides the image conditions, there are also works which incorporate human perception into the conversion. For example, Nikolakis *et al.* [2] considered the effect of the Just Noticeable Difference (JND) when mapping the grey value of an image to a haptic perceptible value. The JND for distance discrimination, the smallest difference that can be sensed by a human, is used as a threshold for generating the height map. Three-dimensional forces are directly calculated based on the height map and the kinesthetic information in the previous frame.

While a color image can be converted to a grayscale image for haptic rendering, some approaches work on generating haptic texture using color information. Le Mercier *et al.* [44] proposed four ways to use the illuminance value of a pixel, which is derived from the normalized RGB color components, to compute the texture elevation at that point. Then based on the elevation map and the original object surface, the haptic cursor position along the height direction and the tactile force are estimated. Li *et al.* [36] presented an HSV-based method to compute the normal and tangential components of the haptic texture separately. In this method, it is assumed that warmer/brighter color areas in the images are bumps and cooler/darker color areas are depressions. The normal component of the texture force is computed from the hue and the luminance values at the HIP while the tangential component is generated according to the relative RGB color variation between the HIP and its neighbors.

The estimated height map provides a three-dimensional texture from the image. However, when the interaction goal is just to feel the contours displayed in an image, height map is not necessary. The edges and contours can be directly extracted from the images using edge detection techniques and then used for haptic interaction. Pai *et al.* [38]

presented one of the earliest haptic interaction systems for feeling the boundary curves in the images. In this system, edges are first detected and linked to create a closed boundary curve. Then, multiresolution function-based models of the boundary curve, as well as their bounding boxes at each level, are constructed for fast collision detection. If there is an edge contact between the HIP and the boundary, a force is generated perpendicular to the edge. Besides the outcome of edge detection, the way the contours are haptically rendered is also largely determined by the application of the haptic interaction method. For example, in order to trace the contours in an effective way, different methods of rendering contours have been proposed, such as constraining the HIP to the object contour with attractive forces [45], incorporating the movement of the HIP into the force response generation [46] and interpreting the boundary and edges as ridges or grooves [47-50]. If assisted by audio feedback, enabling tracing the edges and boundaries in the images is a promising means to generate haptic maps for the visually impaired [45, 51-53].

As discussed previously, a common way to make tangible contours is to detect and store the contours. In contrast, methods such as the ones by Vasudevan *et al.* [37] and Rasool *et al.* [54] provide a solution to feeling the contours displayed in the image without the necessity of storing any geometric information. Based on the assumption that the darker/brighter color areas of an image represent concave/convex regions, a two-dimensional texture force is generated with the gradient technique [55], always attracting the user towards the darker areas and repelling the user from the brighter areas. The same technique is applied by Kim *et al.* [56] and Wu *et al.* [57] to generate electrovibration on touchscreen devices using the image gradient and the finger motion when sliding on the screen. The electrovibration creates lateral friction between the finger and the touchscreen,

resulting in an illusion of touching bumps. Notably, there is a growing number of image-based haptic interaction methods designed for haptic touchscreen devices [58, 59], in which the contours are only detected for marking the tangible area on the image. In the works of Xu *et al.* [58] and Israr *et al.* [59], it was explored how to help the visually impaired to explore the surrounding environment with TeslaTouch [11] by capturing the surrounded objects with a camera and displaying the tangible images on TeslaTouch. When the users move their fingers to the area enclosed by the detected contours, pre-defined electrical signals are generated to create tactile feedback to the fingers.

2.2.2 Augmenting images with haptic data

Instead of retrieving haptic feedback from images, a number of methods derive a force response based on the models augmented to the image. Mostly, captured or pre-defined 2D haptic information is superimposed on the image for interaction. It can be the depth image captured by external sensors, friction map, electrical signals, etc. Besides, there are also works which directly augment images with 3D haptic models representing the scene geometry.

2.2.2.1 Superimposing 2D haptic information

The 2D haptic information used for augmentation can be divided into two categories based on the source: captured and pre-defined.

The idea of superimposing images with captured depth values for each pixel, also called Depth Image-Based Representation (DIBR), was first proposed by Bayakovski *et al.* [60]. It was designed to convey the geometry information of the image scene without using complex polygonal models. This representation method was adopted by several works for real-time haptic interaction with videos [61, 62]. In these works, videos are considered as a

series of images. For each image, the 2.5D surface model of the image scene from the same viewpoint is constructed based on the corresponding captured depth image. Normally, the depth data captured by depth cameras contains noise and errors and haptic interaction with the captured depths produces distorted haptic feedback. To allow a smooth haptic interaction with the original captured videos in real time, Kim *et al.* [61] focused on sampling and filtering the depth values to only keep the feature points for the polygonization. Cha *et al.* [62] presented an alternative approach to accelerating the haptic rendering of the reconstructed 2.5D surface models by adopting the “neighborhood watch” algorithm proposed by Ho *et al.* [31]. Connectivity between the primitives is built before the rendering for fast searching of the contacted primitive. Besides reconstructing a 2.5D mesh from the depth image, another way to interact with the captured depth image is to convert it to a point cloud. Several methods have been proposed for haptically rendering static or streaming point clouds by representing the whole point cloud as implicit functions [63, 64] or by estimating the local surface enclosing/representing the neighboring points of the HIP [65].

With captured depth data, the scene reconstruction is not constrained by any assumptions on the image conditions. However, a common problem of using captured depth images is that, since normally a full scene cannot be captured with one single depth camera, the haptic models retrieved from the depth images only represent the geometric structure of the scene facing the camera, i.e. the interaction is still in 2.5 dimensions. In addition, many haptic rendering methods mentioned above used low cost depth cameras such as Microsoft Kinect™ to capture depth data. Due to the limited precision of these depth cameras, the fine texture details on the object surfaces are generally missing. To perceive the fine details, either high-resolution depth maps are to be used or complementary haptic texture

information needs to be added as in Rasool *et al.* [3]. They proposed to combine image-based haptic texture rendering and depth-based haptic rendering so that the captured depth data is only used for computing a basic constraint force while the details on the surface are provided by a texture force generated using the image-based method in [37].

In addition to captured depth images, haptic texture models built from recorded signals are also used by specific haptic devices. Recently, customized databases of haptic texture models have been established for Geomagic Touch, a commonly used device in research labs. Penn Haptic Texture Toolkit (HaTT) [66] released by the haptic group at University of Pennsylvania, was one of the first texture databases for Geomagic Touch. It provides 100 texture and friction models that were built based on the speed and force signals captured in the recording. The haptic group at Technische Universität München also released a texture database [67] for Geomagic Touch. It contains texture models for 43 different materials. Compared to HaTT, this database provides two texture models for each material, which were generated with the scan parameters collected during the controlled well-defined and the uncontrolled human exploratory texture scans respectively.

As to the pre-defined 2D haptic information for augmentation, one notable example is the electrical signals used by touchscreen devices like TeslaTouch. Bau *et al.* [11] identified the electrical signals for simulating perceptions of different textures on TeslaTouch through a set of experiments. These pre-defined signals were directly used in [58, 59] for delivering distinct haptic textures. Besides texture information, other haptic material properties such as stiffness and friction can also be pre-defined and augmented to the images. For example, Kim *et al.* [5] proposed to make interactive images by embedding the images with compressed depth, stiffness and viscosity maps of the same sizes. Among

them, the stiffness and viscosity maps are pre-defined while the depth map is generated based on image intensity. Li *et al.* [68] presented a multi-sensory interface for medical images where the haptic material properties and the applied regions can be interactively defined by the medical professionals.

2.2.2.2 Images with augmented 3D models

In order to allow the viewers to touch the objects in the image scene from the back side, the full 3D geometry of the image scene is necessary. Existing methods for augmenting images with 3D haptic models mostly rely on multi-view reconstruction techniques or 3D scanning to obtain the haptic model. Multiple images or depth images of an image scene taken from different views are required for the reconstruction. One notable algorithm for reconstructing 3D scenes using depth images is KinectFusion [69], which enables real-time capturing and reconstructing indoor scenes with Microsoft Kinect™. In KinectFusion, the image scene is represented as truncated signed distance function (TSDF) for update and displayed as mesh through ray-casting. This algorithm was incorporated by Tian *et al.* [64] into haptic interaction with the deformable streaming data. In [64], collision detection and deformation is directly performed with the implicit surface obtained from KinectFusion. Based on the experiment results, real-time haptic interaction can only be achieved with low TSDF resolution in this method. Since here the TSDF resolution is related to the 3D surface resolution, it means that this method can only work with low-resolution reconstructed models.

Real-time haptic interaction with high-resolution models reconstructed from images has always been a problem. The reason behind it is that 3D reconstruction of a real life scene using computer vision techniques often yields large-scale partially dense polygon

meshes which cannot be haptically rendered in real time. To meet the real-time requirement, different methods have been proposed, such as using multi-resolution 3D models [70, 71], replacing complex models with simplified models complemented by image-based texture force [3] and restricting the contact regions [72]. These methods can accelerate the collision detection process, but at the cost of losing geometry details on the reconstructed surface.

When it comes to 3D reconstruction with only one image, since this is an underdetermined system, the task is almost impossible unless prior information about the object geometry in the image is given. Current single-view reconstruction methods mostly rely on computer vision techniques and pre-defined geometric constraints, class-specific features, databases, etc., to retrieve the camera parameters. Therefore, these methods are normally customized and restrict their applications to images of specific category of objects (context), such as architecture [73, 74], human faces [75, 76] and polyhedral objects [77, 78]. Five types of pre-defined geometric constraints – coplanarity, parallelism, perpendicularity, symmetry and distance, as summarized by van den Heuvel *et al.* [73], are commonly used for calibrating the camera, and then models are loaded from the database based on computer vision techniques. Besides geometric constraints, class-specific features are also important references for mapping the pre-defined model to the image. For example, given that every human face shares the same structure features, e.g., eyes, nose, mouth and facial contour, a frontal face image could be defined by vectors representing these features. Based on this, Lu and Jie [76] obtain vectors of feature points from a database of facial models and use these vectors to approximate the target feature points. Recently deep learning models such as Convolutional Neural Networks (CNNs) have also been used for predicting object geometry from single images. There has been some progress [79, 80], but

the models thus reconstructed only have basic geometry structures unless additional constraints/features, e.g., assumption on the lighting of the images [81], are applied during the reconstruction to help add details based on the image content.

Despite that pre-defined constraints/features can help retrieve the camera parameters, map the target model to the image or add geometric details to the rough models, they also restrict the applications of these methods to certain type of images. Moreover, the use of computer vision techniques poses constraints on contour condition in the reconstruction process, which is, the objects in the images need to have a clear contour. If the objects to be reconstructed are partially occluded or their silhouettes are not clear in the image, the reconstruction results would be ambiguous. In order to remove the constraints on contour condition, a flexible way to define prior information for constructing models from one single image is required. The previous works [82-84], which incorporate user input into single view reconstruction, imply that the user input is a promising way for introducing human perception into the reconstruction and it is able to provide diverse prior information based on the image content. In most of these methods, the user's sketch is given as an input. Compared to traditional WIMP (window, icon, menu and pointer) interfaces, it is easier and more intuitive to define geometric features through sketching.

2.2.3 Critical analysis and conclusion

Methods which retrieve the haptic response from images rely on image processing techniques to estimate the haptic forces directly from the image content. They allow for haptic perception of the whole image while giving prominence to the sensations of touching object edges and textures. The advantages of these methods are apparent. They are simple, computationally efficient and can achieve real-time interaction. However, they fail to

provide the perception of the whole geometry displayed in the image, which means that the image is perceived tangibly only as an embossment of the relief. Besides that, since none of the visual features can always represent the scene geometry of any image, these techniques can only be applied to a specific group of images.

In contrast, methods which augment images with haptic data do not rely on any assumptions on image conditions. They make images tangible by computing the haptic response based on the augmented models. Usually, the augmented models are captured or pre-defined 2D haptic information, from which a data-driven haptic texture or 2.5D haptic model can be extracted for interaction. Since the goal is to convert the image into a 3D interactive scene, full 3D geometry of the image scene needs to be augmented. However, few attempts have been done to incorporate 3D haptic models to the interaction with images and most of them only work with low-resolution reconstructed models.

Based on the survey on haptic interaction with images, it is concluded that there is room for research on achieving real 3D interaction with images by augmenting images with 3D haptic models. To obtain the corresponding 3D models, the following problems are expected in the reconstruction process.

- If the 3D reconstruction from images is done with a large number of images or depth images, the model thus reconstructed is likely to have millions of triangles inside, which normally cannot be haptically rendered directly with readily available APIs. Simplifying the mesh or the collision detection process can help, but at the same time it leads to a loss in geometry details on the reconstructed surface.

- If only one image is given, reconstruction from the image is almost impossible unless prior information about the object geometry in the image is given. However, normally pre-defined prior information can only provide limited geometric clues for reconstruction and it is not always available.

To tackle both problems, a way to haptically render original resolution reconstructed meshes needs to be found and a method of retrieving haptic models from single image without the use of pre-defined prior information needs to be developed. After surveying single-view reconstruction methods, user input is considered as a promising candidate for resolving the ambiguity in model construction. The following two chapters will deal with the two problems mentioned above separately.

Chapter 3 Interactive Augmenting Images with Haptic Models

This chapter discusses how to make an image tangible if it is the only image that is provided for interaction and there is no readily available haptic model of the image scene. Given this scenario, the first question that needs to be answered is how to obtain a haptic model for interaction from one single image.

Based on the survey on reconstruction from one single image in Section 2.2, it is known that in order to remove the constraints on contour condition a general and flexible way to resolve the ambiguity in single view reconstruction is required, and the user's sketch has been considered as a promising candidate to provide diverse information without generating any constraints. Converting 2D sketches into 3D solid models is usually done using inflation or template matching techniques. Methods using inflation techniques use a 2D silhouette sketch input and inflate it into a 3D polygonal model. They are suitable for building free-form planar symmetric models but lack flexibility in the definition of depth offset and often require a complete sketch of the silhouette and more fine strokes for detailed topological features on the surface [85-87]. Some works, such as [88], avoid tedious drawing procedures by using image processing techniques to extract the structure of the object from the image, however, a clean image background is required for them. Template matching methods suffer from limited range of application because they require specific templates for each category of objects [89-91]. Some works, such as 3-Sweep [82], offer the users more generic primitive templates, expanding the application range to real photos and presenting some successful examples with models of complex structure.

By analyzing the disadvantages of existing single view reconstruction methods and the most promising ways of resolving ambiguity of 3D reconstruction from single images, a method of augmenting displayed images with invisible haptic models is proposed, where the 3D models are interactively sketched on the images and responsible for haptic interaction.

Inspired by sketch-based methods such as 3-Sweep [82], which provide simple templates to match different parts of the sketch individually, it is assumed that the tangible part of the image could be broken down into several subparts that can be represented as a combination of primitives. *Based on this assumption, it is hypothesized that the proposed method can convert images of real scenes into a tangible form without the necessity of its full 3D reconstruction.* The specifics of the proposed method compared with other methods on image-based modelling and haptic interaction is that the proposed method works with real photos, which often have images of objects with uncertain or partially occluded contours. Therefore, numerous ambiguities have to be resolved by the users.

3.1 Overview

In the proposed method of interactive augmenting images with haptic models (as illustrated in Figure 3-1), invisible haptic models are interactively defined through sketching to match the contours of the respective parts of image. After that, the haptic models are projected onto the image plane for alignment. Combined with the invisible haptic models and using a special visual/haptic player for visual/force rendering, the images can deliver multisensory interactive experiences as if they were actual 3D scenes.

To minimize the haptic model size, so-called FReps [92] (i.e. variants of implicit functions $f(x, y, z) \geq 0$) are mostly used in this approach for defining the haptic models.

In this way, the haptic models can be saved in a small haptic file of a few kilobytes that can be easily shared along with the image online. This makes the proposed method specifically suitable for real-time web applications such as e-shopping and web-based medical training.

Figure 3-1: The pipeline of the proposed image-inspired haptic interaction approach where the haptic models are interactively defined by user sketch.

Based on Figure 3-1, each step of the proposed method is elaborated in the following sections. Section 3.2 presents a novel way to interactively sketch haptic models on the image. The alignment between the haptic models thus constructed and the image, i.e. the projection of the haptic models onto the image, is described in Section 3.3. Section 3.4 focuses on explaining how the models are haptically rendered with respect to collision detection and force rendering. Then in Section 3.5, the concept of a “haptic” file is introduced and described. Finally, several examples of haptic interaction with photos using the Geomagic Touch are given in Section 3.6 to show the results.

3.2 Sketching haptic models

While making haptic models, the 3D locations of the objects in the image are not retrieved

or estimated from the image properly based on a camera model. Instead, each of the haptic models is located in front of the image plane neglecting their actual relative depth in the real scene while still preserving mutual occlusions wherever they happen, i.e. the relative depth of the models in the haptic workspace is assigned by the user. Therefore, the user will have the ability to touch any part of the scene made tangible with the “virtual hand”. Besides this, the camera parameters are not estimated in the modelling process. In many cases, it is not even possible if there are no proper reference objects displayed in the image. Instead, a new method is proposed to create haptic models in a 3D haptic workspace distorted by the perspective projection so that if the haptic models were projected onto the image with parallel orthographic projection, their boundaries would match the contours of the respective parts of the image. During the haptic interaction, the 3D virtual representation of the HIP (i.e. the haptic cursor) is also projected onto the image using parallel orthographic projection (Figure 3-2).

As shown in Figure 3-2, to match the contours of the perspectively distorted images of the cube and the cylinder, two 3D haptic objects are created and placed in front of the image: a polyhedron bounded by 6 non-parallel polygons and a truncated elliptical cone with non-parallel bases. Given how haptic interaction with one HIP is performed, i.e. by perceiving the whole model shape based on the haptic feedback while the haptic cursor is moving along only one contour or surface at a time, geometric relations between two tangible surfaces, such as parallelism and orthogonality, would not be haptically perceived by the users, thus it is not necessary to have them preserved in haptic modelling. Therefore, as long as the movement of the HIP follows the visually perceived edges and surfaces while

sliding on it, this method provides for believable combination of visual and haptic immersion into the image implementing the principle “What You See is What You Touch”.

Figure 3-2: Location of haptic objects with reference to the image plane.

Not excluding completely definition of haptic objects with other representations like polygons and voxels, analytic FReps [92] (i.e. variants of implicit functions $f(x, y, z) \geq 0$) are mostly used for defining the haptic models in the proposed method since this ensures:

- To easily get solid models rather than just surfaces.
- To generalize collision detection algorithm for different types of haptic models.
- To support Constructive Solid Geometry (CSG) modelling with an open set of primitive haptic models.
- To provide significantly smaller size of the model definitions since only FRep function coefficients and affine transformation parameters have to be stored in the haptic model description.

The use of FReps for defining haptic models makes the proposed method highly expandable – any types of function-based models, as long as they match the image content, can be added for the interaction.

In the proposed method, it is assumed that the tangible part of the image could break into several subparts that can be represented as a combination of primitives. Therefore, simple strokes are used to define haptic primitives in a 3D haptic workspace distorted by the perspective projection, and the tangible part of the image is represented as combinations of distorted primitives such as planes, cones, ellipsoids, or more complex objects such as “inflated” shapes. Next, novel ways of defining the distorted haptic models with a few strokes are proposed. The corresponding function representations of the haptic models are also presented.

3.2.1 Sketching primitives

In constructive solid geometry, models are built as Boolean combinations of simple geometric primitives such as spheres, cylinders, cones, cubes and pyramids. In a 3D haptic workspace distorted by the perspective distortion, geometric relations between two lines/surfaces are lost, e.g., spheres would be projected to ellipses and the projections of the two bases of a cylinder are likely not be parallel. Therefore, in order to match the distorted contours displayed in the images, besides the commonly used regular primitives, basic components of distorted geometries such as the planes and ellipsoids are also required.

Table 3-1 shows all the haptic primitives provided in the proposed method for haptic interaction and the ways to define them with strokes. The stroke patterns are selected so that the essential geometric information can be easily retrieved for definition of the haptic primitive. Auxiliary points and lines would pop up while sketching, e.g., the midpoint of

each stroke is marked as a reference for drawing perpendicular lines, and auxiliary lines which indicate the normal directions of the planes are also displayed while defining the plane normal.

Table 3-1: Stroke patterns for defining primitives. The arrows indicate the drawing direction of the strokes. For cylinders, cones, cuboids and truncated pyramids, there are several different ways for sketching. The users need to follow the instructions for drawing and it requires some drawing skills. Auxiliary lines are marked as red and green. The sign * means that the orientation of this primitive can be further adjusted after definition.

Shapes	Ways to draw	Examples
Planes in a point-normal form	Create with two strokes. The first stroke is a click which defines a reference point on the plane while the second stroke builds a circle. By changing the circle to an ellipse, i.e. changing the length of the major and minor axis, the normal of the plane is defined. The green auxiliary line determines whether the normal direction is inward-pointing or outward-pointing.
Planes in a three-point form	Create with three strokes. Each stroke is a click which defines a point on the plane.
Spheres	Create with one stroke. This stroke defines the diameter of the sphere.
Ellipsoids	Create with two consecutive strokes. The first stroke defines the major axis of the ellipsoid while the second one defines the minor axis by connecting the vertex and the co-vertex.

<p>Right circular cylinders</p>	<p>Create with three consecutive strokes. The first two strokes define the radius and orientation, and the last stroke defines the height.</p>
<p>Right circular (truncated) cones*</p>	<p>Create with four consecutive strokes. The first two strokes define the radius of the lower base and the orientation, and the third stroke defines the height. If the cone is truncated, the last stroke defines the radius of the upper base, otherwise it is replaced with a click.</p>
<p>Infinite right circular cones*</p>	<p>Create with two strokes. The two strokes define the two generatrix lines forming the cone and the angle between the two strokes defines the aperture of the cone.</p>
<p>Cuboids</p>	<p>Create with three strokes. All three strokes must start from the same vertex of the cuboid and follow visible edges. Each of the strokes defines one edge length.</p>

<p>Truncated square pyramids</p>	<p>Create with four strokes. The first two strokes define the edge length of the base and the orientation while the third stroke defines the height. All these three strokes must start from the same vertex of the truncated pyramid and follow visible edges. The last stroke starts from the end point of the third stroke and defines the length of the top edge that is parallel to the second stroke.</p>
--	---	--

Following the stroke patterns displayed in Table 3-1 allows defining the haptic primitives using just a few strokes. Specifically, a novel way of interactive modelling planes is presented. The previous methods for recovering the orientation of a plane mostly work on images with specific features such as regular texture patterns [93, 94] or vanishing points [95], and geometric assumptions as summarized in [73] are commonly required for recovering the perspective distortion. However, there is no such feature or assumption that can be applied to every image. Therefore, in this research, all the ambiguities are resolved by the users so that the existence of specific image features or geometric constraints is not required. It is proposed to reconstruct the plane from its projection by sketching a projection of a circle on the plane (which however requires some drawing skills). Since a circle always projects into an ellipse after it is projected onto a plane (with the extreme cases of the circle itself or a segment), it is possible to compute the plane normal vector based on the projection and thus reconstruct any plane on which this circle is sketched. Figure 3-3 shows how to define the normal vector of a plane by constructing the projected ellipse: First a circle is constructed by drawing its diameter. This circle is set to be on a plane parallel to the image plane. At this time the projection of this circle overlaps with itself and the plane normal is perpendicular to the image plane. Then, this circle is transformed into an ellipse by rotating the circle around its center and changing the lengths of its major (marked as red) and minor

(marked as green) axes. This ellipse is considered as the projection of the circle after a 3D rotation and its semi-minor axis indicates the direction of the plane normal thus defined.

Figure 3-3: The 2D sketch and the corresponding 3D model. The red ellipse is regarded as the projection of a circle on the reconstructed plane. By constructing the projected ellipse, the circle normal is defined and thus the plane normal is also known.

Figure 3-4 further explains how the plane normal is calculated based on the projected ellipse. Here θ is the rotation angle of the ellipse about its major axis. The semi-major axis of the projected ellipse is considered as the radius of the circle. Then based on the minor axis of the ellipse, the projection of the circle's diameter on the image plane, the rotation angle θ can be obtained. Along with the rotation angle around the circle's center, the plane normal is determined.

Figure 3-4: Projection of the circle on the image plane.

All the haptic primitives are initialized with their main axis or reference point on a default plane, which is $z = 0$ in this method, unless they are defined by points with known three-dimensional coordinates. Then the depth position of the haptic primitives could be further changed by the users to ensure the occlusion effect. Note that since the haptic models are invisible, the users decide whether to make adjustments based on touching the models that have been defined. If what they touch does not match what they see, then adjustments need to be made by reassigning the depth positions. Here a way to adjust the orientation of some of the haptic primitives after definition is also offered. The shape will also be changed automatically during the adjustment to keep the projection contour unchanged so that the boundary of the projection would still match the image. Take the infinite cone model as an example. The main axis of an infinite cone is initialized on plane $z = 0$, which is parallel to the image plane. After definition, its orientation can be further adjusted by rotating it towards or away from the image plane along a defined axis. By choosing one point on the side contour, a rotation axis passing through the chosen point and perpendicular to the main axis is set. As it is shown in Figure 3-5, the cone would be stretched while it was rotated to match the upper part of the contour which is between the rotation axis and the vertex. If the rotation axis is line bc , then point b or c should be selected and the new opening angle α' is:

$$\alpha' = 2 * \arctan(\tan \frac{\alpha}{2} \cos \beta) \quad (3.1)$$

where α is the opening angle before the rotation and β is the rotation angle. In this way, the projection of point α' on the image plane matches that of point a , and thus $\alpha'bc$ and abc share the same projected contour.

Figure 3-5: Illustration of shearing an infinite cone.

3.2.2 Sketching complex geometries

Figure 3-6: Illustration of the stroke-based sketching of the truncated cone.

Based on the assumption that the tangible part of the image could break into several subparts, which can be represented as a combination of haptic primitives, in the proposed method images are made tangible by augmenting the haptic primitives into the images through sketching. The sketch-based modelling process is generally done in the following way: Firstly, define a model. Then use this model as a reference to define other models. Adjustment, i.e. displacement in depth and sometimes shearing as well, can be applied to the model whenever it is needed. Two examples are presented below to illustrate how this

works. In both examples, tangible objects are defined as Boolean combinations of simple distorted haptic primitives.

The first example illustrates how to make the conical building displayed in Figure 3-6 tangible by associating the respective part of the image with a haptic model of the building. As can be seen in Figure 3-6, this cone-shaped building has two non-parallel bases. Therefore, it can be defined as an intersection of two planes and an infinite cone. To construct an infinite cone model which matches with the main body of the building, two pairs of points e, f , and g, h are selected first on the opposite contours (sides) of the conical building, defining two vectors $\overrightarrow{ef}$ and $\overrightarrow{gh}$, the intersection of which yields the calculation of the cone vertex as well as its opening angle. Thus the function of the infinite cone can be obtained as in Equation 4.2 and any point on the cone becomes known. The upper base of the cone N can be defined by selecting three points on the contour (points a, b, c in Figure 3-6). The depth values of these points are calculated based on the cone function, and then they can be used along with the two image coordinates to define the upper plane. In the similar way the lower plane of the cone M (which is also a ground plane) can be defined. The haptic model of the conical building can thus be defined as a Set-Theoretic intersection:

$$\begin{aligned}
 f_1(x, y, z) &= \mathbf{N}_1 \cdot (\mathbf{P}_1 - \mathbf{P}) \geq 0 \\
 f_2(x, y, z) &= \mathbf{N}_2 \cdot (\mathbf{P}_2 - \mathbf{P}) \geq 0 \\
 f_3(x, y, z) &= \cos\left(\frac{\alpha}{2}\right) \sqrt{x^2 + z^2} - y \sin\left(\frac{\alpha}{2}\right) \geq 0 \\
 f_i \& f_j &= f_i + f_j - \sqrt{f_i^2 + f_j^2} \\
 f_{truncated_cone}(x, y, z) &= f_1 \& f_2 \& f_3 \geq 0
 \end{aligned} \tag{3.2}$$

where N_1, N_2 are plane normal, P_1, P_2 are points on the planes, α is the opening angle of the cone, and f_1, f_2, f_3 represent the functions of upper base, lower base and the main body of

the building respectively. The Intersection denoted by symbol $\&$ is defined here with a so-called r-function to preserve continuity of the defining function for its smooth haptic rendering [92]. Thus, only N_1, N_2, P_1, P_2 and α need to be eventually stored.

Figure 3-7: Rotation of a truncated cone towards the image plane with the rotation axis defined on the ground plane.

As in Figure 3-7, the axis of the cone model is initialized as parallel to the image plane. Wherever it is needed, displacement in depth and rotation of the cone about a horizontal axis (towards and away from the viewer) can be done interactively. In Figure 3-7, the cone model has been rotated towards the image plane while the rotation axis was on the ground plane. Note that the function of the base above the rotation axis will also be changed along with the rotation to keep its projection on the image plane unchanged. Therefore, it can be seen in the figure that the top base of the cone has changed after rotation while the lower base still remains the same.

In the second example (Figure 3-8(a)), an image of the Louvre pyramid is made tangible by augmenting a polyhedron model representing the pyramid and a plane model representing the ground into the image. Here, with the knowledge that the Louvre pyramid is four-sided, firstly two of the pyramid's adjacent faces are defined, then the depth information is derived out of them, and finally the remaining two (invisible) faces are

defined based on the derived depth information. The construction begins by sketching an ellipse on the plane polygons (P, Q) followed by setting one reference point on them (point a) and assigning a zero depth coordinate to this point. The calculated normal vector N and one point P_0 on the plane yield the equation of the plane $N \cdot (P - P_0) = 0$ which then can be used to define a solid half-space on one of the plane side $f_1 = N \cdot (P - P_0) \geq 0$ or $f_1 = N \cdot (P_0 - P) \geq 0$ to include it in the definition of the haptic object following Equation 3.3. In order to intersect the two haptic planes at the edge matching that of the visual planes (P, Q), their definitions should share the same reference point which has to be located on that edge. For the plane polygons which are invisible or ambiguous, like rear faces of the pyramid or the horizontal plane surface R in Figure 3-8(a), the plane equation coefficients are derived by calculating the depth coordinates of three non-collinear points on their faces (such as points b, c, d located on the pyramid contour, and point e located behind the pyramid), which are also points located on the adjacent visible faces (P, Q) with already derived equations. The rear part of the pyramid can then be interactively edited by relocating the hidden point (point e) within its depth dimension. In Figure 3-8(a), this point and the two sketched ellipses complemented with red arrows interactively prompting the direction of the face normal define the shape of the haptic pyramid model.

The interactively sketched pyramid and the horizontal plane R are finally combined into one solid FRep object, which is placed in front of the image plane (Figure 3-8(b)). This can be visually verified with the interactive FVRML software tool [96] by loading the generated function definition there. The final FRep function of the haptic model is obtained as an intersection of the four half-spaces $f_i = N_i \cdot (P - P_{i0}) \geq 0, i = 1, \dots, 4$ bounding the pyramid followed by the union with the solid space f_5 which is below the ground plane:

$$\begin{aligned}
& f_1 \& f_2 \& f_3 \& f_4 \mid f_5 \geq 0 \\
& f_i \& f_j = f_i + f_j - \sqrt{f_i^2 + f_j^2} \\
& f_i \mid f_j = f_i + f_j + \sqrt{f_i^2 + f_j^2}
\end{aligned} \tag{3.3}$$

(a)

(b)

Figure 3-8: (a) Stroke-based sketching of the plane-bounded solid object. (b) Placement of the FRep solid in front of the image plane.

Note that the relative location of all haptic models in the depth dimension can be adjusted by translating them orthogonally to and from the image plane to ensure mutual occlusion of the tangible models where it is required as well as by applying shear transformation to them. The correctness of the interactive constructions is constantly verified by haptic interaction—following the edges and surfaces of the haptic models being created with the HIP.

The use of FReps for defining haptic models makes the proposed method highly expandable. Readily available implicit-based modelling techniques can be easily incorporated into this method for building haptic shapes. Take variational implicit surfaces [97] as an example. They are created from a set of constraints (i.e. zero-value constraints, positive constraints, gradient constraints, etc.) through interpolation and can be used to represent the iso-surfaces that pass through the zero-value points. Since variational implicit surfaces naturally support CSG modelling, they can be integrated for building freeform haptic shapes by sketching the boundary points and define constraints in the following way. First, we mark the whole contour of the object in the image with a set of points initialized on plane $z = 0$ and their field values are automatically assigned to zero. Besides these boundary points, additional points are selected within/outside the contour of the object with their field values assigned to positive/negative. After that, the function of 2D scalar field enclosed by the boundary points can be obtained by interpolating all the points using variational implicit functions and solving the equation system thus formed. This variational field function approximates the 2D contour of the object and can be then used for inflation. Suppose $m_i(x_i, y_i)$ are the sketched points with field values assigned to v_i and the function of the 2D scalar field is f_c , then $f_c(m_i) = v_i$. Based on the general variational interpolation method, a linear equation system is built as follows:

$$\begin{cases} f_c(\mathbf{u}) = \sum w_i \|\mathbf{u} - \mathbf{m}_i\|^2 \ln(\|\mathbf{u} - \mathbf{m}_i\|) + c_1 u_x + c_2 u_y + c_3 \\ \sum w_i = 0 \\ \sum w_i x_i = 0 \\ \sum w_i y_i = 0 \end{cases} \quad (3.4)$$

where the weights w_i and the coefficients c_1 , c_2 and c_3 can be obtained by solving this equation system and the function of the 2D contour curve is thus established.

(a)

(b)

Figure 3-9: Images with sketch input and the corresponding haptic models displayed using the interactive FVRML software tool [96].

Based on the object in the image, techniques for inflating the field function can be chosen. Figure 3-9 shows examples of creating haptic models with variational implicit functions. In Figure 3-9(a), the contour of the phone is defined by 64 strokes. Another 4 strokes are sketched within the contour and positive field values are assigned to imply that they are inside the shape. The start points of all these strokes are used for establishing the field function and then the phone model is formed by intersecting the field function with two cutting planes parallel to plane $z = 0$. The depth of these two cutting planes are defined

following the thickness of a real phone. The whole sketch and construction time takes around 67 seconds. Figure 3-9(b) shows how models defined by variational implicit functions can be combined with the provided haptic primitives. The glass bottle is defined with 44 strokes on the contour and 22 strokes inside the contour, which takes around 60 seconds to sketch. The variational field function thus constructed is normalized to be between -1 and 1, and then the depth of the bottle is determined based on the normalized field function and an offset function which defines the distance between the points inside the contour of the object to a center point. The box is modelled as a six-sided polyhedron by first defining the front and upper faces and then the invisible faces are determined based on the known faces. It is noted that the depth position of the freeform haptic shapes could be further changed by the users to ensure the occlusion effect.

In the proposed sketch-based modelling method, by introducing human perception through sketch into the interpretation of the object shapes, a part of the haptic model shape can be determined. On the basis of that, the missing depth information of the rest part of the haptic model can be extracted for further modelling. All the haptic models are initialized around a default plane. Their depth positions can be reassigned by translating them orthogonally to and from the image plane to ensure mutual occlusion of the haptic models. Easy rotational adjustments are provided to some primitives to bring more flexibility into the sketch-based haptic modelling. Compared to common image-based modelling methods, allowing the users to subjectively create distorted haptic primitives based on the image content makes it easier to construct perspectively distorted models without concern about the actual relation between different parts of the models or the existence of ambiguous contours. Here it is noted that, although the haptic primitives and operations that have been

implemented and included in the proposed method so far are not able to create all the desired haptic shapes, the presented examples have proved the high expandability of the proposed method. More implicit modelling techniques can be incorporated based on the applications.

3.3 Projecting haptic models

As little information could be obtained by relying solely on one image, reconstructing a perspective projection matrix from a single image is an ambiguous task. In the proposed method, the ambiguity in haptic model construction was resolved interactively with simple sketches. By introducing human perception into the interpretation of the object shape through sketch, a part of the haptic shape is determined first, and then based on that, the depth information of the remainder of the shape is extracted for further modelling. Models are created following the distorted contours of the respective parts in the image so that if they are projected onto the image plane under parallel orthographic projection their contours would match. Therefore, after creating the distorted haptic models with sketch input, the models can be directly projected using parallel orthographic projection for alignment. Here is an example in Figure 3-10: the haptic fence model is constructed based on the white strokes which are drawn following the perspectively distorted fence contours in the image. Instead of the cuboid-shape fence structure, truncated pyramid models are obtained and projected onto the image plane under parallel orthographic projection. Since orthographic projection does not distort the projection contours, the contours of the truncated pyramid models would overlap with the fence contours. All the truncated pyramid models are initialized with their reference point on plane $z = 0$. Their depth information is adjusted afterwards to keep the occlusion effect, e.g., the pyramid model A is partially hidden by pyramid model B .

Figure 3-10: An example to illustrate the projection of the haptic models constructed from the strokes.

Here the model workspace is defined based on the haptic workspace in a way that the origin of the two workspaces overlap. In this manner, if the haptic cursor, the 3D virtual representation of the HIP, is also projected onto the image using parallel orthographic projection, its position can be directly used for collision detection with the haptic models.

3.4 Haptic rendering of the augmented scene

Haptic interaction with the augmented haptic models includes tasks of collision detection and force rendering. In each haptic frame, the position of the haptic cursor, i.e. the proxy of the Haptic Interface Point (HIP), is also computed and sent to the simulation engine for visual display.

3.4.1 Collision detection and proxy calculation

Collision detection with the FRep-defined solids could be simply done by substituting the coordinates of the HIP into the FRep function and checking its value. A value no less than zero means a collision being detected. When FReps are not the best solution for making

haptic models, other representations can also be used for defining the models and converted into polygon meshes for haptic rendering. If the polygon meshes can be directly rendered by a readily available API like the HLAPI (High-Level Application Programming Interface) provided by the OpenHaptics® toolkit, two independent threads with different refresh rates (i.e. 1000Hz for dealing with implicitly defined objects and 100Hz for polygon meshes) can be set up to deal with collision detection with FRep-defined objects and polygon meshes separately. As long as the two refresh rates are both high enough to provide consecutive haptic feedback, a smooth switch between the two types of haptic models could be ensured.

Figure 3-11: An ambiguous situation for proxy calculation.

In a visual-haptic interactive scene, a proxy is calculated to indicate the position of the HIP. When the HIP moves in the free space, the proxy and the HIP overlap with each other. When the HIP collides with the haptic model in a simulation of rigid-to-rigid collision, the proxy locates at a position on the surface of the model which is nearest to the HIP in the direction of the penetration gradient. In the proposed method, the nearest point on the surface is obtained using the subdivision algorithm from [32]. In this algorithm, it is necessary to find a point outside the haptic model in the direction of the gradient as a starting point for subdivision. However, when the HIP is moving in the vicinity of intersection of two haptic models, if the space between two neighboring surfaces is very small (as shown in Figure 3-11), we may never find such a point that makes the function value less than a

certain threshold with a fixed increment in the gradient direction and this will get us trapped in the loop. To avoid such a case, in the proposed method a kinetic model is used to help predict the proxy position in the vicinity of the intersection part. Since haptic refresh rate is high (1000 Hz) and the motion of the HIP is continuous, there is a little increment between the positions of the HIP at two consecutive frames. Thus, the proxy in the previous frame could be used as a reference point. When subdivision method does not work, a point, which is close enough to the proxy in the previous frame, is used to approximate the proxy in the current frame. After determining the proxy position, it will be used for computing the penetration depth and sent to the simulation engine for visualization.

3.4.2 Force rendering

The force feedback rendered from colliding with the surface of the haptic model consists of three components: constraint force (normal force), friction force and texture force. The constraint force $\vec{F}_n$ is generated based on the geometry of the haptic model. It follows the direction of the gradient with a magnitude proportional to the penetration depth of the HIP below the surface of the model. This penetration depth is calculated as the distance between the HIP and the proxy in the proposed method, and the coefficient multiplied by it is defined based on the stiffness of the model. Suppose that $G(x, y, z)$ is the shape function and the HIP contacts with it at point $p(x_0, y_0, z_0)$ and the proxy point on the shape surface for point p is p' then the force direction at point p is a unit vector in the direction of gradient at this point and the magnitude of constraint force is:

$$\nabla \hat{G}_x(p) = \lim_{\delta \rightarrow 0} [(G(x_0 + \delta, y_0, z_0) - G(x_0, y_0, z_0)) / \delta] \quad (3.5)$$

$$\nabla \hat{G}_y(p) = \lim_{\delta \rightarrow 0} [(G(x_0, y_0 + \delta, z_0) - G(x_0, y_0, z_0)) / \delta] \quad (3.6)$$

$$\nabla \hat{G}_z(p) = \lim_{\delta \rightarrow 0} [(G(x_0, y_0, z_0 + \delta) - G(x_0, y_0, z_0))/\delta] \quad (3.7)$$

$$\|\overrightarrow{F_n}\| = k\|p - p'\| \quad (3.8)$$

$$\overrightarrow{F_n} = \frac{\nabla \hat{G}(p)}{\|\nabla \hat{G}(p)\|} \cdot k\|p - p'\| \quad (3.9)$$

The k -value in Equation 3.8 stands for the stiffness of the haptic model. In the implementation, this value is affected by the maximum exerable force of the haptic device. For example, for a hard and stiff interactive model, the k -value needs to be large to generate a strong resistance force. However, at the same time, it cannot exceed the threshold set by the maximum exerable force of a haptic device, otherwise there will be large kick back force. Therefore, it is necessary to check and adjust the k -value before using a different haptic device.

Generally, the gradient of the implicit model at a point can be obtained based on Equations 3.5 to 3.7. However, if the haptic shape is formed as the union or intersection of two implicit models, due to the precision constraints of programming, it may happen that the gradient is miscalculated as zero when the HIP is inside the shape. For example, suppose that there are two haptic models A and B and point p is inside model A but not in B. It is possible that the value difference of functions G_A and G_B at point p is so big that $G_{A \cup B}(x_p, y_p, z_p)$ is equal to $G_{A \cup B}(x_p + \delta, y_p, z_p)$ after the exceeded digits being automatically cut off. To avoid this problem, all the functions are normalized before the union or intersection operation. Besides, when the value of a function is negative, -1 is used instead of the original value.

Figure 3-12: Feedback force model where $\vec{f}_v$ is optional.

For friction component, when the stiffness coefficient of the haptic model exceeds a certain threshold, the model is considered stiff and only static and dynamic friction $\vec{f}_d$ is applied to the HIP while sliding on the surface of the model, otherwise both viscous friction $\vec{f}_v$ and static and dynamic friction $\vec{f}_d$ are generated. Figure 3-12 shows the feedback force model used in the implemented pilot system. Friction $\vec{f}_v$ always works to slow down the motion of the intrusion, thus its direction is against that of velocity. With the known $\vec{F}_n$ and the direction vector of HIP velocity $\mathbf{v}$, the direction of $\vec{f}_d$ can also be obtained based on the facts that $\vec{f}_d$ is on the same plane with $\mathbf{v}$ and $\vec{F}_n$ and that it is always perpendicular to $\vec{F}_n$. Furthermore, the magnitude of $\vec{f}_d$ is a function of $\|\vec{F}_n\|$, which is also the pressure of the HIP onto the surface. In Equations 3.10 to 3.11, $\vec{d}_d$ denotes the direction of $\vec{f}_d$ and μ is the friction coefficient.

$$\vec{d}_d = \mathbf{v} - \|\mathbf{v} \cdot \nabla \hat{G}(p)\| \cdot \nabla \hat{G}(p) \quad (3.10)$$

$$\vec{f}_d = \mu \|p - p'\| \frac{\vec{d}_d}{\|\vec{d}_d\|} \quad (3.11)$$

Besides constraint and friction forces, a texture component can be added for providing subtle tangible geometry details and material properties of the surface. There are several ways of generating haptic texture, such as the image-based methods which calculate texture force based on intensity, color, gradient etc., of the image or using libraries of haptic

texture directly. Texture force allows for making subtle details on the surface tangible without using a complex model with all the geometry details in definition. In this manner, while providing haptic interaction of high precision, the size of the definitions of the models could still be small and easy to share in collaborative virtual worlds. More details of haptic texture generation can be found in Chapter 5.

3.5 Haptic file

After sketch-based modelling of haptic objects is completed, the users could choose whether to store the tangible models in a “haptic” file. Since FRep-defined models could be represented by a number of parameters, each “haptic” file would only take up a few kilobytes, thus it is easy to share across the network. The haptic models are stored in a “haptic” file following the format below:

Shape Parameter 1, Shape Parameter 2, ..., Shape Parameter N, Operation Sign

The shape parameters for each primitive type are listed in Table 3-2. Operation sign indicates the set operator for this model, which is 0 for intersection and 1 for union. For each primitive type, the number of the primitive models is also recorded in the file.

To illustrate how to define a “haptic” file, an example is given in Figure 3-13. The “haptic” file in Figure 3-13(c) is for the pyramid image in Figure 3-13(a) and it contains the definition for 5 planes. Each plane is defined by a normal, a reference point and a signed number which indicates the set operator for the plane. In Figure 3-13(b), each plane is marked with a number and it indicates which line of the plane definitions it corresponds to.

Table 3-2: Parameters for defining haptic primitives.

Shapes	Shape parameters
Planes in a Point-Normal form	Reference point coordinates P_3 Plane normal N_3
Planes in a three-point form	Reference point coordinates P_3 Plane normal N_3
Spheres	Center coordinates C_3 Radius r
Ellipsoids	Center coordinates C_3 Semi-major length l_{major} Semi-minor length l_{minor} Rotation matrix R_{3*3}
Right circular cylinders	Base radius r Height h Rotation matrix R_{3*3} Center translation vector T_3
Right circular (truncated) cones*	Opening angle θ Height h Distance between the vertex and the cone center d Rotation matrix R_{3*3} Center translation T_3 Sign indicating flipping s
Infinite right circular cones*	Opening angle θ Rotation matrix R_{3*3} Center translation vector T_3
Cuboids	Length l Width w Height h Rotation matrix R_{3*3} Center translation vector T_3
Truncated square pyramids	Length of the upper base l_{upper} Length of the lower base l_{lower} Pyramid height h Rotation matrix R_{3*3} Center translation vector T_3

(a)

(b)

(c)

Figure 3-13: An example of a “haptic” file. TAN is used as a file extension for a “haptic” file and each “haptic” file is automatically named after the corresponding image.

It helps to avoid repeating the same sketch by storing the model definitions in a haptic file. If the user is not satisfied with the haptic models defined in the “haptic” file or wants to add more models inside, new models still could be added through sketching.

3.6 Results

The images in Figure 3-14 illustrate how haptic interaction with photos is achieved in the proposed method. Figure 3-14(a) shows how we draw on the iPhone 7 plus image to create a phone model. The main body of the phone is defined with a variational implicit function by sketching on the contour of the phone (40 strokes) and selecting additional points within the contour (4 strokes). Since the proposed method supports CSG modelling, a thin cylinder shape is defined and deducted from the main body to create the slightly concave home

button on the iPhone 7 plus. The speaker at the top of the phone is constructed in the same way as the home button. The whole modelling time was less than 2 minutes, which includes the time for sketching and the time for adjusting the depth values of the cylinders.

(a)

(b)

(c)

(d)

Figure 3-14: (a) Sketching on the image. (b) The haptic model built based on the sketch. (c) Aligning the model with the image. (d) A snapshot of the interactive scene.

After building the haptic model of the iPhone 7 plus, we project it onto the image plane under parallel orthographic projection so that the contour of the phone model matches the respective part of the image. In this connection, the HIP is able to slide along the surface of the phone displayed in the image. The proxy of the HIP, also called the haptic cursor, is displayed as a red sphere in Figure 3-14(d). The red line connected to the sphere shows the force direction, indicating that the cursor is in contact with the haptic model now.

Figure 3-15: Haptic interaction with images augmented with haptic objects. Direction of the force feedback vector is displayed as a red arrow.

More examples of haptic interaction with photos using the Geomagic Touch are given in Figure 3-15. The haptic cursor (displayed as a red sphere) corresponds to the end-effector of the haptic device. We hold and move the handle of the device to control the

position of the cursor in the image scene. A red force vector toggles on when the tangible objects are explored. Particularly, the trace of the haptic cursor is marked with cursor sample points (sampled at 25 Hz) in Figure 3-15(c). Based on the trace, we can see that when the cursor is touching the invisible sides of the tangible objects, there is no visual feedback of the cursor in the image while the object can still be felt.

Figures 3-15(a) and (b) illustrate the two cases considered previously in Figures 3-6 and 3-8. The conical building and the Louvre pyramid models were constructed according to the methods described in Section 3.2.2. The conical building model consisted of one cone and two cutting planes, and the whole sketching and adjusting took around 56 seconds. While drawing, haptic feedback was used to check whether the sketch needs to be adjusted. For example, if it is found that the tangible part is on the wrong side of the cutting plane, then the orientation of the cutting plane should be flipped. The Louvre pyramid model was made by four planes and it took around 2 minutes to build the pyramid and the water surface. Since the sketch-based methods require some drawing skills, the modelling time for each case is given only as a reference. In Figure 3-15(c) a more complex example is shown where the whole photo is made tangible by augmenting it with a Boolean combination of 49 haptic models consisting of spheres, cylinders, cones and planes. Specifically, the trunk of the tree consists of 32 spheres, 2 cylinders and 12 cones smoothly blended together. The water surface and the green background are modeled as a few planes located below and behind the haptic tree model. Here it is noted that every time a haptic primitive is added, we need to touch it to see whether this primitive matches with the image content and then to decide whether to keep or discard it. This process can be very time-consuming for novices, and some practice is definitely necessary. To make the trunk in Figure 3-15(c) tangible, 6 cones

were sketched first to simulate its main structure, which took around 2 minutes. Thereafter more shapes were added to refine the trunk shape and it took us more than 30 minutes to achieve satisfying result. The modelling time largely depends on the modelling requirements. If a precise haptic shape is required, more modelling time is needed. Despite that 49 haptic models are defined for augmenting the image, the haptic file takes merely 4 KB of the disk space.

Figure 3-16: The conference poster is converted to tangible image touched with Geomagic Touch. A few haptic objects are created including implicit surfaces and polygon-based model of the palace. Red vectors in the poster show direction of the haptic feedback.

Finally, haptic interaction with the image augmented with different types of haptic objects is illustrated in Figure 3-16. Here, the image is first used as a guide to augment it using stroke-based sketching with several FRep-defined haptic objects as well as a polygon mesh. For placing the polygon Wavefront .obj model of the Magdalena Palace, only its bounding box was outlined with the strokes. The sphere surrounding the castle is declared as a surface, which can be penetrated through, while other haptic objects are declared as solids. Then,

the image, together with the created haptic file, is visualized and haptically rendered so that different parts of the poster can be felt as 3D objects extracted from the original image.

3.7 Summary

To deal with the situation where there is no available haptic model of the image, image-inspired haptic interaction approach is proposed where the haptic models are interactively defined by user sketch based on the image content. In contrast to common 3D reconstruction, the invisible haptic models are defined in perspectively distorted modelling space and projected onto the image plane with orthographic projection. In the proposed method, function representations (FReps) are mostly used to define the augmented haptic models. In order for smooth haptic interaction with FRep-defined models, an improved haptic rendering algorithm is proposed. Implementations of examples are presented to show the results of the proposed approach.

Chapter 4 Augmenting Images with Reconstructed Haptic Models

Multi-view reconstruction methods such as MVE [98] are able to produce polygon meshes of a complex scene which are sufficient for visualization. However, because of imperfections inherent in passive computer vision techniques, meshes reconstructed in this way usually contain holes, bumpy surfaces and outliers, which cannot be handled easily in haptic interaction. Besides, these meshes often have millions of triangles, which are unevenly distributed based on the image content. For example, if there are several close-up photos of a part of the scene, it is likely to have more details and thus more polygons in this region, resulting in a partially dense model. Collision detection with a large-scale, dense mesh is a challenging task in haptic interaction which requires a high update rate (1 kHz). Simplifying the mesh by decimation can accelerate the collision detection process, but at the cost of losing geometry details on the reconstructed surface. For applications which require simulated tangible perception that exactly matches the displayed geometry, it is necessary to find a haptic rendering algorithm which can maintain both the visual-haptic interactive speed and the accuracy of the haptic feedback.

In this chapter, the scenario where precise polygon mesh of the image scene can be reconstructed is considered and the method for making real life images tangible with reconstructed meshes is presented. Under this scenario, *it is hypothesized that in applications such as digital museums and virtual street walkthroughs, a visual-haptic virtual environment can be simulated by augmenting multiple original images of the real scenes with the reconstructed polygon meshes.* To achieve this goal, an image-inspired

method is proposed to solve the problems of haptic model alignment with multiple images to be displayed as well as smooth interactive haptic rendering of large multi-million polygon meshes, which may have inevitable reconstruction artifacts. Compared to other haptic interaction methods for images, the proposed method is specifically suitable for creating panoramic scenes with high-resolution visual and haptic feedback.

4.1 Overview

Figure 4-1: In the MVE pipeline, Structure-from-Motion (SfM) techniques are used to reconstruct camera parameters and a sparse points set. Then a mesh is reconstructed using Multi-View Stereo (MVS) and Floating Scale Surface Reconstruction (FSSR) approach. In tangible image pipeline, the reconstructed model is matched with corresponding images to provide haptic feedback. Rotation of the scene can be simulated by a series of images.

In the proposed method, the reconstructed models are used to make the corresponding image dataset tangible. Models generated from MVE [98] are used as examples. Given multiple images of a real scene, MVE reconstructs a polygon mesh of the scene, along with the estimated camera parameters for each input image (as shown in the MVE pipeline in Figure 4-1). If the input image dataset contains close-up photos, the output can be a high-resolution 3D scene with millions of polygons and some regions are of a higher resolution than other parts.

In the tangible image pipeline (as illustrated in Figure 4-1), virtual walkthroughs in the real scene were simulated with a series of images from the dataset. Then the reconstructed model is registered with each image using the estimated camera parameters and the respective coordinate transformation. To incorporate reconstructed meshes in the haptic interaction, the worst case scenario is considered in this chapter, i.e. a haptic rendering algorithm for large-scale imperfect meshes is shown. This algorithm is pluggable and can be used for haptic rendering with any large-scale meshes. The proposed tangible image pipeline performs the following tasks:

- **Coordinate transformation.** The haptic display is registered with the photo using the reconstructed camera parameters.
- **Preprocessing.** Duplicate vertices inside the reconstructed mesh are handled and acceleration structures are built for collision detection.
- **Haptic rendering.** A hybrid collision detection algorithm is proposed to handle collision detection with large-scale polygon meshes and how to render force feedback based on the collision results is also explained. The proposed collision detection algorithm is a revised version of the “neighborhood watch” algorithm

[31], which combines the pre-computed connectivity information and spatial partitioning.

These three tasks are discussed in details in the following sections. Section 4.2 explains how to match the reconstructed polygon mesh with the images in a way that the haptic display matches with the image content. Section 4.3 elaborates the way to remove the duplicate vertices inside a mesh and defines the acceleration structure used in the haptic rendering task. Then the third task, haptic rendering, is described in Sections 4.4 and 4.5 in terms of collision detection and force rendering. Finally, examples of the proposed pipeline's results are given in Section 4.6, followed by the comparison experiments in Section 4.7.

4.2 Coordinate transformation

When using images to replace visual rendering of the meshes, the haptic models need to match with the images so that the image content matches the haptic display. In a multi-view reconstruction process, camera parameters of the images can be estimated based on structure-from-motion techniques [99]. Therefore, given a target image and corresponding reconstructed model, the estimated camera parameters could be used to calculate the modelview and projection matrices for projecting the model in the camera frustum. Suppose R_C is the orientation matrix of the virtual camera with respect to the world coordinate system, T_C is the column vector which defines the location of the virtual camera in the world coordinate system, f is the focal length of the camera, img_width and img_height are the width and the height of the given image, pp_x and pp_y are x and y coordinates of the principal point offset of the camera in pixel coordinate system, z_{near} and z_{far} are the z coordinates of the near and far clipping planes, then the 4*4 modelview and projection matrices M_{mol} and M_{proj} can be obtained as follows:

$$M_{mol} = \begin{pmatrix} R_c & T_c \\ 0 & 1 \end{pmatrix} \quad (4.1)$$

$$M_{proj} = \begin{pmatrix} 2f\alpha_x & 0 & 2(pp_x - 0.5) & 0 \\ 0 & 2f\alpha_y & 2(pp_y - 0.5) & 0 \\ 0 & 0 & \frac{z_{far} + z_{near}}{z_{far} - z_{near}} & \frac{-2z_{far}z_{near}}{z_{far} - z_{near}} \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (4.2)$$

$$aspect = img_width / img_height \quad (4.3)$$

$$\alpha_x = \begin{cases} 1, & \text{if } aspect > 1 \\ 1/aspect, & \text{if } aspect \leq 1 \end{cases} \quad (4.4)$$

$$\alpha_y = \begin{cases} aspect, & \text{if } aspect > 1 \\ 1, & \text{if } aspect \leq 1 \end{cases} \quad (4.5)$$

Note that the origin of the image coordinate system for the MVE-produced models is at the top-left corner of the image while it is at the bottom-left corner of the image in OpenGL. Therefore when displaying MVE models in OpenGL, the y-axis needs to be inverted to match the image. This could be done by inverting all elements in the second row of either M_{mol} or M_{proj} .

Three workspaces are involved in the visual-haptic interaction: the camera workspace (defined during the structure-from-motion process), the haptic workspace, and the world coordinate system. The whole mapping and transformation process behind the interaction scene is illustrated in the flowchart in Figure 4-2. The procedures enclosed by the blue dashed lines are for visual rendering. In real 3D scenes, the haptic cursor would be hidden when moving to the back of the objects. To simulate such an occlusion effect with displaying only 2D images, we write the reconstructed models to the depth buffer and then disable writing to the depth buffer right after the writing operation. The depth buffer writing is kept disabled in the following rendering loop. Afterwards, with depth test enabled and `glDepthFunc` depth comparison function set to `GL_LEQUAL`, the depth values of the models

rendered in real-time (e.g., haptic cursor) are compared with the depth values stored in the depth buffer. A pixel of the haptic cursor is only drawn if the incoming depth value at this pixel is less than or equal to the stored depth value. In such a way, if the haptic cursor goes to the back of the reconstructed model (i.e. the incoming depth value is greater than the stored depth value), it is not drawn and the occlusion effect is thus achieved.

Figure 4-2: Flowchart of the mapping process.

In the haptic servo loop thread, the position of the haptic cursor is mapped to the world coordinate system for collision detection and then mapped back to the haptic workspace for force rendering if the collision happens. The generated proxy position is transformed to the camera workspace and sent to the client thread for displaying.

Figure 4-3: Examples of alignment results.

Figure 4-3 illustrates the results of the alignment for the city wall model reconstructed from MVE [98] and the St. Anna Retable model reproduced by Three D Scans

[100]. For the citywall model in Figure 4-3(b), an image dataset is available and the camera parameters are estimated based on structure-from-motion techniques. For the St. Anna Retable sculpture in Figure 4-3(a), only one image is found and the focal length of the camera is also unknown, the camera parameters are just roughly estimated.

4.3 Preprocessing of the reconstructed mesh

In order to apply the collision detection algorithm, there is a need to preprocess the reconstructed mesh, which includes three steps.

The first step is to handle imperfections with regard to duplicate vertices inside the mesh. Reconstructed models are likely to have duplicate vertices, e.g., the city wall model in Figure 4-8(a) has 1883 groups of duplicate vertices. These vertices cause the appearance of holes during haptic rendering leading to pop-throughs during the haptic interaction. The duplicate vertices and zero-area polygons in the mesh are deleted in the following way. All the vertices are traversed to form a list of duplicate vertex groups, and in each group the vertex with the smallest index is considered as effective while the others are deemed duplicates. Then, the polygons with duplicate vertices are divided into two groups. Those with two or more duplicate vertices from the same group (i.e. zero-area facets) are deleted directly, while the others have their duplicate vertices replaced by the effective vertices of the same group.

After removing all the duplicate vertices and zero-area polygons, the second step is to build the connectivity among the vertices, line segments and polygons and store all the neighbors for each primitive. With reference to the “neighborhood watch” algorithm [31], there are three kinds of primitives in a mesh: vertices, line segments and polygons. Thus the concept of *active polygon* is extended to *active primitive*, the primitive that the HIP is in

contact with. Here, we define the *neighbors* for the three primitive types referring to the definitions in [31]:

- For a polygon, the neighbors are its line and vertex components.
- For a vertex and a line, their neighbors include all the polygons connected to it and all the lines and vertices that comprise these polygons.

Figure 4-4 illustrates an example of how neighbors are defined for a vertex, a line segment and a polygon.

Figure 4-4: An illustrative example for neighbor definitions. A vertex neighbor is marked as a small circle, a line segment neighbor is marked in orange color and a polygon neighbor is marked with stripes. (a) The red vertex has 7 polygon neighbors, 14 line segment neighbors and 7 vertex neighbors. (b) The red line segment has 2 polygon neighbors, 4 line segment neighbors and 4 vertex neighbors. (c) The red polygon has 3 line segment neighbors and 3 vertex neighbors.

Based on the connectivity between the vertices and polygons, the vertex normals are recalculated by summing up the weighted normal of the neighboring polygons and normalizing the sum [101] as in Equation 4.6.

$$n_v = \frac{\sum_i \alpha_i * n_{f,i}}{\|\sum_i \alpha_i * n_{f,i}\|} \quad (4.6)$$

Here, the weight is each neighboring polygon's inner angle at this vertex. Besides, whether a line is on a convex or concave surface is also checked and stored. The lines with only one adjacent polygon are marked as edges. These lines may be the edges of the outer contour or the edges of holes on the surface of the mesh.

In the final preprocessing step, a uniform partition is applied to the space within the bounding box of the polygon mesh, dividing this space into cells. The size of the cell is determined by the highest local density of the mesh. To narrow down the search range for active primitive and to meet the real-time requirement, the maximum number of polygons in one cell needs to be constrained. The largest number of polygons in one cell is identified before proceeding to collision detection and the cell size is adjusted based on this number. In this method, a polygon is considered as belonging to one cell if a vertex of the polygon is in this cell, the polygon has an edge intersecting with the bounding box of this cell or the bounding box of this cell intersects with the polygon. This criterion is the same as that in [30].

4.4 Collision detection with the preprocessed meshes

The challenge of collision detection with large-scale meshes is in obtaining the active polygon in real time (1000 Hz). The existence of an active primitive is the necessary and sufficient condition for point-based collision. As illustrated in Figure 4-5, in the proposed method the detection procedure in the current frame is divided into two branches based on the collision status in the immediately preceding frame.

If there is no collision between the HIP and the mesh in the previous frame (the first branch), it needs to be checked whether the ray from the HIP in the previous frame to that in current frame intersects with the mesh. The reason behind it is that when the HIP goes inside of the mesh from outside, intersection always happens. Therefore, based on the intersection test result, this branch is further broken down into two sub-branches:

1. If the ray from the previous HIP to the current HIP intersects with the mesh at one polygon, this polygon is treated like the previous active primitive and serves as start point in the tracking for active primitive in the current frame.
2. Naturally, if there is no intersection then there is no collision in the current frame.

Figure 4-5: Flowchart of collision detection process. P_0 , P_1 denotes the HIP in the previous and the current frame. AP means active primitive.

Correspondingly, if the HIP collides with the mesh in the previous frame (the second branch), then the active primitive in the previous frame is used as a start point to track the path of the HIP and locate the active primitive in the current frame. If the tracking succeeds, it means that the HIP is still in contact with the mesh in this frame. Otherwise it is considered that the contact has stopped.

During the whole process, there are two key modules: the intersection test between the ray and the mesh and the tracking of the active primitive (marked blue in Figure 4-5). More implementation details about these two modules are presented in the following.

4.4.1 Intersection test

In the previous version of the proposed rendering method, the collision detection algorithm is built on the assumption that if the HIP crosses the mesh surface in a frame then the active polygon in this frame would be in the same cell as the haptic cursor. This assumption enables narrowing down the detection range, however, it does not always hold. When it fails, the detection would also fail, resulting in unexpected pop-throughs.

To remove this assumption, ray tracing is introduced into the first branch of the algorithm, splitting this part into an intersection test, which will be described in the following, and a tracking process, which is the same as the process run in the second branch but with different initial values.

For the intersection test, the first step is to check whether the HIP is inside the bounding box of the mesh in the current frame. If it is inside the bounding box, the next step is to locate the cell that the HIP is in. Suppose P_0 is the HIP in the previous frame and P_1 is the HIP in the current frame. If $P_1 \in cell_0$, then based on the connectivity relation between cells, all the cells $\{cell_0, \dots, cell_n\}$ that the ray P_0P_1 passes through are found out. To find the intersected polygon from these cells, $cell_0$ is used as a start point for the intersection test and whether ray P_0P_1 intersects with any of the polygons inside $cell_0$ is checked. If this is not the case, the test continues with checking whether there is an intersection with polygons inside $cell_1$. It is continued like this until the intersected polygon

example the case in Figure 4-7. The ray P_0P_1 intersects with the mesh, but neither P_0 nor P_1 is inside the mesh, i.e. no collision happens. Therefore, after the intersected polygon is obtained, it will be used as start point to track the active primitive. Only if an active primitive exists can it be confirmed that the collision has happened.

Figure 4-7: An example to illustrate difference between intersection and collision.

4.4.2 Tracking of the active primitive

Based on the geometric connectivity built in the preprocessing step, given a start point, it is possible to follow the path of the HIP and track the active primitive. This start point can be a polygon, a line segment or a vertex. It will be referred to as a *start primitive* in the following. The start primitive can be obtained from two sources: the intersected polygon derived from the intersection test or the active primitive in the previous frame.

Three conditions need to be fulfilled to make a primitive active in one frame:

- **HIP criterion:** the HIP is inside the mesh.
- **Distance criterion:** this primitive has the shortest distance to the HIP compared to its neighbors.
- **Projection criterion:** the orthogonal projection of the HIP onto this primitive is inside its range.

Considering the relations between these three conditions, they are examined in the following order: firstly, the primitive which meets the last two conditions is found, then it is checked whether the first condition is true for this primitive.

Algorithm 1: Algorithm for obtaining the active primitive

$activepri_{prior} \leftarrow startprimitive$

/ Given a start primitive, iterate until finding a primitive which meets the Distance and Projection criteria */*

repeat

$activepri_{temp} \leftarrow activepri_{prior}$

$set A = \{activepri_{prior}, \text{neighbors of } activepri_{prior}\}$

for all polygons $\in A$,

if there exists a polygon which has the projection of HIP onto it inside its range **then**
 $distmin = \min \{|dist_i|: \text{the projection of HIP on } polygon_i \text{ plane is inside } polygon_i, polygon_i \in A\}$

$activepri_{temp} \leftarrow \text{polygon with } distmin \text{ to the HIP}$

else

for all line segments and vertices $\in A$,

if $\min\{dist_i: line_i \in A\} < \min\{dist_j: vertex_j \in A\}$ **then**

$distmin = \min\{dist_i: line_i \in A\}$

$activepri_{temp} \leftarrow \text{line segment with } distmin \text{ to the HIP}$

else

$distmin = \min\{dist_j: vertex_j \in A\}$

$activepri_{temp} \leftarrow \text{vertex with } distmin \text{ to the HIP}$

end if

end if

until $activepri_{temp} == activepri_{prior}$

/ Check whether the primitive obtained from the iteration meets the HIP criterion */*

$vector \leftarrow \text{vector from HIP to the projection of HIP onto } activepri_{temp}$

$normal \leftarrow \text{the normal of } activepri_{temp}$

if $vector * normal < 0$ **then**

$collision \leftarrow FALSE$

else

$collision \leftarrow TRUE$

$activeprimitive \leftarrow activepri_{temp}$

end if

The whole tracking process is represented as a repeat until loop operation in the pseudocode Algorithm 1. The loop starts with the determined start primitive. In each iteration, AP_{new} is selected from the input primitive AP_{prior} and its neighbors based on the

distance and projection criteria for being an active primitive. If AP_{new} is the same as AP_{prior} , it would be considered as a potential active primitive and be checked to find whether it meets the last condition, i.e. the HIP criterion. Otherwise, the loop continues with AP_{new} as the input primitive for the next iteration. A primitive that meets all three criteria is the active primitive in the current frame and it will be saved and used as the start primitive for the tracking in the next frame.

In this algorithm, when examining a primitive and its neighbors based on the distance and projection criteria, the features of each primitive type are incorporated into the checking order. For a polygon, if the projection of the HIP is inside its range, then it definitely has the shortest distance to the HIP compared to its components (three line segments and three vertices). The same rule applies to the line segment: If one line segment has the projection of the HIP on it, it certainly has the shortest distance to the HIP compared to its two vertices. Therefore, the distances of the potential active primitive and its neighbors to the HIP are calculated and compared following this order: polygons first, then line segments, and lastly the vertices (reflected in the blue part of the pseudocode).

4.5 Force rendering

In this chapter, it is assumed that the interactive models are hard and stiff objects, therefore constraint-based haptic rendering is applied: a proxy is computed to represent the haptic cursor so that the cursor is always visible. When the HIP is moving in free space, the position of the proxy matches the HIP. When there is a collision, the active primitive is known and the proxy is assigned as the projection of the HIP on the active primitive.

A spring force model is then used. The magnitude of the force feedback is proportional to the penetration depth of the HIP into the active primitive, which is exactly

the *distmin* that we obtain in the iteration loop of Algorithm 1. Normally, the force is computed in the same direction as the facet normal: $\vec{F}_n = \vec{N} \cdot \text{distmin}$. In the proposed method, this equation is used if the active primitive is a polygon. When the active primitive is a line segment, the force is applied along the direction opposite to the movement, which is from the proxy to the HIP position. In this way, the haptic cursor can be prevented from crossing the edges effectively. Thus, if the cursor slides to a hole on the mesh, it would not fall into the hole. The disadvantage of this strategy is that if the cursor slides along a ragged edge, there are frequent changes in the force direction, since we always give the cursor a resistant force perpendicular to the edge. If the force direction is in the same direction as the velocity, this may lead to a cursor jump.

4.6 Results

The images in Figure 4-8 illustrate how the concepts introduced in the previous sections are implemented given a reconstructed model. Figure 4-8(a) shows the original reconstructed city-wall model created by MVE [98], while the small image in the left upper corner is the image to be used for visual display in the interactive scene. Based on the reconstructed camera parameters of this image, the model is transformed to the camera workspace and the part in Figure 4-8(b) is obtained after clipping. It can be seen that the clipped model matches the content of the image (Figure 4-8(c)). After transformation and mapping, the HIP is able to interact with the city wall in the image as displayed in Figure 4-8(d). The red ball in Figure 4-8(d) represents the proxy of the HIP, i.e. the haptic cursor. A red line pointing to the normal direction is also shown, indicating that the cursor is in contact with the model now.

Figure 4-8: (a) The original reconstructed model. (b) The transformed model displayed in simulated camera frustum. (c) The alignment of the transformed model and the image. (d) A snapshot of the interactive scene.

The examples of haptic interaction with the models reconstructed from images (Figure 4-9) can be seen at https://youtu.be/6_tHrG9q3H8. The proposed method enables us to explore the scene by switching between consecutive images forming a walkthrough and touching the image content with the haptic cursor. With the reconstructed mesh superimposed on the images, the images are tangible like real 3D scenes. When the haptic cursor collides with a tangible object in the image, it always stays on the surface of the object as if it is interacting with real rigid objects. When the cursor goes to the back of the object, it would be hidden.

Figure 4-9: Examples of interactions with the models reconstructed from images. The cursor is displayed as a red ball in the interactive scenes.

4.7 Comparison experiments

In this section, the proposed haptic rendering algorithm is compared with God-object renderer [27] provided by H3D API and OpenHaptics HLAPI [21] using feedback buffer. These two renderers are selected because they are commonly used in haptic interaction with meshes.

The comparison experiment was run on a computer with 8-core CPU working at 2.60GHz. In the experiment, decimated versions (1100k and 2200k triangles) of a reconstructed city wall model were used for haptic feedback while an image of the city wall was displayed for visual feedback (Figure 4-10). The decimated models were chosen because both the HLAPI feedback buffer and the God-object renderer could not handle collision detection with the original city wall model. In addition, frustum culling was applied to the decimated models before collision detection procedure. After culling, the triangle numbers for collision detection were reduced to 70.6K and 140K respectively.

Figure 4-10: The image is displayed using OpenGL in the comparison experiment.

Given the same mesh, the comparison is done in terms of the average combined rendering time, the smoothness of the cursor movement, and whether the algorithm works as intended. To make sure that the combined rendering time is recorded under the same conditions, the same renderer (OpenGL) is used for visualization and the haptic cursor is moved along the same path AB as shown in Figure 4-10. Path AB is chosen because it is a depression curve on the surface of the model and thus it is easy to follow. Here, the combined rendering time refers to the time for rendering the proxy of the haptic cursor and the visual medium (images) in the graphics loop. Since the position of the proxy is calculated at the haptic update rate, the haptic rendering performance would also have an influence on the combined rendering time. Therefore, the combined rendering time is determined by both visual and haptic rendering performance. When the visual rendering time is fixed, haptic rendering performance is the only variable which makes the difference. In the experiments, the combined rendering time is collected in the display function of each frame while the cursor is sliding from A to B, and then an average is computed for

comparison. As for the second comparison item—the smoothness of the movement—it is a visual reflection of the combined rendering time. When the combined rendering time exceeds 1 ms, the movement of the cursor would not be smooth any more. Note that the combined rendering time and the smoothness of the movement only make sense if we are able to touch the model (i.e. the third comparison item). If the model is not touchable, we consider this algorithm as failed.

Table 4-1: Results of the performance comparison. The proposed algorithm is referred to as Hybrid CD.

Renderer	Polygon number (K)	Average combined rendering time (ms)	Movement	Touchable
God-object	70.6	0.181	Smooth	Yes (with pop-throughs)
HLAPI	70.6	39.4	Less smooth	Yes (not in real-time)
Hybrid CD	70.6	0.165	Smooth	Yes
Hybrid CD (no culling)	1,100	0.170	Smooth	Yes
God-object	140	0.168	Smooth	No
HLAPI	140	68.5	Less smooth	Yes (not in real-time)
Hybrid CD	140	0.166	Smooth	Yes
Hybrid CD (no culling)	2,200	0.183	Smooth	Yes
Hybrid CD	1,120	0.165	Smooth	Yes
Hybrid CD (no culling)	17,600	0.172	Smooth	Yes

Based on the results shown in Table 4-1, given the smaller model (70.6K triangles) HLAPI feedback buffer gives the worst performance on combined rendering time among the three algorithms. It also reflects visually on the cursor movement, which is why the cursor could not move very smoothly when using HLAPI. Nevertheless, it outweighs God-

object render in collision detection precision. Correspondingly, God-object renderer shows rather good results with the smaller mesh, but it could not work properly with the bigger mesh, so the obtained combined rendering time is not comparable. Comparing to these two rendering algorithms, the proposed algorithm can work with both of the two simplified models and provide effective results.

To validate that the proposed algorithm can handle collision detection with large-scale meshes, the results of the algorithm working with the original-size city wall mesh (1.76 million triangles) are also presented in Table 4-1. For each version of the model, the performance of the proposed algorithm under two conditions—with and without frustum culling—is recorded. Based on the two groups of data, the difference between the average combined rendering times of the algorithm in these two conditions is very subtle. All the times are at a microsecond level. It proves that the proposed algorithm can work effectively with the original model no matter whether culling is applied or not.

Figure 4-11: (a) An image with which the proposed algorithm fails due to the undesired alignment. (b) The corresponding model.

Despite that culling does not accelerate the speed of the proposed algorithm much in the experiment, it can reduce the memory size occupied by our algorithm. Memory size constrains the cell size of spatial partitioning and thus affects the performance of the proposed algorithm in an indirect way. Since in the experiment the cell size is kept

unchanged, it leads to only subtle difference on the average combined rendering time of the algorithm. Besides the cell size, the selected image for display also has an impact on the haptic rendering performance of the proposed algorithm. With different images selected, the camera workspace, the haptic workspace, and the world coordinate system are aligned in a different way. This may lead to an unsatisfactory result with some images. Figure 4-11 (a) is such an example. As can be seen in Figure 4-11 (a), the depth range (i.e. the distance between near and far clipping planes) of the frustum defined by Figure 4-11(a) is long. Since the haptic workspace is fixed, it means that when the haptic cursor moves one unit distance in haptic workspace, it corresponds to at least several unit distances in camera workspace. In this way, more computations are required to track the active primitive, and this results in many pop-through effects when the cursor moves fast unless the cell size is reduced accordingly.

Additionally, the preprocessing time of the proposed algorithm is related to the cell size and the mesh size. In the whole comparison experiment, the cell size is fixed. Under this condition, the average preprocessing time of the proposed algorithm with three different versions of the city wall meshes (1100k, 2200k and 17600k triangles) are 16.2, 23.9 and 263.7 seconds respectively. This phase can be accelerated in several ways, such as using parallel computing or applying frustum culling.

4.8 Summary

An approach to creating tangible images using models reconstructed by multi-view computer vision techniques is presented in this chapter. To deal with large-size, partially dense reconstructed meshes, an improved hybrid collision detection method is proposed, combining the pre-computed connectivity information and spatial partitioning. The

proposed method allows for handling collision detection with meshes of over ten million triangles. Compared with two widely-used open-source haptic renderers (God-object renderer provided by H3D API and OpenHaptics HLAPI), the proposed haptic rendering algorithm shows a superiority in rendering speed and is fast and robust enough to meet the high requirement of haptic update rate. In the proposed approach, the haptic models are aligned with the images in a way that the haptic display would match the visual content. In this manner, a new modality is added into interaction with images.

Chapter 5 Integrating Haptic Perception of Material Properties and Visual Feedback into Haptic Interaction

As stated in Chapter 2, most haptic devices provide only either tactile or kinesthetic feedback due to different generation principles of these two types of feedback. Geomagic Touch, the main interaction tool used in this research, is a kinesthetic desktop haptic device and can only produce programmable kinesthetic feedback. Therefore, while shape information can be delivered through Geomagic Touch, tactile feedback, which allows one to feel the roughness, hardness, friction, etc., of a texture or material, is generally missing in the interaction. To improve the interaction realism, ways to simulate haptic perception of material properties using Geomagic Touch are studied in this chapter with many implementation details provided.

In this chapter, how to properly simulate the occlusion effect between the haptic cursor and the function-based models is also discussed. Visual display which correctly reflects the haptic output is necessary for creating an interactive visual-haptic environment. In the proposed image-inspired haptic interaction approach, images are displayed for visualization while invisible haptic models are aligned with them for haptic interaction. Along with the image, the proxy of the HIP, i.e. the haptic cursor, is also displayed to indicate the position of the end-effector of the haptic device in the interactive scene. When the haptic model is represented in polygon mesh, its depth information could be directly written into the depth buffer and saved for the depth test. However, when it comes to a model defined in functions, the simulation of occlusion effects becomes a problem since function-based models cannot be directly rendered with most rendering software. To solve

this problem, a heuristic method is presented, and on top of that, limited distortion of the images is implemented to make the virtual environment more responsive.

In sum, two main tasks are discussed in this chapter, which are how to simulate haptic perception of material properties and how to add realistic visual feedback in response to haptic input. Section 5.1 focuses on discussing the first task, where the simulation of roughness, hardness, friction and weighing are explained respectively. Section 5.2 gives solution to the second task and illustrates with examples.

5.1 Adding haptic perception of material properties

In addition to geometry information, material properties also contribute to the haptic perception of an object. In the implementations of the proposed image-inspired haptic interaction approach, three main haptic material properties (i.e. roughness, elasticity and friction) are simulated for a more realistic immersion. Besides that, a novel way of simulating weighing objects with Geomagic Touch is also developed and included. In this section, the implementation details are provided and the potential benefits and problems of adding haptic perception of material properties using haptic devices with one haptic interface point are discussed.

5.1.1 Feeling the texture

A texture component can be added in the haptic interaction for providing subtle tangible geometry details on the object surface and/or delivering the surface roughness. Based on the survey presented in Chapter 2, there are several ways of generating haptic texture, such as the image-based methods which calculate texture force based on intensity, color, gradient etc., of the image, using libraries of haptic texture directly and defining the texture force interactively by sketch [2, 36-42, 44]. In this research, haptic texture is simulated with the

image-based method presented in [37] or using the readily available haptic texture libraries such as Penn Haptic Texture Toolkit (HaTT) [66].

In the implementation of image-inspired haptic interaction, when making subtle geometry details on the visible surface tangible, Vsudevan's method in [37] is followed and the texture force is calculated based on the gradient of the image at the HIP. Firstly pixel intensities in a small neighborhood of the HIP are obtained and multiplied by two predefined Prewitt filter masks in horizontal and vertical directions. The entries of the resulting masks are then summed up to get the texture force component. In the presented examples, it is assumed that the invisible rear part of the haptic models has symmetric texture as the visible part and same texture is applied for texture force generation. Here it should be noted that the texture force thus generated is two-dimensional and has no effect on the depth map. Therefore, there would be no big changes in depth when sliding on the surface while the detailed patterns on the surface could still be felt because of the discontinuity in texture force direction. The comparison between interacting with detailed polygon models and rough sketch-based models complemented by image-based texture force is shown in Figure 5-1. In this figure, an FRep-defined sphere is sketched to match the contour of the synthetic image of a football model on the left-hand side. The same football model defined by polygon meshes is located on the right-hand side. The red arrows inside represent the haptic force vectors. It can be seen from the figure that the FRep-defined model could almost provide the same user interaction experience as the polygon meshes except that there is no big change in force direction when crossing a bump or depression. But the bumps and depressions are still perceivable with texture as a supplement for details. This example has been tried by several testers and there were testers giving better feedback

for the FRep-defined model because it could not be penetrated. The “haptic” file for this football is only 68 bytes and the football image is 79.8 KB while the .obj file for the same football model is quite large (1.7MB). Texture force allows making subtle details on the surface tangible without using a complex model with all the geometry details in definition. In this way, while providing haptic interaction of high precision, the size of the definitions of the models could still be small and easy to share in collaborative virtual worlds.

Figure 5-1: A haptic interaction comparison between the proposed image-inspired sketch-based method and the polygon-based interaction approach.

Besides geometry details, texture force can also be generated to deliver a feeling of surface roughness. The release of HaTT [66] made it possible to simulate roughness and friction using measurement-based data rather than hand-tuned model without a basis in physical reality. With the pre-built measurement-based texture models provided by HaTT, texture vibrations can be synthesized in real time and transformed into a texture force that can be rendered by Geomagic Touch. Therefore, in the implementations of surface

roughness in the tangible online shopping interface in Section 6.2, texture models provided by HaTT are directly used to compute the texture force $\vec{F}_t$ resulting from sliding the HIP on the object surface:

$$\vec{F}_t = (ma)\vec{e}_t \quad (5.1)$$

where m is the mass of the Geomagic Touch handle and the user's hand, which is approximated at 50 gram in our implementation, and a is the acceleration value obtained from the texture model based on the current normal force and tangential speed. As to the direction vector of texture force $\vec{e}_t$, it is chosen perpendicular to both the normal force direction vector $\vec{e}_n$ and the friction force direction vector $\vec{e}_f$. Note that all the direction vectors here are unit vectors.

Figure 5-2: Directions of normal force, texture force and friction force.

5.1.2 Haptic simulation of elasticity and friction

The elasticity of a haptic object is represented in the constraint force (i.e. geometry force), which is in the direction of the normal. In the implementations of the constraint force, a simple spring model is applied:

$$\|\vec{F}_n\| = k \cdot d_{penetration} \quad (5.2)$$

where the spring constant k decides the elasticity of a haptic object and $d_{penetration}$ denotes the penetration depth. Different k values can be assigned to different haptic objects for delivering distinct level of elasticity.

In real life, friction force contains the information of moisture and viscosity. When simulating the friction force in this research, only the viscosity of the material is considered.

Two different ways of generating the haptic friction force are implemented. When measurement-based friction models of a material are available from a library, these models are directly applied in friction computation. For example, for each texture, HaTT also provides corresponding measurement-based friction coefficient along with the texture model. The friction force is simulated using Coulomb friction model: $\vec{f} = -(\mu N)\vec{v}_t$, where μ denotes the friction coefficient, N denotes the magnitude of the normal force and $\vec{v}_t$ denotes the direction vector of velocity in the tangential direction (Figure 5-2). In the other situation, the friction force is calculated in terms of three components: static, dynamic and viscous friction. When the stiffness coefficient of the haptic model exceeds a certain threshold, the model is considered stiff and only static and dynamic friction are applied to the HIP, otherwise all three components are added. As depicted in Section 3.4.2, the viscous friction is always in the direction against the velocity and it is calculated based on Coulomb friction model in this research. The direction of the dynamic friction could be obtained based on the facts that the dynamic friction is on the same plane with the velocity and the constraint force and that it is always perpendicular to the constraint force.

5.1.3 Weight simulation

The weight of an object is a physical property that is generally missing in simulated haptic interaction. A novel way of simulating data-driven haptic weight feedback is proposed and implemented in this research to provide meaningful weight information of the haptic objects. The implementation details are as follows.

When using desktop haptic devices, it is assumed that the force output is equal to the force sent to the device. Based on this assumption, the haptic weight feedback can be calculated as:

$$\vec{F}_w = m\vec{g} \quad (5.3)$$

where m is the mass and g is the local gravitational acceleration. Most haptic devices with one HIP, such as Geomagic Touch, are built as linkage-based structures similar to robotic arms. The users feel the force feedback by holding its handle attached to the mechanical linkage. If such devices are used for delivering weight information to the users, the linear relationship described in Equation 5.3 cannot be adopted to define the haptic weight model considering the effect of the linkage and the weight of the handle. Then, the question is whether there is a correlation function describing the relationship between the haptic weight output and the mass input for any given haptic device.

To answer this question, the weight generated by Geomagic Touch and Novint Falcon was measured using a digital scale (5kg Max/1g Resolution). The force output was set according to Equation 5.3, and the device handle was weighed to show the actual mass that the device delivered. For each measurement, the handle was placed on the scale after the scale was returned back to zero to avoid the influence of the previous measurements. After a few trials, it was found out that the position of the scale and the placement of the

handle largely affect the measurement results for both devices, but nevertheless, if the positions of the scale and the handle were kept still during the measurements, the results of Geomagic Touch showed a certain regularity with the input mass while Novint Falcon failed to do it. This observation suggests a possibility to use Geomagic Touch to approximate haptic weight feedback.

To validate this possibility, the weight of the handle for different inputs was measured and recorded with the scale placed right in front of Geomagic Touch. During the measurement, the handle tip was always pointing to the inkwell of the device – its docking socket (Figure 5-3). The measurement results (i.e. the handle's measured mass) were closest to the input masses in this setting.

Figure 5-3: Scale position and handle placement setup during the measurement.

The input mass ranged from 0g to 300g with an increment in 20g. The weight of the handle was measured five times with each input, of which the average was used as the final result. To minimize the influence of the placement and the previous measurements, the handle was lifted up until the scale went back to zero and put back carefully at the same place every time.

The raw mass/measurement data points are plotted and connected as the red curve in Figure 5-4. It can be seen that there is a nearly linear relationship between the input masses and the raw measurement results, although the raw measurement results and the expected mass output do not overlap. If the input mass is x , the red curve of the raw measurement result can be approximately fitted by a linear function $f(x) = kx + b$. To adjust $f(x)$ so that it approaches the expected output, which is $f'(x) = x$, the input mass x should be set to $(cx + d)$, where $c = 1/k$ and $d = -b/k$.

After obtaining the adjustment parameters c and d , the same measurement was carried out again, where x was replaced with $(cx + d)$. As shown in Figure 5-4, the difference between the expected output and the measurement results is largely decreased after the adjustment. It is under the 10 percent Just Noticeable Difference (JND) threshold [102, 103] at most times except the case where the mass is within the range of 0g to 20g and 40g to 80g. It was also found that the difference can be further reduced if the adjustment is applied one more time. Here it should be noted that the graphical interface has an influence on the haptic device output. The measurements of simulated weight were gathered through a command line application without any GUI. As a result, the processor can perform haptic rendering at its max capacity. However, in the user study on tangible online shopping application, the weight feature was integrated into a GUI, which was refreshed at the rate of 50Hz. It was observed that after the integration, although the same algorithm and parameters were used, the force output was lighter than the command line application. To resolve this issue, the same measurements were applied to the integrated application and the adjustment factors were recalculated based on the collected data. The possible cause of this issue is the addition of GUI, which might potentially reduce the haptic rendering

capacity in a multi-threaded application. The implication of this issue is that different local development environments with varying hardware specifications may require different adjusting factors. Nevertheless, the approach to adjust for the deviation is universally applicable.

Figure 5-4: Line charts of expected output, raw measurements and measurements after adjustment.

5.1.4 Application examples and discussion

Simulated haptic perception of material properties allows feeling the difference between materials through touching. It has been added to some of the image-based applications such as arthroscopic surgery simulation and tangible online shopping interface. In the arthroscopic surgery simulation implemented in the previous work [104] of the research group where the author belongs to, the images captured during minimally invasive arthroscopic surgery are segmented automatically to produce a segmentation map. Different haptic material properties are interactively assigned to different parts of the tissues based on subjective feedback from a practicing surgeon about the haptic properties of different

tissues. A highly viscous effect is assigned to simulate the feeling of cutting the tissues. In this way, behavioral realism of the haptic interaction is improved. In the tangible online shopping interface presented in Section 6.2, the texture and friction forces are simulated based on the data-driven models from HaTT. Different texture and friction models are assigned to different haptic objects based on the materials of the objects. It is also necessary to apply different material properties on different parts of an object. For example, the front side of a phone haptic model should be perceived as made from glass while its back side normally has a different texture based on whether it has a phone case or not and the material of the phone case. Besides, haptic weight simulation is also included in the tangible online shopping interface.

All the techniques presented in this section, except the image-based method for haptic texture generation, simulate haptic material properties based on the data-driven toolkits or the real data. However, given that only one interface point is provided in the interaction, the simulated perceptions are still far from the feeling of touching with our hand. Besides, the perceptions of haptic properties are very subjective and they can be influenced by many factors. Take haptic weight as an example. External factors like the way of holding the haptic device handle and the position of the handle all affect one's feelings about the fidelity of the presented haptic weight.

5.2 Adding visual feedback based on haptic input

In a visual-haptic interactive scene created by the proposed image-inspired haptic interaction approach, the tangible images, as well as the haptic cursor, are displayed for visual feedback. As the proxy of the HIP in the virtual scene, the haptic cursor is computed following the solid-to-solid interaction in real life. To further blend the visual and haptic

experiences within the displayed scenes, limited distortion of the images can be created as a complement for visual rendering based on the image content.

5.2.1 Visual rendering of the haptic cursor

The visual display of the haptic cursor indicates the relative position of the end-effector of the haptic device to the haptic models (seen as the image of objects) in a virtual scene. The haptic cursor position is computed by the CPU (together with other haptic rendering tasks) at the rate of 1 kHz. In each graphics frame rendered at 30-60 Hz, the cursor position is read from the haptic callback function for visual display of the cursor. When the HIP moves in free space, the haptic cursor coincides with the HIP. When the HIP collides with the haptic model, i.e. it is inside the model, in a rigid-to-rigid collision, the cursor lies on the surface of the model. The cursor can be seen in the virtual scene as long as the HIP moves inside the camera frustum, except when there is occlusion or the haptic model is penetrative.

The occlusion happens when the HIP is hidden by the solid haptic objects. For example in Figure 3-8, when the HIP contacts with the rear faces of the pyramid (*bce* and *dce*), it is supposed to be invisible while the user still expects to feel the force feedback from the invisible surfaces. If the haptic models are rendered along with the haptic cursor, the occlusion effect is achieved automatically based on the depth test. Since image-based visualization is used in the proposed image-inspired haptic interaction approach, the visual display of haptic models is not necessary. Besides, for most rendering software such as OpenGL, function-based models cannot be directly rendered. The method presented next is for implementing the occlusion effect in the situation where the function-based haptic models are not visualized.

Firstly, the parametric equation is found for the line segment from the viewpoint in haptic workspace to the HIP:

$$x = m_1t + n_1, y = m_2t + n_2, z = m_3t + n_3 \quad p \leq t \leq q \quad (5.3)$$

Then, substitute x, y, z into the inequalities representing the geometry of the haptic model and get the resulting inequality in the form of $r \leq t \leq s$. If $p < r$ and $q > s$, i.e. the range of t for the haptic model is within the bounds of the range in Equation 5.3, the user's view to the HIP is blocked out and the cursor will not be displayed (e.g., the case of ab in Figure 5-5), otherwise the user would always see the movement of the cursor.

Figure 5-5: Illustration of occlusion.

If the haptic model is penetrative, a threshold can be set to decide whether the haptic cursor should be shown or not. When the applied force is smaller than the threshold, the cursor will be shown in the virtual scene, otherwise it will disappear to visually reflect the penetration effect. Since here the interactive object is soft, the position of the haptic cursor is computed based on the penetration depth into the haptic model and the physical properties of the model.

5.2.2 Interactive image distortion

Creating virtual environments requires virtual objects to look and act like real objects. Therefore, sometimes it is expected that the virtual scenes could respond to the movement of the HIP visually. To achieve that, it is proposed to augment images with some rather small data allowing us to visualize the images with elements of animation as well as haptically interact with the augmented patches.

Given any 2D image, the first step is to segment the input image into foreground and background manually. Then simulation is performed at the foreground layer while image in-painting is applied to fill the empty area in the background. After that, the animated foreground layer is composed with the background layer using alpha-matting to create an illusion that part of the scene is in motion. Different simulators lead to different motion effects. For example, the 2D fluid simulator in [105], developed by a colleague from the same research group, can be used to simulate harmonic motions like swaying vegetation, water surface perturbations, or oscillations of elastic objects caused by fluid flow. A paper was published in collaboration with him, where this simulator was incorporated into the interaction with images for simulating harmonic motions. Figure 5-6 shows a still lake image being converted to animated tangible image in this way. The foreground layer which simulates moving trees and water is outlined in Figure 5-6(a), while the visual-haptic interaction effect is illustrated in Figure 5-6(b). Here, when touching the water surface with the HIP of the haptic device, besides the force feedback, the distortion of the water image will be generated, based on the method in [105].

Figure 5-6: Interactively generate deformed images to simulate animation. (a) Mask the parts of the images to be affected by procedural simulation of harmonic motions. (b) Visual and haptic interaction with the water surface. The area enclosed by the white dashed lines shows the distortion of the three shadows created by the 2D fluid simulator in [105].

5.3 Summary

In this chapter, several ways of integrating haptic perception of material properties and visual feedback into haptic interaction are presented. In particular, a novel way of simulating weighing objects with Geomagic Touch is developed. They can be used along with the haptic shapes to create a more immersive tangible image scene. Some of these techniques were adopted in the user studies presented in Chapter 6.

Chapter 6 User Studies

To prove the hypothesis and investigate the feasibility and acceptance of the proposed approach, a user study was conducted with the tangible image interface where multiple images are augmented with the corresponding high-resolution reconstructed mesh. In the test, users were asked to rate their interaction experience with tangible images and the results were evaluated to assess the likelihood that users would accept and want to use this system. Besides, a potential application of image-inspired haptic interaction—tangible online shopping interface—was investigated. A mock-up e-shop with tangible browsing interface was built by aligning the product images with invisible haptic models. A user study was done in which this haptic interface was compared with a traditional mouse-based interface which has similar layout and operational procedures, and the comparison results were analyzed to investigate whether a haptic device with one HIP can be used in online shopping to compensate the lack of physical touch. Details of these two user studies are given in Sections 6.1 and 6.2 respectively.

6.1 User study on tangible image

“Tangible Image” is a new concept to most people, thus it was found necessary to undertake a subjective user test. In this session the results of this subjective user test are reported, evaluating what the users think about the proposed approach.

6.1.1 Capturing Test Photos

In the reconstruction of the Mathildenhöhe sculpture model, 256 photos were put into the MVE system, including the photos used in this test. The reconstructed model contained around 5 million triangles.

The Mathildenhöhe sculpture photos (Figure 6-1) used in this test were captured by incrementally orbiting a camera around the sculpture. Besides taking photos from normal eye-level viewpoint, photos of the sculpture from high and low viewpoints were also captured. During capturing the camera was always pointed at the central part of the sculpture.

Figure 6-1: Top row: photos taken from high viewpoints. Middle row: photos taken from normal eye-level viewpoints. Bottom row: photos taken from low viewpoints.

Selectively 21 photos were chosen from high, normal eye-level and low viewpoints respectively and put in a 3-row grid to simulate a constrained rotation effect (Figure 6-1). All these chosen photos were preloaded to the tangible image system before the test.

6.1.2 Experimental Setup

The system was run on the same computer (2.60 GHz, 8-core CPU) as the comparison experiments presented in Section 4.7. The users were expected to learn the displayed scene by both visual and haptic interaction with it. The visual interaction was supported as a panoramic rotation of the scene controlled by the left and right arrow keys. With each key

pressed, the respective next image of the captured scene from the image sequence was displayed. Haptic interaction was implemented using Geomagic Touch desktop haptic device placed close to the user's dominant hand (Figure 6-2). The users sat in front of the device and were asked to touch the objects in the scene by moving the haptic cursor displayed in it. The scene could be rotated in 180 degrees counterclockwise to view and touch the objects from different perspective.

Figure 6-2: A beta test participant interacts with the tangible photos.

6.1.3 Experimental design

6.1.3.1 Measurements

A questionnaire as in Table 6-1 was designed to evaluate interaction with tangible images. Based on Presence [106] and [107], four Factors are evaluated in this questionnaire: realism, sensory, comfort and satisfaction.

Among the five questions, the second question contributes to two factors. The correlation coefficient was determined with reference to the presence questionnaire (PQ) developed by Witmer and Singer [106]. In PQ [106], the question about “actively searching” is relevant to two factors – realism and sensory. Besides these two factors, how well a user can use the Geomagic Touch to explore is also largely dependent on whether the user has

used a similar haptic device before. Therefore, we did not want the rating of this question to correlate significantly with the total rating of the system, and the correlation coefficient of this question was set to 0.15. In PQ [106], a correlation coefficient below 0.2 indicates low correlation between the two items. Thus we computed the results for realism and sensory in this way:

$$realism = 0.15 * Q2 + 0.85 * Q1 \quad (6.1)$$

$$sensory = 0.15 * Q2 + 0.85 * Q4 \quad (6.2)$$

Table 6-1: Questions and corresponding factors. These factors are rated on a scale of 1 to 5, where 1 means very negative (not at all), 2 means negative, 3 means neutral, 4 means positive and 5 means very positive (very much).

Question	Factor
How realistic is your haptic interaction with the displayed scenes?	Realism
How well could you actively explore the displayed scenes by touching?	Realism, Sensory
How comfortable do you feel interacting with the displayed scenes?	Comfort
How useful is the haptic feedback in improving your interaction experience?	Sensory
How satisfied are you with your interaction experience?	Satisfaction

6.1.3.2 Procedures

24 users participated in the test, 7 female and 17 male. 1 participant was ambidextrous and tried the system with both hands. 17 of them never used any haptic device. All users were between the ages of 20 and 35 years. The entire test took 20 to 30 minutes to complete. Here are detailed procedures:

1. Demonstration of how to use Geomagic Touch with an example. Proper training is necessary before the test to eliminate the tension of the users, especially for novices.
2. User testing. The users were asked to explore the displayed image scene with the haptic device. Viewpoints and viewing angle can be changed by pressing arrow keys.
3. Filling in the questionnaire.
4. Collection of oral feedback. This step is for gaining a more comprehensive understanding of the ratings. The users' answers are recorded on the questionnaire during the collection.

6.1.4 Results

The results of the questionnaire are summarized in Table 6-2. The goal of this user test is to know what users think of the tangible image system, and more specifically, to assess the likelihood that users would accept and want to use this system. It can be seen from the table that the means of the four factors were all above 3 (neutral), which reflects a positive attitude towards the system. If the true population means are calculated, the results are still positive. Let us consider realism, the factor with the lowest mean, as an example. The 95% confidence interval for its mean 3.46 is 3.05 to 3.87, of which the lower bound is still slightly higher than 3 (neutral).

The relations between the user's gender and the ratings on these five questions are also investigated by performing Fisher's exact test on the contingency tables presented in Appendix A. These 2*2 contingency tables are formed by classifying the number of the users based on their ratings for each question and their genders. The two variables, the user

gender (i.e. male and female) and the rating range (i.e. rating ≥ 3 and rating < 3), determine the row and column categories of these tables respectively. For each table, the corresponding Fisher's exact test result is also displayed along with it side by side. In this test, the null hypothesis is that there is no correlation between the ratings of the questions and the user gender. The computed p -values indicate that the null hypothesis could not be rejected at $p < 0.05$, thus there is no sufficient evidence to prove that the user gender would have an influence on the ratings of the questions in Table 6-1.

Table 6-2: Results of the questionnaire. These factors are rated on a scale of 1 to 5, where 1 means very negative (not at all), 2 means negative, 3 means neutral, 4 means positive and 5 means very positive (very much).

Factor	Mean	Standard deviation	95% Confidence Interval
Realism	3.46	0.98	0.41
Comfort	3.58	1.14	0.48
Sensory	3.64	0.90	0.38
Satisfaction	3.58	1.02	0.43

6.1.5 Evaluation

Comments from users are evaluated and categorized into five groups.

6.1.5.1 Pleasure

Most users found it impressive to feel the depth of the object in the photo, especially when experiencing significant changes in depth, e.g., sliding from a platform away from us to one closer to us (as in Figure 6-3). Besides, we got comments that they enjoyed this user test and would like to try our system again.

Figure 6-3: Example of sliding from far surface to near surface. The haptic cursor trace is marked with cursor sample points (sampled at 20 Hz), which are represented as red balls. The red line always points to the force direction.

6.1.5.2 Force feedback

Most users encountered problems while sliding the cursor on the surface of small structures with large curvature, because they found it hard to constrain the cursor to the surface. Two of them suggested that a zooming operation should be provided so that they could touch small details better. Another user compared this phenomenon to the real life situations and explained it as lack of automatic assistant force from the wrist which we obtain when sliding our finger on a real curve.

Six users expected to feel the physical properties of the objects in the interaction, e.g., stiffness, friction and texture. Constrained by the device, interaction with rigid bodies cannot be realistically simulated, but in the future we could make force feedback more convincing by adding haptic texture to the models and applying friction based on the real material properties.

Another interesting finding from the users' oral feedback is that most of them believe that there is too much roughness at some places which are supposed to be smooth.

This may reflect an unconscious relation between visual feedback and haptic feedback. The users have an expectation about what the haptic feedback should be like based on what they see in the photos. If they do not visually perceive the details that they are touching, they are likely to deny these details and interpret them as unexpected roughness. This partially explains why the average rating on realism is just mediocre (3.46 on a scale of 1 to 5). Based on this, it can be concluded that such a system should not provide haptic details that cannot be perceived by eyes. In addition, the force should be smoothed so that the users do not get frustrated because the HIP is stuck at small surface details.

6.1.5.3 Device

Five users pointed out that they felt tired or uncomfortable holding the handle for a prolonged time and three of them explicitly wrote that this has negative influence on their ratings for satisfaction. While the ergonomics of the device could not be changed, there could be some ways to improve the comfort level, e.g., using some form of cushioning or support for the hand.

Another complaint about the device is that it is not so intuitive, which results from limited force output and only one haptic interface point. These are limitations of such ground-based haptic interfaces. If a body-based haptic interface, such as gloves, suits and exoskeletal devices, is used instead, the user experience could be improved to some extent, but the cost would also increase largely.

6.1.5.4 Usefulness

Most users showed reserved positive attitude towards the usefulness of the haptic feedback in interaction with photos. Only three out of twenty-four users gave negative feedback.

Those who gave positive or neutral feedback believed that having one more dimension of feedback is better than simply viewing the photos. They commented that this system could be useful for people with bad depth perception or if the photo content involves unclear structure. One user also mentioned an inspiring observation: her memory about photos is largely enriched in this way and she can remember the content of the photos better after touching them.

6.1.5.5 Others

Before the test, the users were not informed which part of the photos is tangible, so they needed to explore it themselves. Three users found that only the sculpture part is tangible and commented that they also wanted to touch other objects in the photo background, e.g., trees, houses and cars. Therefore, one of the goals in the future is to make the whole photo tangible or to think of a way to communicate to the users which parts are tangible.

Moreover, it was noticed that two users were confused about what touching feels like at the beginning of the test. After the explanation they knew that seeing the haptic cursor does not indicate the occurrence of contact with the objects in the scene. They would feel the haptic feedback only when reaching the depth of the object with the cursor. This confusion is due to the fact that people are not used to derive depth information in the virtual environment without reference. Therefore, additional training about what it means to touch might be necessary and assistive visual feedback could be helpful.

6.1.6 Conclusion

Despite the limitations of the device (i.e. not so intuitive, feeling uncomfortable if holding the handle for long time), the results of the user study show that the users' attitude towards this interactive environment is between neutral and positive on the rating scale used in this

test. By applying Fisher's exact test on the results, it is also found out that the user gender has no influence on the results. It means that the users generally accept this new interactive approach no matter what their gender is.

This "tangible image" system was built based on the hypothesis that a visual-haptic virtual environment can be simulated by augmenting multiple original images of the real scenes with the reconstructed polygon meshes. Image-inspired haptic interaction method proposed in Chapter 4 was used to augment the photos of Mathildenhöhe sculpture with the reconstructed mesh. Based on the users' feedback, it can be concluded that the provided 3D haptic interaction with the photos was general smooth, since there was no reported pop-throughs or getting stuck from the users. Besides, most users found it impressive to feel the depth of the object in the photos and enjoyed this experience. The results and the feedback from the users have proved the feasibility of the proposed "tangible image" system and the hypothesis as well. In this scenario, the proposed image-inspired approach brings new possibilities into interaction with images and provides a way to enrich images with a touch interface and haptic feedback.

6.2 Tangible online shopping application

In online shopping applications, shape information of a product is mostly presented in the form of images, and the sense of touch is generally missing. To investigate whether the haptic devices with one HIP can help to enhance the shoppers' understanding on the physical properties of a product, a mock-up e-shop with tangible browsing interface was built using the proposed image-inspired haptic interaction methods. In this interface, invisible haptic models are aligned with the product images to provide physical information of the given products – weight, shape and texture. Among them, the shape models are

defined as FRep solid geometric shapes using functions $f(x, y, z) \geq 0$ while pre-built measurement-based texture and friction models from Penn Haptic Texture Toolkit (HaTT) are applied based on the material of the product.

6.2.1 Hypothesis

Peck and Childers [108] classify sensory information extracted from touching the products as: /1/ instrumental information obtained from the physical properties of the products and /2/ hedonic response related to the immediate sensation (i.e. without being processed by the brain) aroused by touching. A person's hedonic response for touching is a subjective element which is difficult to control. It can be influenced by many personal factors such as *Need for Touch*. Excluding the effect of hedonic response, the introduction of data-driven haptic feedback into online shopping is likely to present the shoppers more meaningful information about the physical properties of the products, compared to traditional online shopping environment. In this user study, three types of haptic information—weight, shape and texture—are simulated, which leads to the following hypotheses:

H1: The simulation of haptic weight feedback enhances online shopper's understanding of the weight information of the product.

H2: The simulation of haptic geometric property enhances online shopper's understanding of the shape information of the product.

H3: The simulation of haptic texture properties enhances online shopper's understanding of the texture information of the product.

H4: Haptic feedback improves online shopper's satisfaction on the product browsing experience.

6.2.2 User interface design

To prove the above hypotheses, two mock-up e-shops with two graphical user interfaces—a traditional mouse-based interface and a haptic interface—were made to simulate online shopping product browsing context. Specifically, phone cases of iPhone 7 and iPhone 7 plus were chosen as the target products. In both interfaces, weight, geometry and texture information of a phone with and without a case is provided as a reference to the user:

Mouse-based Interface: Product information is given in the forms of texts and images. Mouse is used for navigation.

Haptic Interface: Product information is given in the forms of texts, images and haptic responses. Geomagic Touch is used for navigation, touching and weighing.

Both interfaces have two tabs, one for shape and texture information and the other one for weight information. The screenshots of these two interfaces are given in Figure 6-4. Despite the difference in interaction tool, the two interfaces have similar layout and operational procedures except for the extra instructions and operations added for haptic interface users. These extra operations include feeling the shape and texture of the products with haptic cursor and weighing the products by holding the handle of Geomagic Touch. The position of haptic cursor (displayed as a red ball in Figures 6-4 and 6-5) corresponds to the handle tip position. Under the shape & texture tab, by moving the handle, the users can slide the haptic cursor on the product surface to feel its shape and texture. The illustrative video embedded into the interface prompts the users how to use Geomagic Touch properly. To minimize the influence of the interface, the design of the mock-up e-shop was deliberately simplified, limiting the provided information to the three types of information

mentioned above only. More details about the interfaces can be found in the video at <https://youtu.be/OJTd51WYsa4>.

Figure 6-4: Screenshots of mouse-based and haptic GUIs.

In the user study, the participants were randomly assigned to either mouse-based interface group (i.e. control group) or the haptic interface group (i.e. experimental group) with no knowledge of existence of the other group. Both groups were asked to browse through the given interface to collect shape, weight and texture information of phone cases and answer the same questionnaire afterwards. In this way, the independence of two sample groups was ensured.

6.2.3 Experimental environment setup

The user study was conducted on a computer with 8-core CPU working at 2.60GHz. For the experimental group, Geomagic Touch was placed to the side of the participant's dominant hand. The distance between the device and the edge of desk was adjusted on a case-by-case basis so that the participants can rest their forearm on the desk while operating the device. For the control group, a mouse was used instead and placed at the same convenient position. All the participants were instructed to adjust the chair to a comfortable position and height. The environment setup for the experimental group is shown in Figure 6-5.

Figure 6-5: Environment setup for the experimental group.

6.2.4 Experiment design

The user study took 10 to 20 minutes for each participant and consisted of five steps:

1. Random assignment of the participant to the control/ experimental group.
2. Demonstration of how to use mouse/Geomagic Touch with an example and introduction of the functionalities of the mouse-based/haptic interface.

3. User testing, where the participants were asked to browse the given interface freely to collect product information after they confirmed their familiarity with the use of the given device.
4. Filling in the questionnaire.
5. Collection of oral feedback.

Table 6-3: Questionnaire for both two groups. All the questions are rated on a scale of 1 to 7 based on a comparison with in-store shopping experience.

Question	Factor
Compared to in-store shopping, my understanding of the weight information of the given products through this interface can be rated as: (1 – very poorly; 7 – as good as in-store shopping).	Weight information
Compared to in-store shopping, my understanding of the shape information of the given products through this interface can be rated as: (1 – very poorly; 7 – as good as in-store shopping).	Shape information
Compared to in-store shopping, my understanding of the texture of the given products through this interface can be rated as: (1 – very poorly; 7 – as good as in-store shopping).	texture information
Compared to in-store shopping, I find this browsing experience satisfying: (1 – totally disagree; 7 – totally agree)	Satisfaction

Based on the proposed hypothesis, a questionnaire was designed in terms of weight, shape and texture information and overall satisfaction as in Table 6-3. To keep the independence for the two sample groups and to make the questionnaire universal, the users were asked to compare their browsing experience with the common real-life in-store shopping. Here, it was assumed that we can obtain the most information about a product in a physical shop. Therefore, for the first three questions, 7 means that the user can get as much product information through this interface as in a physical shop. Based on this, ratings

1 to 7 indicate the amount of information obtained from the interface compared to from a physical shop.

6.2.5 Results

Thirty-three users participated in this experiment, with sixteen of them in the control group and the other seventeen in the experimental group. All users were between the ages of 20 and 35 years and had online shopping experience. The unpaired *one-tail Student's t-test* was used to compare the statistical difference between the average ratings of two sample groups for each question. It was one-tail because H1-H4 are all one-tailed hypotheses which will be accepted only when the rating of experimental group (i.e. haptic interface group) is significantly higher than that of the control group (i.e. mouse-based interface group). The results of the questionnaire are summarized below.

Table 6-4: Average ratings of the two groups for the questionnaire and the t-test results.

	Weight	Geometry	Texture	Overall
Haptics	5.235	5.647	4.176	5.412
Mouse	4.000	4.750	3.938	4.000
t-value	2.376	1.753	0.459	3.563
p-value	0.012	0.045	0.325	0.001
Significance (p<0.05)	Significant	Significant	Insignificant	Significant
Effect Size (Cohen's d)	0.823 (Large)	0.608 (Medium)	N/A	1.235 (Very Large)

As to the ratings for weight, geometry and overall satisfaction, the *p*-values obtained by t-test indicate that the rating differences were significant at $p < 0.05$. Thus, hypotheses H1, H2, H4 can be accepted. In addition, Cohen's *d* was calculated to measure the effect size of the average rating difference for significant results. As to the texture information,

although the result was not statistically significant at $p < 0.05$, the average rating of the experimental group is still higher than that of the control group.

Fisher's exact test is also performed on the contingency tables in Appendix B to study the relations between the user's gender and the ratings on the four questions in Table 6-3 for the experimental group. In the experimental group (i.e. the haptic group), there were 10 female and 7 male users. The 2*2 contingency tables are formed by classifying the number of the users in the experimental group based on their ratings for each question and their genders. Since all the questions were rated on a scale of 1 to 7, the column category is determined based on whether the rating falls in the range of [1, 4) or [4, 7]. The Fisher's exact test results for each table are also presented in Appendix B. In this test, the null hypothesis is that there is no correlation between the ratings of the questions and the user gender. The computed p -values indicate that the null hypothesis could not be rejected at $p < 0.05$, which means that the user gender has no influence on the ratings for the designed haptic interface.

Here it is noted that the ratings from both groups are approximately normally distributed, except for the ratings from the experimental group (i.e. the haptic group) on the weight question. In order to make the results more convincing, Mann-Whitney test, which does not require the assumption of normal distributions, is also performed to compare the weight ratings of the two groups. Here the one-tailed hypothesis is the same, i.e. the simulation of haptic weight feedback enhances online shopper's understanding of the weight information of the product. For the weight ratings of the two groups, the U -value is 75.5 while the critical value of U at $p < 0.05$ is 89. Since 75.5 is not greater than 89, the hypothesis should not be rejected. The result of the Mann-Whitney test indicates a

significant difference on the weight ratings at $p < 0.05$, reaching the same conclusion as the t-test.

6.2.6 Evaluation and discussion

6.2.6.1 Weight

Eight out of sixteen participants from the control group explicitly complained that they could not understand the weight difference based on mere textual description and that they wanted to weigh the products with their hands. As to the participants from the experimental group, they generally commented that the haptic feedback provided them with a better understanding of the weight information as a complement to numerical figures, while at the same time the following problems about the haptic weight simulation were reported.

- **Fidelity:** Seven out of seventeen participants from the experimental group doubted the fidelity of the haptic weight simulation. Three of them pointed out that the weight difference between the phone cases is too subtle to perceive. An explanation to this could be that the weight difference is below Haptic Just Noticeable Difference. For example, the weight difference between a plastic cover and a leather cover is only 3 gram. The other four participants felt that the simulated weight was heavier than the real weight. Although the haptic weight information used in the experiments was extracted from the real data of a product, the perception of weight could be influenced by many factors, such as the way of holding the handle and the handle's position.
- **Unnatural gesture for feeling the weight:** Four participants from the experimental group commented that the gesture for feeling the weight was unnatural. One of them tossed the handle into the air repeatedly to feel the weight,

like how he does it in real life, but the feedback he obtained was not as he expected. The other three suggested to replace the pen-shaped handle of the haptic device with an object of which the shape resembles that of the real product so that the holding gesture would be more natural.

6.2.6.2 Shape

Most participants from the control group rated this question higher than neutral (4.0) because the shape of the given product (i.e. phone case) was familiar to them. Four of them wrote that front and back images were enough for simple geometry like that of the phone cases while for products with more complex geometry it would be better to have additional information such as 3D model or pictures taken from different angles. As to the experimental group, most of the positive feedback was related to the fact that the haptic interface enabled them to touch the parts of the product that could not be seen in the images. Other feedback from the experimental group was listed below.

- **Functionality:** An interesting finding was that some of the participants from the experimental group expected the buttons on the products to be functional. They clicked on the buttons in the image using the haptic cursor and got disappointed because they were not functional. This phenomenon implies that introduction of haptic feedback into online shopping could make the image of products more lifelike.
- **Visual effect:** Three participants from the experimental group found that the size of the product image was different from that of a real product while none of the participants in the control group mentioned about it. One explanation could be that with the haptic feedback the shoppers are more prone to expect seeing

product images which matches the real size of the products. Besides, there were also participants from the experimental group suggesting to display 3D visual models of the products.

6.2.6.3 Texture

Most participants from the control group rated this question higher than neutral (4.0) because the shape of the given product (i.e. phone case) was familiar to them. Four of them wrote that front and back images were enough for simple geometry like that of the phone cases while for products with more complex geometry it would be better to have additional information such as 3D model or pictures taken from different angles. As to the experimental group, most of the positive feedback was related to the fact that the haptic interface enabled them to touch the parts of the product that could not be seen in the images. Other feedback from the experimental group was listed below.

- Fidelity: Comments on haptic texture feedback were very subjective. For example, five participants commented that they were not able to feel the difference between some of the textures while other four participants reported that they were able to feel the difference but they could not tell what the texture was solely through touching with a pen tip. In terms of the roughness aspect, most of the users from the experimental group gave a positive feedback, however, four participants reported that the texture felt machine-generated.

6.2.6.4 Discussion

When the behavior of the participants in both groups was compared, it was found that adding a sense of touch made the products more real to them. The participants in the experimental group expected more realism in the shopping experience, such as that the size

of the product image should match that of the real product and that the product image provides the same functionality as the real product. These were never requested in the control group.

The introduction of haptic weight feedback was also welcomed by the participants based on the statistical results of the user study. However, it must be admitted that given the available haptic technologies and their cost to the end-users, a reliable, user-friendly and affordable haptic interface for online shopping appears unlikely to be built in the near future. Besides, weighing and touching with a stylus is not a natural gesture for the users, while there are no affordable haptic devices of which the design is based on human ergonomics. Although the haptic texture provided in the user study was data-driven, the participants were not accustomed to touch with a stylus, which caused doubts about the fidelity of the feedback.

Based on the results of the user study, it can be concluded that the users expect and enjoy touching the product, despite the limitations of the device. They also value an ability to weigh the products physically. Although it was proved that tangible images can help the users to gain a better understanding on the physical properties of the products, it cannot really be expected that such device with only one HIP will be used in every household in the next few years, even if it is hypothetically distributed for free. A more user-friendly interface and convincing force feedback is expected and required.

6.2.7 Conclusion

A mock-up e-shop with tangible browsing interface was built using the image-inspired haptic interaction method proposed in Chapter 3. Invisible geometric models were generated through sketching and augmented with the images to provide 3D haptic shape

information of the products. The haptic material properties were simulated using the pre-built measurement-based texture and friction models from HaTT. To prove the feasibility and meaningfulness of incorporating haptic feedback into online shopping, a user study was conducted where the results of using a traditional mouse-based interface and this mock-up haptic interface were compared and evaluated. The results have shown that tangible images can help the users to gain a better understanding on the physical properties of the products. Based on the users' feedback from the experimental group, the users expected to touch the products despite the limitations of the haptic device. They appreciated the ability to touch the parts of the product that could not be seen in the images. The results of the user study have also shown that the user gender does not have an influence on their attitudes towards the provided tangible online shopping interface. Therefore, it can be concluded that tangible images can serve as a promising way to add a new modality into online shopping. Nevertheless, given the undeveloped haptic technology, there is still a long way to go to have a user-friendly and reliable tangible interface for online shopping.

Chapter 7 Conclusion and Future Work

7.1 Conclusion

In this research, an image-inspired haptic interaction approach is proposed with the hypothesis that images of real scenes can be converted into a tangible form with the proposed approach so that one could interact with them using desktop haptic devices as if they were real 3D scenes. In order to prove the hypothesis, image-inspired haptic interaction methods with respect to different ways of obtaining the 3D haptic models have been proposed and based on that, the overall hypothesis is divided into two sub-hypotheses:

1. A novel way of interactive augmenting images with haptic models is proposed.

In contrast to common 3D reconstruction of the whole image scene, haptic models are created through sketch in a 3D haptic workspace distorted by perspective projection. The sub-hypothesis is that this method can convert images of real scenes into a tangible form without the necessity of its full 3D reconstruction. In this method, function representations are used to define the augmented haptic models. Novel haptic rendering algorithm is proposed for smooth rendering of combination of function-based models. Besides implementations of examples in Chapter 3, a mock-up e-shop with tangible browsing interface was built using the proposed image-inspired haptic interaction method to validate the sub-hypothesis. This haptic interface was also compared with a traditional mouse-based interface in a user study, and the results of this user study have shown that the product images augmented with 3D haptic information can enhance the shoppers' understanding on the physical properties of a product. This user study has proved the feasibility and usefulness

of the proposed image-inspired haptic interaction methods, thus proving the sub-hypothesis as well.

2. The problem of haptic interaction with the reconstructed real life scenes is solved by replacing the interactive 3D scenes with multiple original images of the real scenes augmented with the reconstructed polygon meshes. The sub-hypothesis is that in applications such as digital museums and virtual street walkthroughs, a visual-haptic virtual environment can be simulated by augmenting multiple original images of the real scenes with the reconstructed polygon meshes. To achieve that, the solutions to interactive haptic rendering of large polygon meshes with reconstruction artifacts and haptic model alignment with the images are provided. In particular, a novel hybrid collision detection method is proposed to meet the real-time requirements for haptical rendering. Several representative examples are given in Chapter 4 to illustrate the feasibility of the proposed method, and on top of that, a user study was conducted with the tangible image interface where multiple images are augmented with the corresponding high-resolution reconstructed mesh. The results of the user study have shown that the users' attitude towards this interactive environment is between neutral and positive on the rating scale used in this test. Based on the comments, most users found it impressive to feel the depth of the object in the photos and enjoyed this experience. The results and the comments of the users have proved the feasibility of the proposed "tangible image" system and the sub-hypothesis as well.

Therefore, it can be concluded that the proposed image-inspired approach presents

a way to create 3D visual-haptic scenes with images under different circumstances. With the presented methods of augmenting images with 3D models, the images of real scenes could become tangible in three dimensions. Despite the limitations of the haptic device used in the studies, the new “touching” modality has made the interaction experience more fun and at the same time provided the users with meaningful 3D haptic information of the objects in the images.

Potentially, the proposed methods can be combined with image-based applications such as virtual street navigation and online shopping to deliver additional information to the users in a tangible form. Although only single-point haptic interaction is simulated in the research, all the proposed methods can be directly used for multi-point contact except for the sketch-based modelling methods presented in Chapter 3.

The proposed methods and results have been published in 2 journal and 3 conference papers, which are listed in Section 1.3.

7.2 Future Work

While the hypothesis has been proved with the research presented in this report, more work should be done in future to make the proposed image-inspired approach more applicable and to provide more realistic haptic interaction with images. Three future research directions are addressed in this chapter – automatic template matching for single-view reconstruction, automatic estimation of camera parameters from single image and physically-based haptic rendering of deformable objects.

7.2.1 Automatic template matching for single-view reconstruction

One issue that needs further research is the automatic template matching for single-view reconstruction. In the proposed image-inspired haptic interaction methods, only simple

primitive templates are provided for template matching while there are many available 3D shape datasets online which can also be used as templates for augmentation. These datasets, such as ModelNet40 [109], contain models of specific categories and are suitable for representing the objects as a whole. When there are plenty of templates, defining unique stroke patterns for each of them is a tedious work and not practical.

Instead of finding the optimal template merely relying on user input, methods have been proposed to detect the edges displayed in the images and load one or several templates which best match the edges from the databases semi-automatically with human-guided annotations [82, 90]. With deep learning networks and some prior information on the image, some methods manage to make the whole process work automatically or with weak supervision [110, 111]. However, human input is still required in edge detection and classification when there is noise or unclear contour lines in the image and more research is needed to handle such images automatically. Besides, the situations where there are multiple objects in the images also need to be considered.

After obtaining the templates, they can be deformed to match the detected contours in the images. Despite that there are many available models such as Active Shape models [112] that can be used for the deformation, how to build a suitable model for each category of images is yet another challenge.

7.2.2 Automatic estimation of camera parameters from single image

When the reconstructed mesh of an image scene is available, it can be aligned with the image for interaction if the camera parameters are known. However, many models captured by 3D scanning are posted online without giving the camera parameters used in the scanning. In order to use those models, camera parameters need to be estimated from the image.

Since the corresponding model is known, an intuitive way is to find pairs of matching points/lines on the model and the image so that camera transformation matrix could be obtained based on the mathematical relationship between the coordinates of the 2D points/lines in the image and their corresponding 3D points/lines in the real scene. Much research has been done on finding the best transformation given a set of 2D/3D point/line correspondences. These correspondences are normally derived based on the geometric constraints, features or the local optimization of the cost function [113-115]. However, in some cases, such as when the 3D model is untextured or when the projection of the 3D model is heavily perspective distorted in the image, accurately locating those pairs of feature points/lines without human labeling is a challenging task. A number of methods have been proposed so far aiming at solving the problem of the lack of feature correspondences, such as simultaneously solving alignment and correspondence [116] and the globally optimal 2D-3D registration approach [117]. This could be considered as a direction for the future research.

7.2.3 Physically-based haptic rendering of deformable objects

In most of the examples presented in the previous chapters it is assumed that the objects are hard and stiff, however, many of the objects in real life are deformable and this will affect the haptic interaction in many ways, such as the way the friction is generated and the position of the haptic interface point (HIP) when in contact with the object. In this thesis, the hardness of an object is only reflected in the constraint force while other effects of deformation are not considered. To be able to present realistic haptic feedback with deformable objects, more customized friction, HIP computation methods, etc., are required. Haptic collision detection with deformable haptic objects is normally computationally

expensive, not to mention the situation that large-scale polygon mesh is incorporated. Besides, ways to visually present the deformation on the objects are also need to be further explored.

Appendices

Appendix A: Contingency tables where the number of the users are classified based on their ratings for each question in Table 6-1 and their genders.

Factor		Male	Female	Total	Fisher's exact test result ($\alpha = 0.05$)
Q1 (Realism)	Rating ≥ 3	14	5	19	p = 0.60804 The null hypothesis cannot be rejected.
	Rating < 3	3	2	5	
Total		17	7		

Factor		Male	Female	Total	Fisher's exact test result ($\alpha = 0.05$)
Q2 (Realism, Sensory)	Rating ≥ 3	14	7	21	p = 0.52964 The null hypothesis cannot be rejected.
	Rating < 3	3	0	3	
Total		17	7		

Factor		Male	Female	Total	Fisher's exact test result ($\alpha = 0.05$)
Q3 (Comfort)	Rating ≥ 3	13	7	20	p = 0.28327 The null hypothesis cannot be rejected.
	Rating < 3	4	0	4	
Total		17	7		

Factor		Male	Female	Total	Fisher's exact test result ($\alpha = 0.05$)
Q4 (Sensory)	Rating ≥ 3	14	7	21	p = 0.52964 The null hypothesis cannot be rejected.
	Rating < 3	3	0	3	
Total		17	7		

Factor		Male	Female	Total	Fisher's exact test result ($\alpha = 0.05$)
Q5 (Satisfaction)	Rating ≥ 3	13	7	20	p = 0.28327 The null hypothesis cannot be rejected.
	Rating < 3	4	0	4	
Total		17	7		

Appendix B: Contingency tables where the number of the users are classified based on their ratings for each question in Table 6-3 and their genders.

Factor		Male	Female	Total	Fisher's exact test result ($\alpha = 0.05$)
Q1 (Weight)	Rating ≥ 4	5	9	14	p = 0.53676 The null hypothesis cannot be rejected.
	Rating < 4	2	1	3	
Total		7	10		

Factor		Male	Female	Total	Fisher's exact test result ($\alpha = 0.05$)
Q2 (Shape)	Rating ≥ 4	7	10	17	p = 1.0 The null hypothesis cannot be rejected.
	Rating < 4	0	0	0	
Total		7	10		

Factor		Male	Female	Total	Fisher's exact test result ($\alpha = 0.05$)
Q3 (Material)	Rating ≥ 4	5	8	13	p = 1.0 The null hypothesis cannot be rejected.
	Rating < 4	2	2	4	
Total		7	10		

Factor		Male	Female	Total	Fisher's exact test result ($\alpha = 0.05$)
Q4 (Satisfaction)	Rating ≥ 4	6	10	16	p = 0.41176 The null hypothesis cannot be rejected.
	Rating < 4	1	0	1	
Total		7	10		

Reference

- [1] S. Rasool and A. Sourin, "Haptic interaction with 2D images," in *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, 2011, pp. 13-22.
- [2] G. Nikolakis, K. Moustakas, D. Tzovaras, and M. Strintzis, "Haptic representation of images for the blind and the visually impaired," *CD-ROM Proc. HCI International*, 2005.
- [3] S. Rasool and A. Sourin, "Tangible images," in *SIGGRAPH Asia 2011 Sketches*, 2011, p. 41.
- [4] B.-K. Han, S.-C. Kim, S.-C. Lim, D. Pyo, and D.-S. Kwon, "Physical mobile interaction with kinesthetic feedback," in *Haptics Symposium (HAPTICS), 2012 IEEE*, 2012, pp. 571-575.
- [5] S.-C. Kim, K.-U. Kyung, and D.-S. Kwon, "Haptic annotation for an interactive image," in *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*, 2011, p. 51.
- [6] K. Salisbury, F. Conti, and F. Barbagli, "Haptic rendering: introductory concepts," *IEEE computer graphics and applications*, vol. 24, pp. 24-32, 2004.
- [7] S. Okamoto, H. Nagano, and Y. Yamada, "Psychophysical dimensions of tactile perception of textures," *IEEE Transactions on Haptics*, vol. 6, pp. 81-93, 2013.
- [8] A. A. Stanley, J. C. Gwilliam, and A. M. Okamura, "Haptic jamming: A deformable geometry, variable stiffness tactile display using pneumatics and particle jamming," in *World Haptics Conference (WHC), 2013*, 2013, pp. 25-30.
- [9] T. Carter, S. A. Seah, B. Long, B. Drinkwater, and S. Subramanian, "UltraHaptics: multi-point mid-air haptic feedback for touch surfaces," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*, 2013, pp. 505-514.
- [10] D. A. Kontarinis, J. S. Son, W. Peine, and R. D. Howe, "A tactile shape sensing and display system for teleoperated manipulation," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, 1995, pp. 641-646.

- [11] O. Bau, I. Poupyrev, A. Israr, and C. Harrison, "TeslaTouch: electrovibration for touch surfaces," in *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, 2010, pp. 283-292.
- [12] J. Pasquero, "Survey on communication through touch," *Center for Intelligent Machines-McGill University, Tech. Rep. TR-CIM*, vol. 6, 2006.
- [13] S. Limited. (2017). *Soundbrenner - World's First Wearable For Musicians*. Available: <http://www.soundbrenner.com/>
- [14] A. Inc. (2017). *Apple Watch Edition - Apple (SG)*. Available: <https://www.apple.com/sg/apple-watch-edition/>
- [15] Lechal. (2017). *Lechal - GPS Navigation, Fitness Tracking & More For Your Fit*. Available: <http://www.lechal.com/>
- [16] C. Pacchierotti, G. Salvietti, I. Hussain, L. Meli, and D. Prattichizzo, "The hRing: A wearable haptic device to avoid occlusions in hand tracking," in *Haptics Symposium (HAPTICS), 2016 IEEE*, 2016, pp. 134-139.
- [17] R. Q. Van der Linde, P. Lammertse, E. Frederiksen, and B. Ruiter, "The HapticMaster, a new high-performance haptic interface," in *Proc. Eurohaptics*, 2002, pp. 1-5.
- [18] B.-K. Han, S.-C. Kim, S. Ryu, and D. S. Kwon, "Haptic Snake: Line-Based Physical Mobile Interaction in 3D Space," in *Haptic Interaction*, ed: Springer, 2015, pp. 273-276.
- [19] R. Gassert, J. Metzger, K. Leuenberger, W. L. Popp, M. R. Tucker, B. Vigar, *et al.*, "Physical Student–Robot Interaction With the ETHZ Haptic Paddle," *Education, IEEE Transactions on*, vol. 56, pp. 9-17, 2013.
- [20] M. T. Inc. (2014). *MPB 6 Degree of Freedom Haptic Interface - Description*. Available: http://www.mpbtechnologies.ca/mpbt/mpbt_web_2009/_en/6dof/index.html
- [21] Geomagic. (2015). *Haptic Devices from Geomagic*. Available: <http://www.geomagic.com/en/products-landing-pages/haptic>
- [22] Novint Technologies Inc. (2012, 2015). *Novint-Products*. Available: <http://www.novint.com/index.php/products>

- [23] Force Dimension. (2017) *Force Dimension - products*. Available: <http://www.forcedimension.com/products>
- [24] B. Siciliano and O. Khatib, *Springer handbook of robotics*: Springer, 2016.
- [25] S. AB. (2012). *Open Source Haptics - H3D.org*. Available: <http://www.h3dapi.org/>
- [26] (2014). *CHAI 3D*. Available: <http://www.chai3d.org/index.html>
- [27] C. B. Zilles and J. K. Salisbury, "A constraint-based god-object method for haptic display," in *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, 1995, pp. 146-151.
- [28] D. C. Ruspini, K. Kolarov, and O. Khatib, "The haptic display of complex graphical environments," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997, pp. 345-352.
- [29] A. Gregory, M. C. Lin, S. Gottschalk, and R. Taylor, "A framework for fast and accurate collision detection for haptic interaction," in *proceedings of Virtual Reality conference*, 1999.
- [30] T. Anderson and N. Brown, "The activepolygon polygonal algorithm for haptic force generation," in *Proceedings of the sixth PHANToM Users Group Workshop, Aspen, CO, USA*, 2001.
- [31] C.-H. Ho, C. Basdogan, and M. A. Srinivasan, "Efficient point-based rendering techniques for haptic display of virtual objects," *Presence: Teleoperators and Virtual Environments*, vol. 8, pp. 477-491, 1999.
- [32] K. Salisbury and C. Tarr, "Haptic rendering of surfaces defined by implicit functions," in *ASME Dynamic Systems and Control Division*, 1997, pp. 61-67.
- [33] L. Kim, A. Kyrikou, G. S. Sukhatme, and M. Desbrun, "An implicit-based haptic rendering technique," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, 2002, pp. 2943-2948.
- [34] Z. Gao and I. Gibson, "Haptic sculpting of multi-resolution B-spline surfaces with shaped tools," *Computer-aided design*, vol. 38, pp. 661-676, 2006.
- [35] J. Hua and H. Qin, "Haptic sculpting of volumetric implicit functions," in *Computer Graphics and Applications, 2001. Proceedings. Ninth Pacific Conference on*, 2001, pp. 254-264.

- [36] J. Li, A. Song, and X. Zhang, "Image-based haptic texture rendering," in *Proceedings of the 9th ACM SIGGRAPH Conference on Virtual-Reality Continuum and its Applications in Industry*, 2010, pp. 237-242.
- [37] H. Vasudevan and M. Manivannan, "Tangible images: runtime generation of haptic textures from images," in *Haptic interfaces for virtual environment and teleoperator systems, 2008. haptics 2008. symposium on*, 2008, pp. 357-360.
- [38] D. K. Pai and L.-M. Reissell, "Haptic interaction with multiresolution image curves," *Computers & Graphics*, vol. 21, pp. 405-411, 1997.
- [39] Y. Ikei, K. Wakamatsu, and S. Fukuda, "Image data transformation for tactile texture display," in *Virtual Reality Annual International Symposium, 1998. Proceedings., IEEE 1998*, 1998, pp. 51-58.
- [40] J. Wu, A. Song, and C. Zou, "A novel haptic texture display based on image processing," in *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*, 2007, pp. 1315-1320.
- [41] J. Li, A. Song, and X. Zhang, "Haptic texture rendering using single texture image," in *Computational Intelligence and Design (ISCID), 2010 International Symposium on*, 2010, pp. 7-10.
- [42] D. Chen, A. Song, and L. Tian, "A novel miniature multi-mode haptic pen for image interaction on mobile terminal," in *Haptic, Audio and Visual Environments and Games (HAVE), 2015 IEEE International Symposium on*, 2015, pp. 1-6.
- [43] T. Ping-Sing and M. Shah, "Shape from shading using linear approximation," *Image and Vision computing*, vol. 12, pp. 487-498, 1994.
- [44] B. Le Mercier and A. Kheddar, "A simple way of integrating texture in virtual environments for haptic rendering," in *Advanced Robotics, 2005. ICAR'05. Proceedings., 12th International Conference on*, 2005, pp. 755-760.
- [45] J. P. Fritz and K. E. Barner, "Design of a haptic data visualization system for people with visual impairments," *Rehabilitation Engineering, IEEE Transactions on*, vol. 7, pp. 372-384, 1999.
- [46] F. Janabi-Sharifi, V. Hayward, and Q. Y. Wang, "Design and implementation of a graphic-haptic display system," *Displays*, vol. 28, pp. 118-128, 2007.

- [47] T. Kagawa, N. Shimamura, H. Nishino, and K. Utsumiya, "A method of haptic interaction for photo-retouch tools," in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, 2007, pp. 2443-2448.
- [48] T. Kagawa, T. Shimamoto, H. Nishino, and K. Utsumiya, "A Method of Image Edition with Haptic Devices," in *Complex, Intelligent and Software Intensive Systems, 2009. CISIS'09. International Conference on*, 2009, pp. 1042-1047.
- [49] D. Lareau and J. Lang, "Haptic rendering of photographs," in *Haptic Audio Visual Environments and Games (HAVE), 2012 IEEE International Workshop on*, 2012, pp. 107-112.
- [50] D. Lareau and J. Lang, "Instrument for Haptic Image Exploration," *Instrumentation and Measurement, IEEE Transactions on*, vol. 63, pp. 35-45, 2014.
- [51] K. Kostopoulos, K. Moustakas, D. Tzovaras, G. Nikolakis, C. Thillou, and B. Gosselin, "Haptic access to conventional 2D maps for the visually impaired," *Journal on Multimodal User Interfaces*, vol. 1, pp. 13-19, 2007.
- [52] R. Iglesias, S. Casado, T. Gutierrez, J. Barbero, C. Avizzano, S. Marcheschi, *et al.*, "Computer graphics access for blind people through a haptic and audio virtual environment," in *Haptic, Audio and Visual Environments and Their Applications, 2004. HAVE 2004. Proceedings. The 3rd IEEE International Workshop on*, 2004, pp. 13-18.
- [53] C. Sjostrom, "Designing haptic computer interfaces for blind people," in *Signal Processing and its Applications, Sixth International, Symposium on. 2001*, 2001, pp. 68-71.
- [54] S. Rasool and A. Sourin, "Image-Driven Haptic Rendering in Virtual Environments," in *Cyberworlds (CW), 2013 International Conference on*, 2013, pp. 286-293.
- [55] M. Minsky, O.-y. Ming, O. Steele, F. P. Brooks Jr, and M. Behensky, "Feeling and seeing: issues in force display," *ACM SIGGRAPH Computer Graphics*, vol. 24, pp. 235-241, 1990.
- [56] S.-C. Kim, A. Israr, and I. Poupyrev, "Tactile rendering of 3D features on touch surfaces," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*, 2013, pp. 531-538.

- [57] S. Wu, X. Sun, Q. Wang, and J. Chen, "Tactile modeling and rendering image-textures based on electrovibration," *The Visual Computer*, vol. 33, pp. 637-646, 2017.
- [58] C. Xu, A. Israr, I. Poupyrev, O. Bau, and C. Harrison, "Tactile display for the visually impaired using TeslaTouch," in *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, 2011, pp. 317-322.
- [59] A. Israr, O. Bau, S.-C. Kim, and I. Poupyrev, "Tactile feedback on flat surfaces for the visually impaired," in *CHI'12 Extended Abstracts on Human Factors in Computing Systems*, 2012, pp. 1571-1576.
- [60] Y. Bayakovski, L. Levkovich-Maslyuk, A. Ignatenko, A. Konushin, D. Timasov, A. Zhirkov, *et al.*, "Depth image-based representations for static and animated 3D objects," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, 2002, pp. III-25-III-28 vol. 3.
- [61] S.-M. Kim, C. Jongeun, R. Jeha, and K. H. Lee, "Depth video enhancement for haptic interaction using a smooth surface reconstruction," *IEICE transactions on information and systems*, vol. 89, pp. 37-44, 2006.
- [62] J. Cha, M. Eid, and A. El Saddik, "DIBHR: Depth image-based haptic rendering," *Haptics: Perception, Devices and Scenarios*, pp. 640-650, 2008.
- [63] A. Leeper, S. Chan, and K. Salisbury, "Point clouds can be represented as implicit surfaces for constraint-based haptic rendering," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 5000-5005.
- [64] Y. Tian, C. Li, X. Guo, and B. Prabhakaran, "Real Time Stable Haptic Rendering Of 3D Deformable Streaming Surface," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 2017, pp. 136-146.
- [65] F. Ryden and H. J. Chizeck, "A proxy method for real-time 3-DOF haptic rendering of streaming point cloud data," *IEEE transactions on Haptics*, vol. 6, pp. 257-267, 2013.
- [66] H. Culbertson, J. J. L. Delgado, and K. J. Kuchenbecker, "One hundred data-driven haptic texture models and open-source methods for rendering on 3D objects," in *Haptics Symposium (HAPTICS), 2014 IEEE*, 2014, pp. 319-325.

- [67] M. Strese, J.-Y. Lee, C. Schuwerk, Q. Han, H.-G. Kim, and E. Steinbach, "A haptic texture database for tool-mediated texture recognition and classification," in *Haptic, Audio and Visual Environments and Games (HAVE), 2014 IEEE International Symposium on*, 2014, pp. 118-123.
- [68] B. N. Li, X. Shan, J. Qin, W. Huang, and N. An, "SenseViewer: A unified rendering interface of visual and haptic cues in medical images," in *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*, 2013, pp. 2209-2212.
- [69] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, *et al.*, "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011, pp. 559-568.
- [70] C. Basdogan, "From 2D images to 3D tangible models: Autostereoscopic and haptic visualization of Martian rocks in virtual environments," *Presence*, vol. 16, pp. 1-15, 2007.
- [71] M. A. Otaduy and M. C. Lin, "Sensation preserving simplification for haptic rendering," in *ACM SIGGRAPH 2005 Courses*, 2005, p. 72.
- [72] S. P. Walker and J. K. Salisbury, "Large haptic topographic maps: marsview and the proxy graph algorithm," in *Proceedings of the 2003 symposium on Interactive 3D graphics*, 2003, pp. 83-92.
- [73] F. A. Van Den Heuvel, "Exterior orientation using coplanar parallel lines," in *PROCEEDINGS OF THE SCANDINAVIAN CONFERENCE ON IMAGE ANALYSIS*, 1997, pp. 71-78.
- [74] N. Jiang, P. Tan, and L.-F. Cheong, "Symmetric architecture modeling with a single image," in *ACM Transactions on Graphics (TOG)*, 2009, p. 113.
- [75] I. Kemelmacher-Shlizerman and R. Basri, "3D face reconstruction from a single image using a single reference face shape," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, pp. 394-405, 2011.
- [76] C. Lu and Y. Jie, "Automatic 3D face model reconstruction using one image," in *Advances in Machine Vision, Image Processing, and Pattern Analysis*, ed: Springer, 2006, pp. 235-243.

- [77] T. Xue, J. Liu, and X. Tang, "Symmetric piecewise planar object reconstruction from a single image," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011, pp. 2577-2584.
- [78] H. Shimodaira, "A shape-from-shading method of polyhedral objects using prior information," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, pp. 612-624, 2006.
- [79] C. Häne, S. Tulsiani, and J. Malik, "Hierarchical Surface Prediction for 3D Object Reconstruction," *arXiv preprint arXiv:1704.00710*, 2017.
- [80] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," in *European Conference on Computer Vision*, 2016, pp. 628-644.
- [81] E. Richardson, M. Sela, R. Or-El, and R. Kimmel, "Learning detailed face reconstruction from a single image," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5553-5562.
- [82] T. Chen, Z. Zhu, A. Shamir, S.-M. Hu, and D. Cohen-Or, "3-Sweep: extracting editable objects from a single photo," *ACM Transactions on Graphics (TOG)*, vol. 32, p. 195, 2013.
- [83] M. Prasad and A. Fitzgibbon, "Single view reconstruction of curved surfaces," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006, pp. 1345-1354.
- [84] L. Olsen and F. F. Samavati, "Image-assisted modeling from sketches," in *Proceedings of Graphics Interface 2010*, 2010, pp. 225-232.
- [85] T. Igarashi, S. Matsuoka, and H. Tanaka, "Teddy: a sketching interface for 3D freeform design," in *Acm siggraph 2007 courses*, 2007, p. 21.
- [86] R. Schmidt, B. Wyvill, M. C. Sousa, and J. A. Jorge, "Shapeshop: Sketch-based solid modeling with blobtrees," in *ACM SIGGRAPH 2007 courses*, 2007, p. 43.
- [87] L. Olsen, F. Samavati, and J. Jorge, "Naturasketch: Modeling from images and natural sketches," *IEEE Computer Graphics and Applications*, vol. 31, pp. 24-34, 2011.

- [88] P. Buchanan, R. Mukundan, and M. Doggett, "Automatic single-view character model reconstruction," in *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, 2013, pp. 5-14.
- [89] L. B. Kara, C. M. D'Eramo, and K. Shimada, "Pen-based styling design of 3D geometry using concept sketches and template models," in *Proceedings of the 2006 ACM symposium on Solid and physical modeling*, 2006, pp. 149-160.
- [90] K. Xu, H. Zheng, H. Zhang, D. Cohen-Or, L. Liu, and Y. Xiong, "Photo-inspired model-driven 3D object modeling," in *ACM Transactions on Graphics (TOG)*, 2011, p. 80.
- [91] C. Yang, D. Sharon, and M. van de Panne, "Sketch-based modeling of parameterized objects," in *SIGGRAPH Sketches*, 2005, p. 89.
- [92] A. Pasko, V. Adzhiev, A. Sourin, and V. Savchenko, "Function representation in geometric modeling: concepts, implementation and applications," *The Visual Computer*, vol. 11, pp. 429-446, 1995.
- [93] T.-i. Ohta, K. Maenobu, and T. Sakai, "Obtaining surface orientation from texels under perspective projection," in *IJCAI*, 1981, pp. 746-751.
- [94] J. Aloimonos, "Shape from texture," *Biological cybernetics*, vol. 58, pp. 345-360, 1988.
- [95] F. Fang and Y. T. Lee, "3D reconstruction of polyhedral objects from single perspective projections using cubic corner," *3D Research*, vol. 3, p. 1, 2012.
- [96] A. Sourin and L. Wei, "Visual immersive haptic mathematics," *Virtual reality*, vol. 13, pp. 221-234, 2009.
- [97] G. Turk and J. F. O'brien, "Variational implicit surfaces," Georgia Institute of Technology 1999.
- [98] S. Fuhrmann, F. Langguth, N. Moehrle, M. Waechter, and M. Goesele, "MVE—An image-based reconstruction environment," *Computers & Graphics*, vol. 53, pp. 44-53, 2015.
- [99] R. Szeliski, *Computer vision: algorithms and applications*: Springer Science & Business Media, 2010.
- [100] (2017). *Three D Scans*. Available: <http://threedscans.com/>

- [101] G. Thürrner and C. A. Wüthrich, "Computing vertex normals from polygonal facets," *Journal of Graphics Tools*, vol. 3, pp. 43-46, 1998.
- [102] S. Allin, Y. Matsuoka, and R. Klatzky, "Measuring just noticeable differences for haptic force feedback: implications for rehabilitation," in *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. 10th Symposium on*, 2002, pp. 299-302.
- [103] B. Kramer, "The effects of active lifting and contact force distribution on weight discrimination," 2010.
- [104] S. Rasool, A. Sourin, and F. Kagda, "Image-driven haptic simulation of arthroscopic surgery," *Studies in health technology and informatics*, vol. 184, pp. 337-343, 2013.
- [105] K. Chen and H. Johan, "A simple method to animate vegetation in images using simulation-guided grid-based warping," in *Computer-Aided Design and Computer Graphics (CAD/Graphics), 2013 International Conference on*, 2013, pp. 244-251.
- [106] B. G. Witmer and M. J. Singer, "Measuring presence in virtual environments: A presence questionnaire," *Presence: Teleoperators and virtual environments*, vol. 7, pp. 225-240, 1998.
- [107] F. Danieau, J. Fleureau, A. Cabec, P. Kerbiriou, P. Guillotel, N. Mollet, *et al.*, "Framework for enhancing video viewing experience with haptic effects of motion," in *Haptics Symposium (HAPTICS), 2012 IEEE*, 2012, pp. 541-546.
- [108] J. Peck and T. L. Childers, "To have and to hold: The influence of haptic information on product judgments," *Journal of Marketing*, vol. 67, pp. 35-48, 2003.
- [109] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, *et al.*, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912-1920.
- [110] T. Durand, T. Mordan, N. Thome, and M. Cord, "Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 2017.
- [111] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for fine-grained category detection," in *European conference on computer vision*, 2014, pp. 834-849.

- [112] T. Cootes, E. Baldock, and J. Graham, "An introduction to active shape models," *Image processing and analysis*, pp. 223-248, 2000.
- [113] P. David and D. DeMenthon, "Object recognition in high clutter images using line features," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 2005, pp. 1581-1588.
- [114] B. Kamgar-Parsi and B. Kamgar-Parsi, "Matching 2D image lines to 3D models: two improvements and a new algorithm," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011, pp. 2425-2432.
- [115] B. K. Srikrishna, U. Musti, and J. Heikkilä, "Geometry based exhaustive line correspondence determination," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, 2016, pp. 4341-4348.
- [116] F. Moreno-Noguer, V. Lepetit, and P. Fua, "Pose priors for simultaneously solving alignment and correspondence," in *European Conference on Computer Vision*, 2008, pp. 405-418.
- [117] M. Brown, D. Windridge, and J.-Y. Guillemaut, "Globally optimal 2D-3D registration from points or lines without correspondences," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2111-2119.