Title	ImpAct: immersive haptic interface exploring direct touch and manipulation techniques for surface computing
Sub Title	
Author	Withanage don, Anusha I.(Inami, Masahiko) 稲見, 昌彦
Publisher	慶應義塾大学大学院メディアデザイン研究科
Publication year	2010
Jtitle	
JaLC DOI	
Abstract	
Notes	修士学位論文. 2010年度メディアデザイン学 第53号
Genre	Thesis or Dissertation
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40001001-00002010-0053

Master's Thesis

ImpAct: Immersive Haptic Interface
Exploring Direct Touch and Manipulation Techniques for Surface Computing

by

Anusha Withana

Submitted to the Graduate School of Media Design
in partial fulfillment of the requirements for the degree of

MASTER OF MEDIA DESIGN

at the

KEIO UNIVERSITY

Academic Year 2010

ImpAct: Immersive Haptic Interface

Exploring Direct Touch and Manipulation Techniques for Surface Computing

by

Anusha Withana

B.Sc., Electronic and Telecommunication Engineering
University of Moratuwa, 2007

Submitted to the Graduate School of Media Design
in partial fulfillment of the requirements for the degree of

MASTER OF MEDIA DESIGN

at the

KEIO UNIVERSITY

Academic Year 2010

Certified by . . . .
Professor. Masahiko Inami
Professor, Graduate School of Media Design
Thesis Supervisor

Certified by . . . .
Asst.Professor Maki Sugimoto
Asst.Professor, Graduate School of Media Design
Thesis Co-Supervisor

Accepted by . . . .
Professor Masa Inakage
Professor and Dean, Graduate School of Media Design

ImpAct: Immersive Haptic Interface

Exploring Direct Touch and Manipulation Techniques for Surface Computing

by

Anusha Withana

Submitted to the Graduate School of Media Design
on August 26, 2010, in partial fulfillment of the
requirements for the degree of
Master of Media Design

Abstract

This thesis explores *Direct Touch* and *Manipulation* techniques for surface computing environments using a specialized haptic force feedback stylus named *ImpAct* (**Im**mersive Ha**p**tic **A**ugmentation for Dire**ct** Touch). Main focus of this thesis is to create a theoretical framework for concept of *direct touch* and translate it to a design concept which can be implemented as a prototype. We propose *ImpAct* as a concept design to implement *direct touch*. *ImpAct* is a stylus which can dynamically change its effective length and equipped with sensors to calculate its orientation in world coordinates. When a user pushes it against a touch screen, physical stylus shrinks and a rendered projection of the stylus is drawn inside the screen giving the illusion that it submerged into the display device. Once user can see the stylus immersed into the digital world below the screen, he/she can manipulate and interact with the virtual objects with active haptic sensations. Furthermore, *ImpAct*'s functionality, design and prototype applications are described in detail with relevance to the concept of *direct touch* giving special attention to design challenges and limitations. Furthermore, a technical evaluation is conducted to measure the accuracy and controllability of *ImpAct*. Thesis concludes by discussing the current limitations and future perspectives of *ImpAct* as a direct touch and manipulation tool.

Keywords: Haptic stylus, Direct touch, 3D manipulation, Surface computing, Through surface interfaces, HCI tools

Thesis Advisor: Professor. Masahiko Inami
Thesis Co-Advisor: Asst.Professor Maki Sugimoto

Table of Contents

Appendices

List of Tables

List of Figures

Acknowledgements

I would like to take this opportunity to thank my supervisor, Prof. Masahiko Inami for all the guidance and support given from the idea formation to the end of the project.
And I am indebt to Asst. Prof. Yasutoshi Makino for giving me the initial idea for the concept, and also, this project would have not be a reality if I haven't had the support and continuous supervision form Asst. Prof. Maki Sugimoto. And also my special thanks goes to JST Matsubara CREST project for funding the project and ease me with the financial burden.
Driving force behind this project was, my two colleagues and members of this project, Mr. Makoto Kondo and Mr. Gota Kakehi. They developed many applications to explore the capabilities of ImpAct. I am highly grateful for their hard work towards the success of this project.
I would like to thank Mr. Charith Fernando and Mr. Yuta Sugiura for helping me in the process of implementation and supporting me in various ways. And my thanks goes to all the members of Inami lab and secretaries for their invaluable help and support.
Also I am grateful to Mr. Takuji Tokiwa for the guidance given in designing the user study. And my deepest appreciation goes to participants of the user study for their time and invaluable feedback.
And I would like to thank Prof. Adrian Cheok for supporting me in various ways to successful completion of my master degree. I am indebt to all the professors in Keio Media Design(KMD) for their insightful feedback during meetings, presentations and demonstrations. And my thanks goes to all the students and members of KMD for various help an supports given to the project and bearing the troubles I cause them in the process.
Last, but not least, I am grateful for my parents and two brothers for continuous caring and encouragement given to me throughout my life.

Anusha Withana
2010-08-01

Chapter 1

Introduction

This thesis explores the *Direct Touch* and *Manipulation* techniques for surface computing environments using a specialized haptic force feedback stylus named *ImpAct* (**Im**mersive Ha**p**tic **A**ugmentation for Dire**ct** Touch). Proposed haptic stylus is a pen shaped device which can change its length when it is pushed against a display surface. Along with this length change, a virtual stylus is rendered inside the display device making user to believe that the stylus penetrated the display surface and went into the shallow region below the screen. Once user can see the stylus immersed into the digital world below the screen, he/she can manipulate and interact with the virtual objects displayed inside the digital world as he/she would use a stick to manipulate objects in a bottom of a pond. Haptic sensation of virtual touches are provided to user's hand via a force feedback mechanism built into the physical stylus. Therefore, *ImpAct* provides an interface which spatially coincide haptic and visual information with multiple degrees of freedom, thus we call it a *Direct Touch* interface.
Contrast to existing interface techniques, direct touch and manipulation provides a broader interaction space and novel design possibilities. Proposed system can be used to improve the user experience of existing surface computing environments and give rise to novel application and interactive techniques. We propose *ImpAct* as a HCI (Human Computer Interaction) *tool* to enable direct touch on existing surface computing platforms.

1.1 Concept of Direct Touch

Human understand and model its environment using information perceived by senses. As Gibson introduced, visual information plays a very important role in human perceptual system[9]. Along with the visual cues, haptic information about the environment plays a major role in understand and interacting with humans surroundings. This holds a major importance

when a human performs a task on the environment which introduce changes to one or more components(objects) from the surroundings. It is obvious that the awareness of the environment is strengthen by the combined effect of multiple sensors.
Direct touch is the way we touch and manipulate objects in the real world. Geometric coordinates of the visual system and the haptic system is perfectly superimposed in the real world. In simple form, if we touch an object in the real world, we can see the surface of object, our hands and the contact point between them. And our cognition can relate to the location of the touch visually and haptically to the same location in the space, which is the contact point between the two surfaces(object surface and skin). Though it is quiet common in the real world, many computer haptic display systems tend to follow an indirect touch approach[24, 26, 32]. This is specially true with haptic systems based on kinesthetic sensation, which are commonly used for object manipulation tasks[21, 30]. Furthermore, direct touch is a very hard to implement concept when it comes to screen based display systems. Our approach is to bring the direct touch techniques to the surface or screen display based computing platforms.

1.1.1 Haptic Perception in Manipulation Tasks

Gibson introduce the theory of affordances explaining how human beings perceive its environment based on different affordances presented by the ambient array of visual information[8]. Let us further study the affordances with a case study of a human subject seeing an object (lets say similar to a tennis ball in size and shape) lying on ground. According to Gibson's theory, since the object is relatively smaller to subjects hand, it presents the affordance of grabbing. In other words, visual information presented is enough to judge that this object is grabbable. And possibly afford picking up too. However, rather than these primary affordances, the object might have other affordances such as affordance to be squeezed or affordance to give a soft feeling to the skin. These information does not contained in the visual array. Furthermore, affordances presented by visual information could be false, for an example object could be too heavy to be picked up or it could be attached to the ground, restricting it to be lifted.
Furthermore, Massie introduce possible information cues that can be per-

ceived when a human hand(or a tool manipulated by human) is moving an object. He categorise them into three classes, *Geometry*(shape, locality, identity), *Attributes*(Constrains, Impedance, Friction, Texture) and *Events*(Constraint change, Contact, Slip)[29]. Manipulation requires visual information to localize(locate the object), approach(make contact with object with hands or tools) and understand the results(manipulation effects can be perceived visually, rotation, movement) of a manipulation task while haptic information such as *attributes* and *events* explain affordances of movement(whether an object can be moved/rotate or not), feedback for motor controls(how much force should be applied) and predictions of future states(prediction of speed of movement, possible threats such as breaking the object or injuries to hand).
Importance of haptic information as much as the visual information to properly understand or perceive the environment is evident. In other words, to properly perceive the affordances of the surroundings, specially in order to manipulate objects, haptic information or haptic perception holds a great importance. Gibson himself stated that the manipulation tasks are very complex to explain solely by visual perception and hard to form regularities[9, chap13].

1.1.2 Haptics in HCI

From the advent of the computational machines, need for seamless interfaces to communicate and collaborate between human and machine has been emphasized. From early days on, researchers has shown the importance of user interface as much as its internal architecture to create a symbiotic relationship between computers and human[28]. With the digital revolution computers became a household artefact and fairly ubiquitous.
Possibility of haptic sensation based interactions are studied under two main disciplines, namely, kinesthetic(force) feedback and cutaneous(tactile) feedback[42]. Kinesthetic sensation actuates muscles and tendons of human body. For example, a rigid wall restricting a human hand gives static force feedback to the hand and a moving pedal in a bicycle gives dynamic force sensation to legs. Cutaneous sensation produce effects on human skin which are captured by the nerve endings, such as texture or heat. For an example, difference between a surface of a sand paper compared to a glass can be

easily pursued by human skin.

1.1.3 Importance of Direct Touch

Concept for the *Direct Touch* for surface display systems can be depicted as shown in the Figure 1.1. Concept describes a way to touch, manipulate and interact with the virtual objects behind the screens directly as if a user's hand can penetrate the surface and sink in to the digital world. Or in other words, the digital world merge with the real world without any barriers in-between. However, this image represents the ideal goal of the direct touch approach.

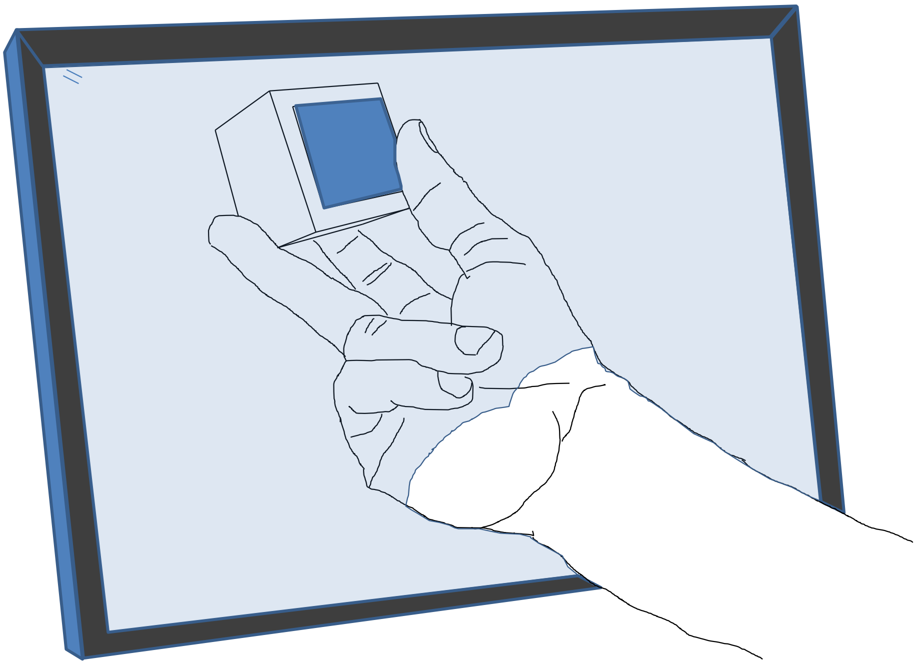

Figure 1.1: Concept for Direct Touch on Screen Display System

It is obvious that any user would agree to the concept shown in the Figure 1.1 to be quiet intuitive and promising as a user interface technology. Not only common sense, but also latest research has shown that there are perceptual links between different sensory events which are spatially coincident. Driver and Spence experimentally showed that human cognitive system perceives multiple sensory information (visual and haptic) as a single unit when their source is spatially coincident[6]. Furthermore, Rob and Tan further extend these findings to the dynamic stage of the stimulation source, concluding perception of multi-modal stimulations coincide on a moving source also provides a combined cognitive effect to the human brain[10]. These researches suggest that the symbiotic nature of multiple sensors in the human body and their perceptive system. Better the perception of the environment, minimum the cognitive load required to take a decision, manipulate actuators and perform a task. Improvements in perceptual system by combined effect of touch and vision has been further explained by Kennett et al[22].

Advantage of the direct touch system is not solely limited to the improved perception and efficiency provided by the system. Direct touch also can improve the expression capabilities of human towards a computer. As shown in the Figure 1.1, Direct touch enable multiple Degrees of Freedom(DoF) within a given three dimensional space. Interaction space is predefined according to the provided display surface and depth of the 3D visualization. Within these borders, user has the freedom of movement, touch and manipulation. Importance of such interactions arise when a user need to express information which are non-explicit in nature. Source of such information arise from the *Tacit Knowledge* of the user, which was explained in the philosophy of Personal Knowledge by Polanyi[36]. This concept was further explained by Nanoka, saying that knowledge is not always can be expressed by quantifiable data, codified procedures and universal principles but it needs subjective insights, intuitions and hunches of human[34]. For an example, lets take a case of an artist carving a piece of wood. It is straightforward that even though the artist is well confident of making the artwork, he/she will not be able to teach(or instruct) a computer to carve the artwork on a virtual model of wood. Skills of carving and the artwork itself can not be expressed with quantifiable data. Such knowledge is technically called *Tacit Knowledge* or *less structured knowledge*. However in a case of an office worker who wants to do his accounting ledgers in a spreadsheet application would be fairly comfortable of instructing the computer with traditional interfaces. This type of knowledge is called *Explicit Knowledge*.

Existence of these two types of knowledge categories has been identified as the *duality of the knowledge*[13]. If we think about the example case introduced in this paragraph with the carving artist, if he/she happen to have a direct touch interface for a computer, it would be fairly superfluous to perform his carving work on a virtual model of wood, utilizing his tacit knowledge in the subject and making use of tools available, and utilizing manipulation and interaction freedom within the digital environment.

1.1.4 Enabling Criterion for Direct Touch

As per the discussion in Section 1.1.3, we can observe two necessary enablers required to make a direct touch interface. They are derived from the two beneficial features of the direct touch for its users; efficient and improved perception of information via spatially coincident visual and haptic display, and possibility to express tacit knowledge based information to a machine. Two enablers can be expressed as follows.

1. **Primary:** *Visual and haptic information should be spatially coincide.*
   In a direct touch interface, presentation sources of visual and haptic information should be located in the same spatial coordinates compared to the user. In case of intervention of a mediation tool(e.g. mouse, joystick, stylus, etc.) between user and the primary interface(e.g. display device, holographic environment, 3D environment created by an HMD), combined system should be able to express the haptic information as it is projected from the active point of interaction between the tool and the digital elements in the visual information display system.

2. **Secondary:** *Multidimensional interaction should be enabled within the digital space provided.*
   Direct touch environment should be provided free-form movements, touch and manipulation within the given interaction space. It could be part of the user(e.g. hands) or a mediating tool(e.g. joystick) which has access to the digital interaction space. However, either user or the tool should have 6 degrees of freedom manipulation possibilities unless otherwise it is limited using a visual or haptic restriction. For an example, a virtual object or a wall could be blocking the movement of hand.

Two enablers given above should be satisfied to provide a direct touch interface. However, the primary enabler is the basic and must to implement requirement. It defines the basic requirement for the direct touch. Secondary enabler can be partially implemented depending on the application and resource availability. However, it is necessary to have the secondary requirement at least partially implemented with predefined limitations.

1.2 Direct Touch and Surfaces

In commercial user interface techniques, screen based display output and key-board/mouse based input became dominant. Early focus given to the development of display technologies as the medium to present internal computational data lead to made modern display technologies to be very mature and near perfect in both hardware and software. Nowadays, Computer Generated (CG) 3D graphics has become as much as realistic as the real world.

Going a step forward from the traditional screen displays, touch and multi-touch based computer platforms, namely surface computing has become the modern trend. Commercial success of multi-touch devices such as iPhone(Apple Inc.) and Android(Google Inc.) mobile platforms and developments of FTIR based multi-touch technologies[12] such as Microsoft Surface(Microsoft Inc. `http://www.microsoft.com/surface`) are promising signs for future dominations. However, touch surfaces pose several limitations in the context of touch, such as limited interactions to the 2D surfaces and lack of physical feedback[49].

HCI researchers are trying to overcome these limitations by adding new features to touch surfaces such as detection of contact area, contact shape, orientation and adding interactions above the surface area to improve the interaction possibilities[1, 48, 49, 52]. Also, many assistive haptic display devices are proposed to work along with touch screen displays[24, 26].

Though there are many haptic feedback user interface techniques available, none of them use direct touch on a surface computing platform. Existing direct touch implementations are mainly built using Head Mounted Display(HMD) technologies or holographic displays[15, 35]. Haptic systems implemented on surface display technologies usually limit the interaction to 2D surface or the haptic device and visual system spatially separated[16, 19, 21, 30, 51]. And majority of them only capable of providing cutaneous sensations[24, 26]. However, since the most popular and widely available computer display system being screen or surface based devices, it is important to explore the possibility of a direct touch technology for such systems.

1.3 *ImpAct*: Immersive Haptic Interface

As briefly introduced earlier, *ImpAct* is a special stylus designed to enable *Direct Touch* for touch screen based display devices. ImpAct consist of a scalable stem, making it possible to change its effective length when it is push against a screen. Simultaneous to this changes in the length of the physical stylus, a virtual stylus is rendered(drawn) inside the digital space below the screen surface along the axis of the physical stylus(Simulated Projection Rendering, see more at Section 3.1). This will deceive the user that the physical stylus penetrate the screen surface and went into the virtual space below the screen. This process is shown in the Figure 1.2. From the perspective of user, it presents a visually continuous interface from the physical world to digital world.

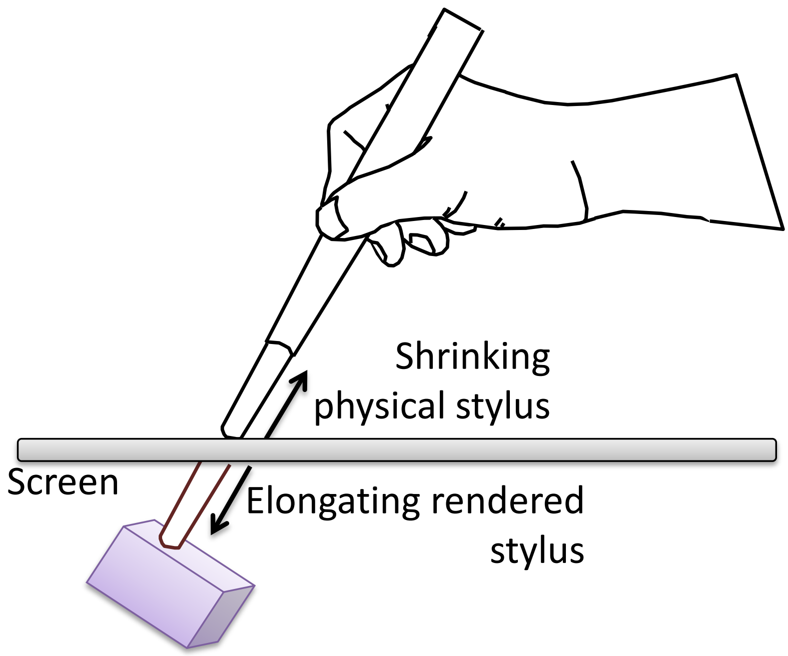

Figure 1.2: Operating Principle of *ImpAct*. When a user push *ImpAct* against the screen, physical stylus shrinks and the virtual elongates. User can use *ImpAct* to directly touch and manipulate objects inside the screen.

Stem of the *ImpAct* is created using two co-centric cylindrical shafts, one hollow(like a tube) and the other solid, making the solid shaft can linearly move inside the outer tube(grip). User grips *ImpAct* using this outer tube(we may call the grip) allowing the inner shaft or the moving shaft to be movable within the grip making the physical stylus to change its length. back end of the moving shaft is internally attached to a DC(Direct Current) motor via a rack-pinion type transmission mechanism. This configuration is shown in the Figure 1.3. DC motor can restrict the movement of the inner shaft, and also it can forcibly move the inner shaft through the gear mechanism. This can be utilize to implement a force-feedback haptic interface using *ImpAct*. For an example, if tip of the *ImpAct* virtual stylus hit a rigid wall inside the screen(i.e. a digital object), applying restriction to the moving shaft will stop user from pushing it further down the screen. Furthermore, if there is a moving object, *ImpAct* can simulate the effect of motion against the tip of the *ImpAct* by forcibly elongating or contracting the length(i.e. moving the inner shaft in either direction).

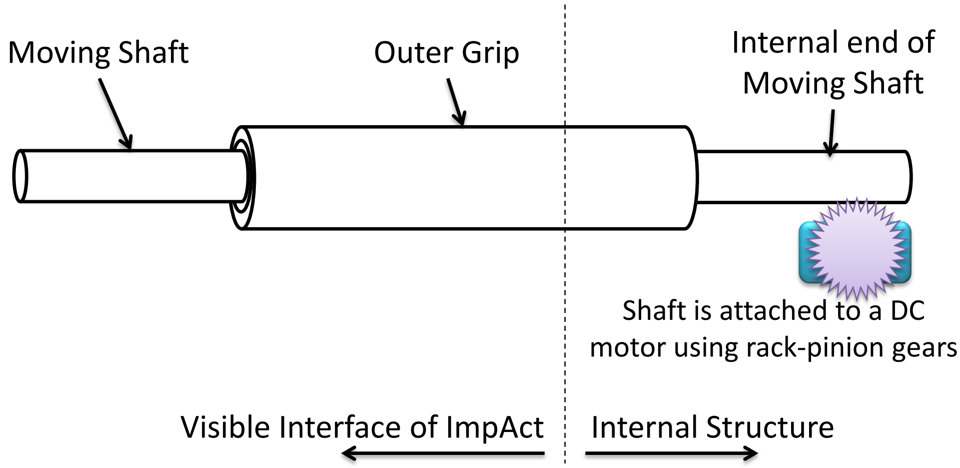

Figure 1.3: *ImpAct* consist of two co-centric cylindrical shafts, one hollow(like a tube) and the other solid, making the solid shaft can move inside the outer grip. Back end of the moving shaft is attached to a DC motor to provide haptic feedback to the user.

ImpAct's capability to provide a visually continuous interface to the digital world and haptic sensation which spatially coincide with the visual display system enables it to satisfy the primary criteria(see Section 1.1.4 for criterion) to become a *Direct Touch* tool for HCI. Therefore *ImpAct* is eligible to be considered as a *Direct Touch* tool if we can satisfy the second criteria.

Lets take a case where *ImpAct* is used to interact with a flat touch screen display. To understand the geometry, take the surface of the touch screen as x,y plane and the z axis is directed inward the screen. Changes in the effective length of *ImpAct* are considered as the z axis controls on the touch screen. Touch sensors on display itself provides measurements to understand the movements along x, y plane using the touch point of *ImpAct* and the screen. Further, *ImpAct* is equipped with an accelerometer-magnetometer pair to calculate its orientation and angular measurements such as yaw, pitch and roll. These angular measurements are added to capture 6-DoF transitions as input variables for interaction. When users push *ImpAct* against the screen and went into the display area, he/she has the freedom to manipulate it using movements along x,y and z directions and rotations such as yaw, pitch and roll. This interaction space is limited by the dimensions of the display screen and the maximum span of the *ImpAct* inner shaft movement. It defines a three dimensional digital space, within that user can freely use the movements and rotations based manipulation and interactions with active force-feedback haptic sensation.
At this point, given the spatially coincident visual and haptic display and the free and multiple degrees of freedom interaction space provided by *ImpAct*, we can conclude that it satisfies both primary and secondary criterion required to become a *Direct Touch* interface for HCI (see Section 1.1.4 for criterion). Figure 1.4 shows the prototype of *ImpAct* on a users hand and the illusion of penetrating the display surface and going into the digital space using Simulated Projection Rendering.

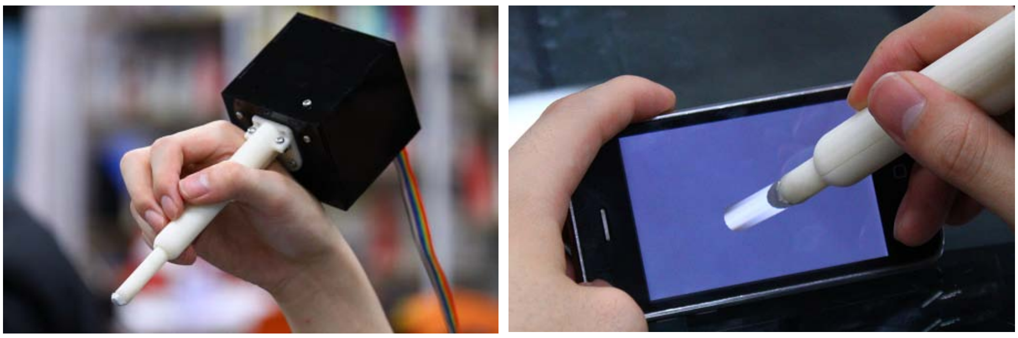

Figure 1.4: Prototype of *ImpAct*. Left: A user holding *ImpAct* in his hand. Right: View of combine effect of the physical length change along with the virtual stylus to make the effect of penetrating into the digital space.

1.3.1 ***ImpAct* as an HCI Tool**

There are many different perspectives in which we can study usage of computers and there peripherals. In 1988, Kammersgaard presented four different perspectives to study the design and usage of human computer interaction, namely, Systems perspective, Dialogue partner perspective, Tools perspective and Media perspective[20]. According to Kammersgaard, in tools perspective, computers(or its applications) are supposed to provide a set of tools of which user has the full control of and can apply them to materials exist in the computer application to make more refined products. *ImpAct* fits into this category so beautifully and our further analysis of usage and user perspective of *ImpAct* can be done considering it as a tool for human computer interaction.

Tools are not designed to automate a process or a part of it. Kammersgaard further explained that skilled users in a particular field should be able to effectively and efficiently use the tools provided in a computer system. And also, user should be able to select from multiple tools if the task requires it. We can use the case of the carving artist in the context of tools. He/she is supposedly a skilled person in the relevant field and can produce a good artwork with real world tools. We can make rendered end of *ImpAct* to imitate carving tools such as *palm handle, gouge, straight veiner, chisel, etc.* (names of carving tools) needed by the artist so that he can use them as real world tools. Since *ImpAct* is capable of direct touch and manipulation, we can expect that the artist will be able to use it as an effective and efficient tool.

Another interesting point to consider when using a tool is awareness. Polanyi pointed out the existence of two kinds of awareness in his book Personal Knowledge in 1959[36, pp57-59]. He named them *focal awareness* and *subsidiary awareness*. He explained these two types of awareness in the point of view from a human using a tool, specifically a hammer. He said that when someone uses a hammer, his/her primary focus (*focal awareness*) is given to hammer head striking a nail rather than his palm holding the hammer. In general, when using a tool, interface between the tool and the material(on which we apply the tool) is the *focal point of awareness* and the interface between the tool and human is the *subsidiary point of awareness*[20]. Lesser the intellectual attention needed for *subsidiary point* *of awareness*, better the tool. Ideally, interface should disappear at the usage of tools. Skilled users of computer mouse might experience this while they are working on 2D applications such as image editing. However, in 3D applications, this is not true, somehow, complexity is added in the process of translation from 2D mouse input to 3D workspace. It is interesting to examine how *ImpAct* could contribute to decrease the amount of attention needed to *subsidiary point of awareness*. As shown in the Figure 1.5, *ImpAct* provides a familiar interface tool for users, such as a writing pen. Further, in probing the cylindrical structure shown in figure 1.5, arrow 'A' shows the line where users primary focus is given to, while he/she can pay little attention to the subsidiary point (shown in arrow 'B') specially because of skills gained using day-to-day tools such as pencils, paint brushes or industry specific tools (such as a scalpel in surgery) in there particular field of interest.

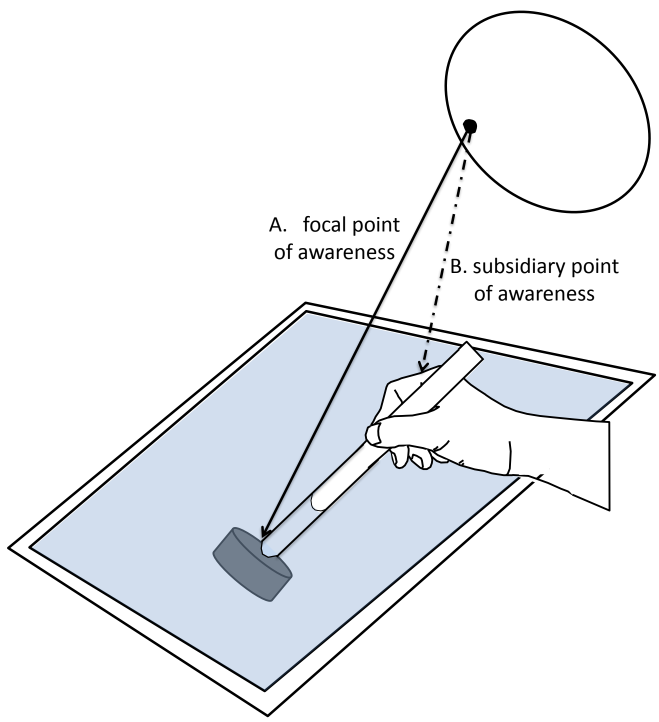

Figure 1.5: *focal point of awareness* of *ImpAct* (A) and *subsidiary point of awareness* of *ImpAct* (B) with respect to a user holding it like a pen or a drawing tool.

1.4 Scope and the Structure of Thesis

Primary focus of this thesis is to demonstrate *ImpAct* as a real implementation of a direct touch tool for surface computing. We introduce the concept of direct touch and our approach to implement it suing *ImpAct*. Design process is discussed in detail giving special attention to the adaptation of direct touch features to *ImpAct* and practical limitations and challenges.
Chapter 2 examines the related works and advancements relevant to the concept of direct touch and *ImpAct*. Chapter 3 describes the adaptation of direct touch concept into the design of *ImpAct* and challenges faced. Real implementation of *ImpAct*, its characteristics and developed applications are presented in Chapter 4. In Chapter 5, details of the technical evaluation conducted to investigate the accuracy and operability of *ImpAct* is presented. Chapter 6 gives a general discussion on the overall system with design challenges and future works. Finally, thesis is concluded in Chapter 7.

Chapter 2

Related Work

Concept of *ImpAct* combines two different fields of research into a single stream. They are surface computing and haptic interfaces. These two fields have mutually exclusive approaches to implement rich user interfaces and experiences. And also, there are efforts to combine these two fields into a single interface strategy. In this chapter, we are going to review previous work done in surface computing, haptic interfaces and approaches to combined two technologies in relation to direct touch concept. Specifically, contributions of previous research and implications which strengthen the design process of *ImpAct* is introduced. Furthermore, we will describe *ImpAct*'s contribution to the research community.

2.1 Advancement in Surface Computing

Advancement of surface computing is primarily govern by the touch sensitive screen technologies. Though it is greatly depend on surface display systems, paradigm shifts were marked by touch sensing technologies. In this section, major focus is given to the advancements of surface based interactive technique. However, prior to introducing the interactive techniques, history and development of touch sensing technologies are briefly described.

2.1.1 Introduction to Touch Sensing Technologies

First introduction of activation or manipulation using a pointing device was first introduced by Ivan Sutherland in Sketchpad system[44]. Sketchpad used a light pen to point to objects on a display screen, which eventually lead to develop the direct manipulation for human computer interaction and gave initial implications for possibility of touch interfaces. However, unlike the Sketchpad, early touch sensitive devices were not used as touch screens since they were not transparent enough to put on top of a screen.

There are many different technologies used to sense and locate a touch point on a surface. Some of the early technologies are, Resistive, Surface acoustic wave(SAW), Capacitive, Infra-red and Strain gauge based touch sensing. Optical imaging, Frustrated Total Internal Reflection(FTIR), Dispersive signal technology, Acoustic pulse recognition and Coded LCD: Bidirectional Screens are some of the new technologies for touch detection. This thesis will not go into details of these technologies since technology itself does not hold great relevance to the scope of our research. However, we will review some of the novel interaction possibilities introduced via different touch sensing technologies in detail.

Earliest development of complete touch screen based computer is PLATO IV terminal[5]. Which used Infra-red(IR) sensors to detect single touch actuations on the terminal screen. Further developments in touch sensing technologies were used in early robotics applications to detect shapes and orientations[53]. This touch sensor developed by Jack Rebman was capable of detecting multiple point of touch simultaneously, introducing multi-touch sensing. First multi-touch surface which was solely designed for human computer interaction was introduced in 1982 by Metha in his research 'A Flexible Machine Interface'[31]. Multi-Touch Screen developed by Bob Boie in 1984 was the first of its kind to combine the touch and vision together. Following year, Lee, Buxton and William introduced the Multi-touch tablet, which introduced, individual touch point detection with their respective degree of touch along with algorithms to interpolate and improve the resolution[27]. Next remarkable point in surface computing was the introduction of Digital Desk by Wellner in 1991[50]. Digital desk used front projection to project GUI components on an existing surfaces (such as a table) and used combined input from vision and acoustic sensing to calculate touch points. It was the first to introduce gestural inputs for touch surfaces which are very popular today such as two fingered pinching and scaling.

Recent development of touch sensing technologies has given incredible accuracy and higher resolution for both locating the touch point and the degree of touch. Not only touch point, but, area of the touch, shape and many other information can be extracted. Rekimoto introduced a capacitance based touch sensitive Smartskin in 2002[38]. This is a good example for high resolution sensor with minimum number of individual touch points.

Furthermore, FTIR based multi-touch surface proposed by Han is another good example of robust and low cost multi-touch sensing systems[11] and it is one of the most widely used technologies today.

2.1.2 Interactive Techniques on Surface Computing

In order to analyse the requirements for direct touch on surfaces, it is important to understand the gradual development of existing interactive techniques on surface computing and their limitations. In first few paragraphs of this section, we try to introduce the formation of basic interaction scenarios on surface computing environments and then we will present the existing approaches to extend the interaction space. Finally, we will introduce the contribution of *ImpAct* in the context of surface computing.

As briefly described earlier, initial interaction with surfaces was limited to point and activation. However it may seems simple and limited, many complex operations could be deduced from these simple input variables. Good example would be Sketchpad, which was design to detect initial and end point of users touch using a light pen and utilize them to create wide variety of meaningful operations inside the computer.

By early 80's, touch sensitive surfaces has been developed to give promising future directives towards human computer interaction techniques. In 1983, Nakatani and Rohrlich presented a philosophical framework for development of touch screen based computer interfaces, which they called *Soft Machines*. They pointed out the hard controllers such as key boards lead to make inflexibility and complexity in computer environments. They proposed this can be eliminated by using soft interfaces, which are graphical presentation of controls on a touch screen surface. Soft controls provide ways to eliminate those two disadvantages in hard-machines by adjusting the interface according to the requirement. For an example, lets say, there is a need to insert a users age into a computer program, in this case, computer can display a numeric keypad on screen, without alphabetic keys to reduce the complexity of the input method for user. This is not possible with a hard keyboard, since the layout or visible buttons can not be changed or in other words, it is not flexible enough to adjust to the function. This lays the basic foundation for development of interactive techniques based on soft interfaces and eventually lead to progress of the surface computing.

Though the VIDEOPLACE project created by Krueger et. al. is not directly related to touch screens, it introduce many positive implications towards the development of interactive techniques of surface computing[23]. Basically, VIDEOPLACE is a system where a video camera captures a user and project his/her silhouette on to a screen along with synthesized graphical objects. User can see the combined projection and can interact with projected virtual objects. This research project introduced simultaneous and free form interaction with multiple control points. For an example, user can grab and object using end points of thumb and index fingers of both hands (total of four contact points) and manipulate them continuously to make them scale or rotate. Following this research, Multi-touch tablet was introduced, which is capable of continuously tracking finger inputs on a surface along with the measurements of the degree of touch (indication of pressure of touch) to formulate the basic interactive techniques for surface computing[27]. Furthermore, Multi-touch tablet introduced GUI controllers and basic interface components to work with touch input and relevant gestures to use them. These two projects demonstrated a clear interaction scenario for multi-touch surfaces which was adapted later by Wellner in his DigitalDesk[50] and many others followed.
Evolution of surface computing we have been described so far defines the general interaction scenarios associated to touch sensitive screens. And they are widely available in most of the existing commercial systems, with incremental improvements and customized functionalities. However, in the research world, lot of limitations were identified in general surface computing systems. Two basic limitations are that 1)all the interactions are limited to 2D surface and 2)feedback is limited to graphics (no physical feedback)[49]. We believe that direct touch and manipulation techniques of *ImpAct* will eventually overcome these limitation. However, there are existing advancements in research to address both of these issues. In this section, we will analyse some of the research approaches to extend the interaction space beyond the 2D surface with touch screen computing environments.
There are many approaches to extend the interaction space of surface computers beyond the 2D plane gestures. One of the early approaches was introduction of sensing the degree of touch, or the amount of pressure excreted by users on the surface[27]. If we consider the plane of surface to

be x-y, then this technique enabled the z axis control on the surface. This approach was further developed in by Sinclair in 1997 in the Haptic lens project using degree of touch to shape and form 3D clay like structures inside the screen[41]. Applying different levels of pressure on the screen, users can change the shape of the visual representation of the virtual 3D material given inside the screen. Later, pressure based widget controls for GUI based computations were introduce by Ramos[37]. However the detectable range of pressure variation is very limited and has a poor controllability at high resolutions. Direct z axis controls has been implemented by Lapides et al. as *3D Tractus* using a moving display, however, the display can move only in one direction and no rotations are allowed[25].

Furthermore, size and shape of the touched area on the surface was taken as an independent variable for input commands[38, 43]. This can enable different interaction scenarios for surface computing such as virtual force metaphor, rotation metaphor, etc, presented by So et. al.[43]. Addition to this, possibility of using direction of finger touch point as an input is explored by Wang et. al.[48]. They presented a technique to identify the finger orientation and method to manipulate and interact with visual objects using it. Even though the interaction space is expanded by these technologies, it still remains limited to the surface plane.

Wilson et. al. proposed a technique to detect user hand movements above the display surface area to bring the planar gestures of surface computing to three dimensional gesture space[52]. They further implement physics on these interaction techniques to enable real world like object manipulations. BiDi screen is another gesture manipulation system which use a novel vision based technique to capture 3D movements of the users hand[14]. 3D interaction space enabled by these technologies provide much freedom in control, however there are no feedbacks to user hands and there are no visual contact between manipulation object and hands. This could lead to ambiguity and confusion in object selection and manipulation.

In order to improve the interaction capabilities and user experience on surface computing systems, there are many assistive technologies used. Stylus is an example for such technologies which enable precise control on touch surfaces and help to overcome human limitations such as *fat finger problem*. Suzuki et al proposed a set of enhancements to stylus by attaching an accelerometer to it and tracking user actions performed in air[45]. Fur-

thermore, Bi et al explored the possibility of using pen rolling as an input for pen based interactions[4] and Tian et al presented the concept pf *Tilt Menu* to further explore stylus based interactions[47]. These researches highly influence the design of *ImpAct* since it uses the rolling and orientation tracking for its operation. Using tilt and rolling as direct cues for interaction could be very useful, however, since normal styluses are completely external to the touch surface, input and function could have lesser correlation. However, *ImpAct* does not use orientation and rolling as sole interaction cues, rather they are used to calculate its projection inside the screen (see section 3.1). And this projection or the virtual stylus is used to generate meaningful manipulation and probing tool for interactions.
Other than, stylus, there are many assistive technologies used along side surface computing in research world. In *Bricks* project, Fitzmaurice, Ishii and Buxton presented a graspable user interface concept with related to a table top surface computing environment[7]. In *Bricks*, virtual objects inside the screen can be manipulated by physical objects placed on the screen so that the users can extend the limited two dimensional interaction space to multidimensional and graspable physical space. Furthermore, Sato et al presented Photo elastic touch, a soft touch interface for surface computers[39].

2.2 Haptic Interfaces

It is interesting to see that early adaptation of haptic feedback systems were used in tele-operation and tele-manipulation systems[3, 40]. It has been emphasized the importance to have the presence of impedance(in the sense of haptics) specially in manipulation tasks in tele-existence systems[46]. Then they emerged into the general computer human interaction field including virtual reality and gaming applications[2, 2, 19, 24, 26, 30, 33].
Early adaptation of a tactile display for human computer interaction is the sand paper system developed by Minsky and her collogues[33]. System was designed to give cutaneous sensation to the users hands according to the different textures displayed on a screen. *SmartTouch* is another implementation of wearable haptic device to give tactile sensation to a human finger according to the visual information on the place finger touches using mechanical means[19]. Functional electrical simulation[18] and focused acous-

tic waves[17] can also be used to create tactile sensation. Tactile display systems are capable of providing texture data to user. Usage is straightforward in a surface with a wearable device. However, our approach is different since manipulation tasks depends more on kinesthetic information rather than tactile information.

Massie proposed the PHANTOM, a point force-feedback display system solely designed for human computer interaction purposes[30]. *ImpAct* is highly influenced by the learnings from PHANTOM as much as all the other preceding haptic interfaces does. In this project, Massie proposed three guidelines for designing force feedback haptic displays. They are 1) Free space must feel free, 2) Solid virtual objects must feel stiff and 3) Virtual constraint must not be easily saturated. *ImpAct* is design to adhere to these guidelines as much feasible. Main difference between PHANTOM and *ImpAct* is that *ImpAct* follows a direct touch approach while PHANTOM was originally designed for indirect touch. However, recent developments can help PHANTOM to be used for direct touch with head mounted display systems.

Since *ImpAct* is a haptic enabled stylus for touch screens, it is better to examine few existing haptic styluses. Haptic pen is a successful haptic stylus implementations with tactile sensation[26]. It gives different tactile sensations according to different GUI events such as mouse down, mouse up, click, etc. using small displacements of a solenoidal motor attached to it. wUbi-Pen is another tactile stylus which can give the sensation of impact (colliding of two objects) to the users fingers[24]. However, both represent 2D surface details as cutaneous sensations and does not have any means to enable direct touch. *Pen de touch* is much more advanced haptic display, which can give partial kinesthetic sensation to users fingers. However, it is meant to use above the display surface and does not provide direct touch features.

2.3 Contribution of *ImpAct*

As described in the introduction, main contribution of *ImpAct* is, it enables direct touch. In our reviewing of novel interaction scenarios introduced to surface computing, we could not identify a technique which can enable direct manipulation of displayed objects. Most of the proposed techniques

still hold the boundary between real world to digital world on the surface of touch and others bring the interaction above the screen disrupting the visual continuity of touch and display system. Furthermore, most of the haptic display systems reviewed are design to use indirect touch. Direct touch is only enabled using head mounted displays or virtual reality mechanisms.
Therefore, contribution of the *ImpAct* to the HCI community is novel and it enables new interaction possibilities between human and machines. Specially, in the context of this thesis, direct touch and manipulation techniques proposed for surface computing using *ImpAct* can be considered as non-trivial.

Chapter 3

Design of *ImpAct*

In this chapter, we will discuss the design of *ImpAct* as a direct touch tool. Design is classified into four different categories. First is Simulated Projection Rendering. In this section, we will describe the measurements required to generate the simulated projection rendering and methods used to construct the render projection. Second category is design of haptic feedback system. In this section, we will discuss the methods used to generate haptics feedback to user's hand and the principle of combining visual and haptic information into the same spatial coordinates. Third and forth categories are allocated to describe hardware and software models used to implement the actual system.

3.1 Simulated Projection Rendering

Simulated Projection Rendering(SPR) is one of the core concepts that drives the direct touch features of *ImpAct*. As described in the introduction part, *ImpAct* consists of a movable ram inside its outer grip (works like a solenoid) and this ram movement is used to change the overall length of the stylus. When user pushes *ImpAct* against the screen, physical stylus will shrink while a rendered projection will be drawn inside the screen continuously mapping angular and length changes of *ImpAct* to that of the projection (shown in Figure 3.1). This process is called simulated projection rendering.
As shown in the Figure 3.1, in the process of simulated projection rendering, all the orientation changes and length changes of the physical stylus is matched to the rendered stylus so that it visually aligns with the physical stylus. In other words, projection is simulated according to the dynamics of the physical stylus, thus the name simulated projection rendering.

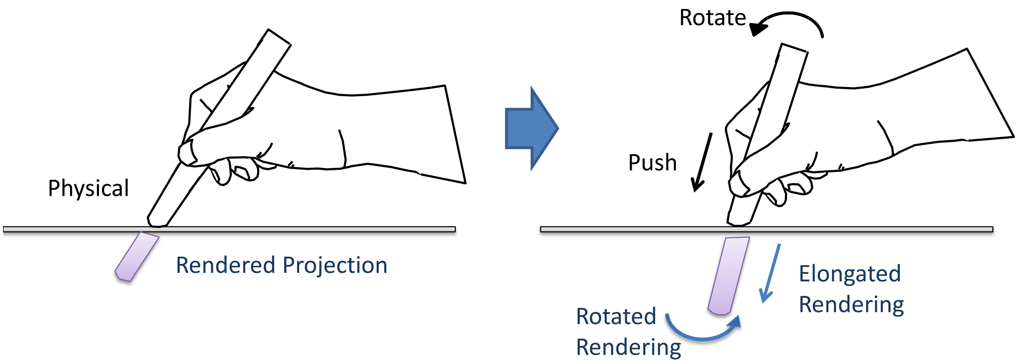

Figure 3.1: Operating principle of the simulated projection rendering. Left: Virtual stylus is rendered along the physical stylus to make a visually continuous interface. Right: When user changes the orientation or length of the stylus, rendered stylus is matched and redrawn to project changes in orientation and length.

3.1.1 Measurements for Projection Rendering

There are few important measurements required to implement the simulated projection rendering. These measurements are visually represented in the Figure 3.2. First measurement required is the length of the rendered stylus to be drawn (Figure 3.2 (A)). Required length of the virtual stylus to be drawn is equal to the difference between the original and current lengths of the physical stylus. Electromechanical displacement measurement mechanism is used inside the *ImpAct* to measure these length changes.
Second measurement required is the touch point of *ImpAct* on the surface. This is the point of insertion of virtual ram into the digital world on the screen (Figure 3.2 (B)). Location of the touch point is acquired from the touch sensitive surface measurements. This is a primary feature of touch surfaces, however, we took special attention in implementation of *ImpAct* to make it compatible with different touch sensing technologies. Acquired touch position is transferred to the user touch point in the 3D virtual world from the perspective of the user.

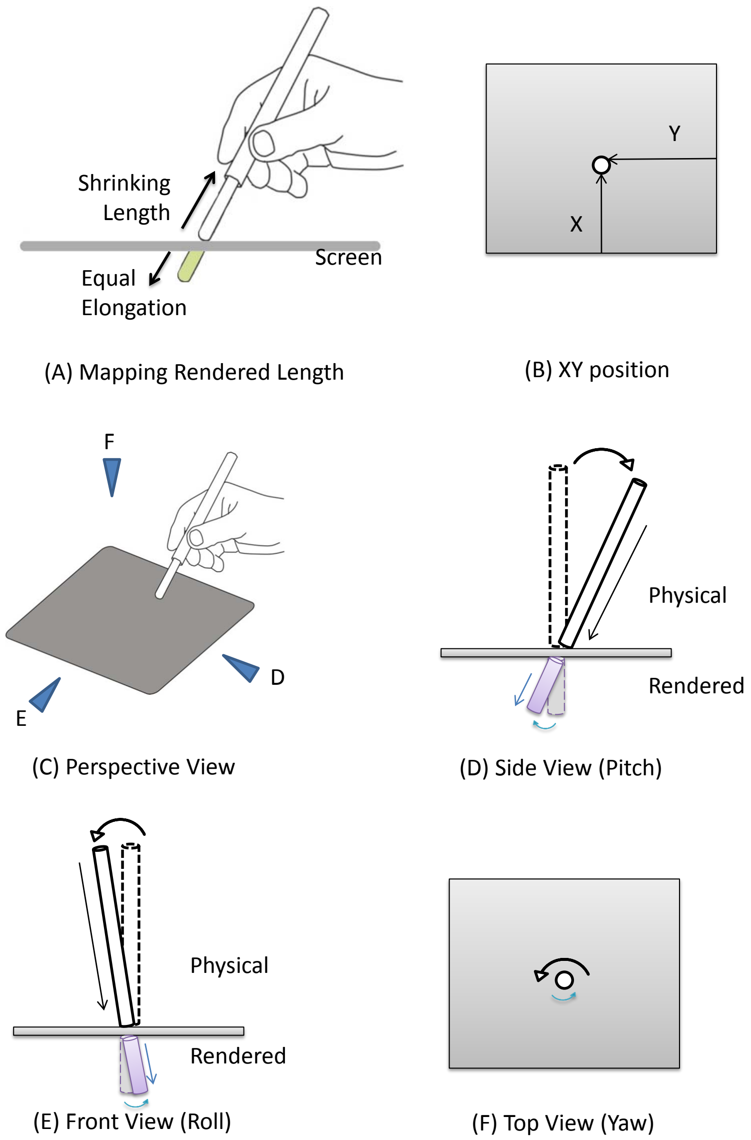

Figure 3.2: Measurements required to implement the Simulated Projection Rendering. (A) Matching the length change of the physical stylus to rendered stylus, (B) Pointing the stylus on the touch surface, (C) Perspective view of *ImpAct* interacting on a surface, (D) Matching the pitch of the physical stylus to rendered stylus, (E) Matching the roll of the physical stylus to rendered stylus, (F) Matching the yaw of the physical stylus to rendered stylus.

Third measurement is the orientation of *ImpAct*. Orientation is measured as the pitch, roll and yaw values of *ImpAct* as shown in (Figure 3.2 (D), (E) and (F) respectively. There are many possible ways to measure the orientation of an object such as accelerometer/compass based tracking, motion capture, computer vision, etc. In order to enable *ImpAct* to be used as a mobile device, we chose embedded accelerometer/compass sensing module to measure the orientation.
Finally, altogether, there are 6 independent variables measured from the input parameters to generate the projection inside the screen and possible activities. They can be listed as bellow.

1. Length change of the *ImpAct* as z value.
2. Location of the x value of *ImpAct* on screen derived from the touch point
3. Location of the y value of *ImpAct* on screen derived from the touch point
4. *Yaw* angle (ψ)
5. *Pitch* angle (θ)
6. *Roll* angle (φ)

3.1.2 Rendering Projection

In this section we describe the method used to render the virtual stylus below the display using the 6 measurements acquired from user interaction. Virtual stylus is a model which is similar to combination of a cylinder and a sphere, which shares the same diameter. Sphere is placed at the end of the cylinder such that the centre of the sphere coincide with the centre of the circular face of the cylinder at the end. Structure of the virtual stylus and a rendered model is shown in the Figure 3.3.
Once a touch is registered on the surface, system acquires the touch point and move the origin of the virtual stylus, which is located at the centre of the flat circular face of the cylindrical part, to the touch point. And set the height of the cylindrical part of the virtual stylus to be equal to that of the length change occur in *ImpAct*. Spherical part of the stem is drawn inside

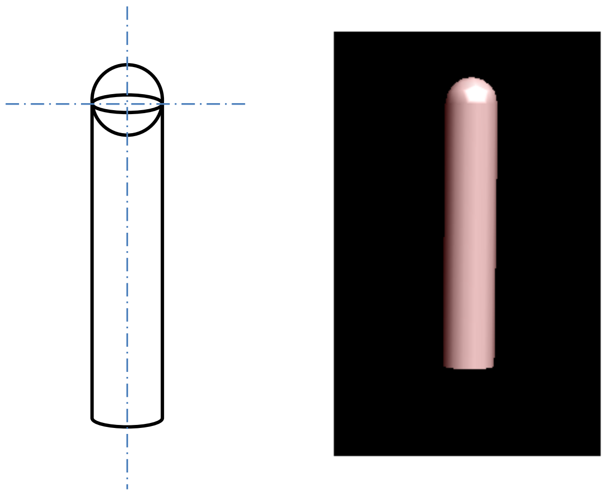

Figure 3.3: Structure of the virtual stylus and a rendered model.

the screen when a touch is registered, irrespective of the length change in *ImpAct*. However, this does not make any visual effect on the user, because when the length is supposed to be zero, height of the cylindrical part is equals to zero making its a half sphere which is displayed inside the screen and *ImpAct*'s physical stem occludes this half sphere. Figure 3.4 shows two cases where *ImpAct* touching the screen but not pushed in (A) and *ImpAct* is pushed length x into the screen (B). Further, if there is no touch is registered, then there will be no virtual stylus rendered inside the screen.

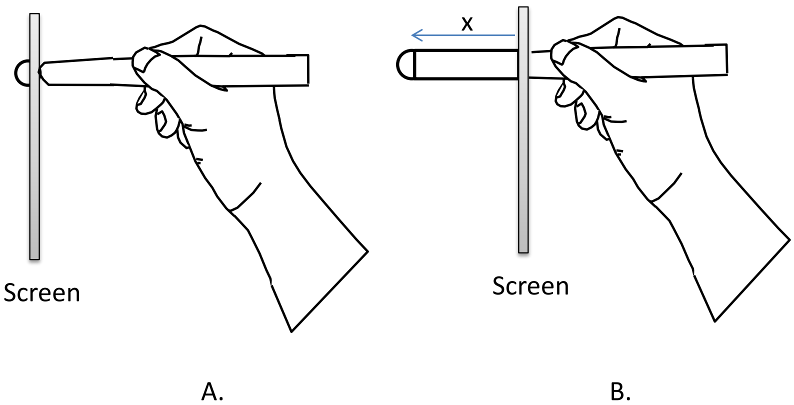

Figure 3.4: Rendering Virtual Stylus Length.

Once the virtual stylus is rendered inside the screen with desired position of touch and length, it should be oriented inside the screen to match the orientation of *ImpAct*. This is done turning the virtual stylus model about z, y and x axis's respectively with yaw, pitch and roll values measured from *ImpAct* orientation.

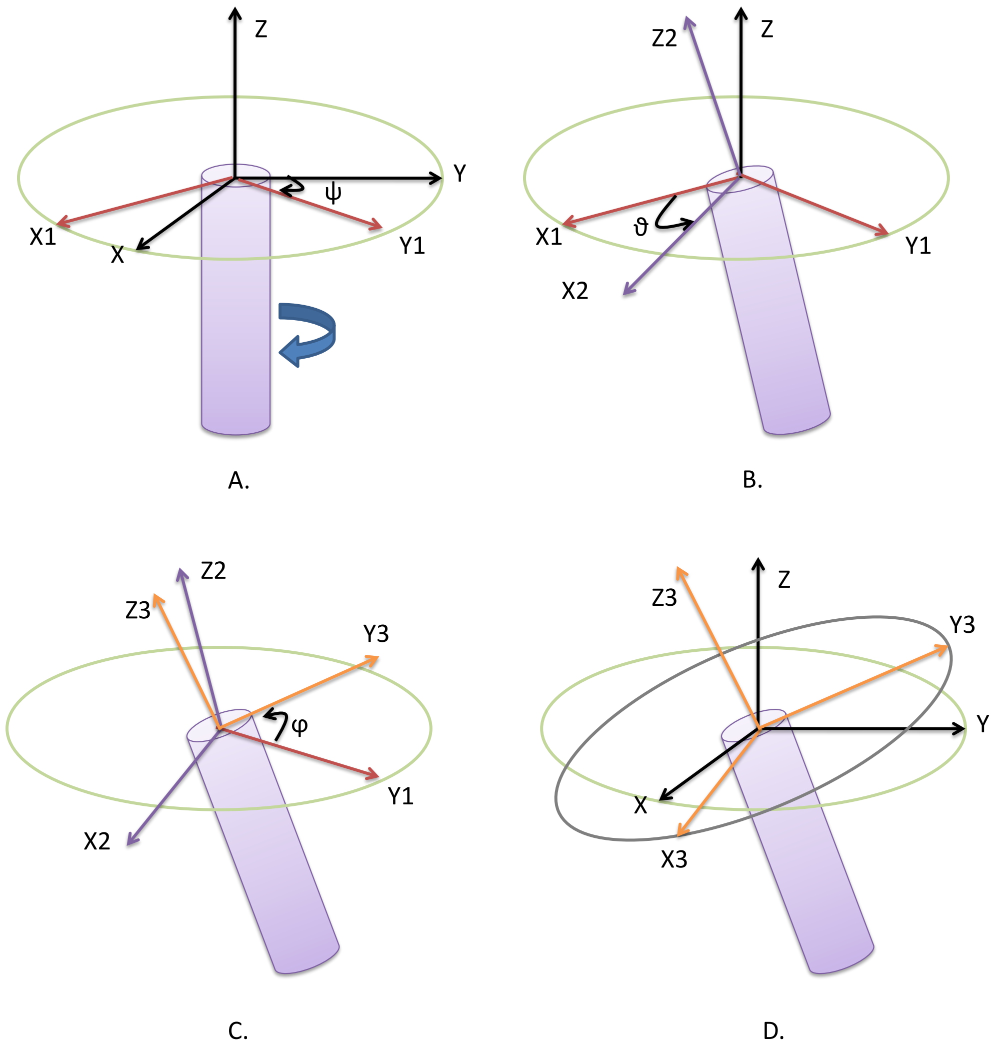

Figure 3.5: Rendering Virtual Stylus Orientation.

Figure 3.5 shows the process of orienting the virtual stylus, the circular shape marks the plane of the screen surface. Figure 3.5-A shows the application of yaw angle ψ to rotate the virtual stylus about Z axis making local x and y axis of virtual stylus to be moved to $X1$ and $Y1$ directions respectively. In Figure 3.5-B, virtual stylus is rotated the pitch angle θ about the moved y axis $Y1$, making the x and z axis of the virtual stylus to be change to the $X2$ and $Z2$ directions. Finally, roll angle φ is applied about the changed $X2$ axis of the virtual stylus to change y and z axis of the virtual stylus to be moved towards $Y3$ and $Z3$ directions as shown in Figure 3.5-C. Figure 3.5-D shows the initial X, Y, and Z directions and final $X3$, $Y3$ and $Z3$ after the orientation is completed.

3.2 Haptic Feedback

This section describes the design of haptic force feedback mechanism for *ImpAct*. It includes design of the physical model to exert a force on user's hand, grounding mechanism and affecting parameters for force exertion. Furthermore, section discuses the forces that can be simulated by *ImpAct* and forces it can not simulate. Addition to the *ImpAct* hardware model, section discuss the software model used to calculate the force exerted by *ImpAct* to enable interactions with virtual objects inside the screen.

3.2.1 Force-feedback in *ImpAct*

Force-feedback display systems require a grounding mechanism to create the force. Many existing force-feedback systems are grounded[16, 29]. All the forces simulated are created with respect to the ground and it makes a final sink to all the forces. *ImpAct* is designed as a mobile haptic device. We can not afford to have a hard link to any grounding entity. However some sort of grounding is compulsory to exert a force-feedback on user's hand. Therefore, we use the display surface as the grounding for *ImpAct*. Figure 3.6 demonstrates the mechanism used to exert force on user hands by *ImpAct*. As described in Section 1.3, moving shaft of the *ImpAct* is attached to a DC motor via a rack-pinion type gear mechanism. This motor can exert a torque τ on the moving shaft, and moving shaft conveys the force Q along the axis of shaft to the display surface. This force creates two reactive forces on the touch point of *ImpAct* and surface, normal force (N) and friction force (F). According to Newtons laws, two forces, N and F should create a resultant force R which is equal to the initial force Q in magnitude and opposite in the direction (as shown in Figure 3.6) to keep the equilibrium. This equation is valid true as far as the *ImpAct* touch point does not move relative to the screen surface (i.e. no slip). Since the resultant force R is created backward along the *ImpAct* axis, user feels this as a force-feedback.

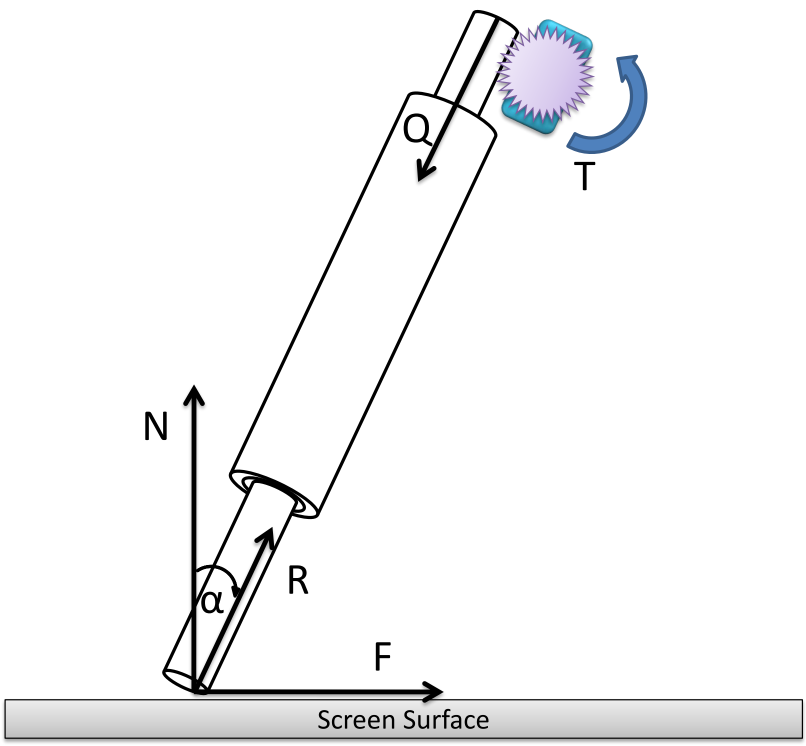

Figure 3.6: Force-feedback using screen surface as grounding

Surface base grounding system provides the ability to make *ImpAct* to be a mobile device. However, this comes with few limiting factors to its operation.

Since the force exerted on the user hand depends on the equilibrium of the touch point, friction between the *ImpAct* and the surface is an important factor for design of force-feedback system. Friction force has a limiting value, which is governed by the *coefficient of friction* (μ) between display surface and *ImpAct*. Relationship between friction force (F), force exerted by *ImpAct* on the surface (Q) and incident angle of *ImpAct* to the surface (α) is related as given in Equation 3.1.

$$\begin{aligned} F &= \mu N \\ N &= R\cos(\alpha) && \text{by force division} \\ N &= Q\cos(\alpha) && \text{Since R=Q} \\ F &= \mu Q\cos(\alpha) \end{aligned} \tag{3.1}$$

Since *cosine* function reaches 0 when subjected angle reaches 90^o, it is evident that *ImpAct* has lover strength in giving consistent force to user when it is at higher incident angles. This is a design limitation we accepted as unavoidable in *ImpAct* and software simulation process for haptic generation takes this fact into account.

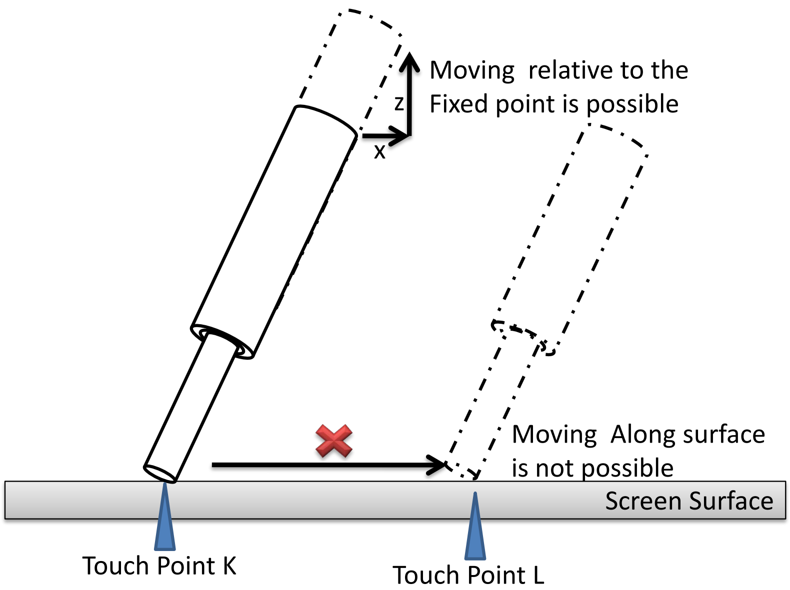

Figure 3.7: Limitation for moving across display plane

Another limitation of the *ImpAct* is that it can not create a dynamic movement across the XY plane, or the display surface plane. By applying force to the moving shaft of *ImpAct* will allow it to move relative to a fix point on the surface, however, it can not move the touch point on the screen, unless user moves it across. This is shown in Figure 3.7, where, *ImpAct* can forcefully move x and y distances relative the point K on the display surface while it can not move to another point L on the display surface without user intervention.

3.2.2 Haptic Model of *ImpAct*

Given the physical design of *ImpAct*, it is important to create a haptic model which can simulate plausible haptic cues for a user's hand utilizing its capabilities. As decried earlier in section 3.2.1, its capabilities are limited to the forces that can be transmuted to users hand along the direction of its actuation axis(i.e. axis of the cylindrical shaft). In other words, if a prospective haptic cue does not contain a force component in the direction of the central axis of *ImpAct*, then that force can not be simulate.

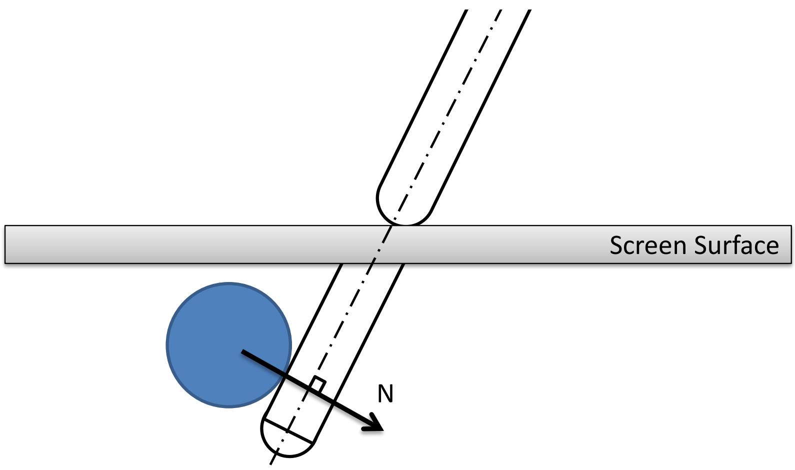

Figure 3.8: Perpendicular forces on *ImpAct* stem.

Figure 3.8 shows an example of such a case. Force exerted on *ImpAct* by the object, N, is perpendicular to the force actuation direction of *ImpAct*. Only possible haptic detail *ImpAct* can produce about this object is the friction between *ImpAct* stem and the object. However, it could involve complex analysis to calculate these friction components and torque components. Which is beyond the scope of this thesis. Haptic model of *ImpAct* described in this document is governed by following basic rules.

1. Only the forces with non-zero component directed along the axis of actuation of *ImpAct* is simulated.

2. Friction components and torque components are neglected.

3. Forces are simulated only if they interfere with the tip of *ImpAct* virtual stylus.

As we described earlier, rule 1 is derived because *ImpAct* is not capable of interpreting forces which are perpendicular to *ImpAct* actuation axis. Second rule discards the friction components to eliminate complex calculations required and also because the magnitude of friction components could be negligible compared to force component. Also, previous researches has shown that many meaningful haptic interactions involves little or no torque[30].

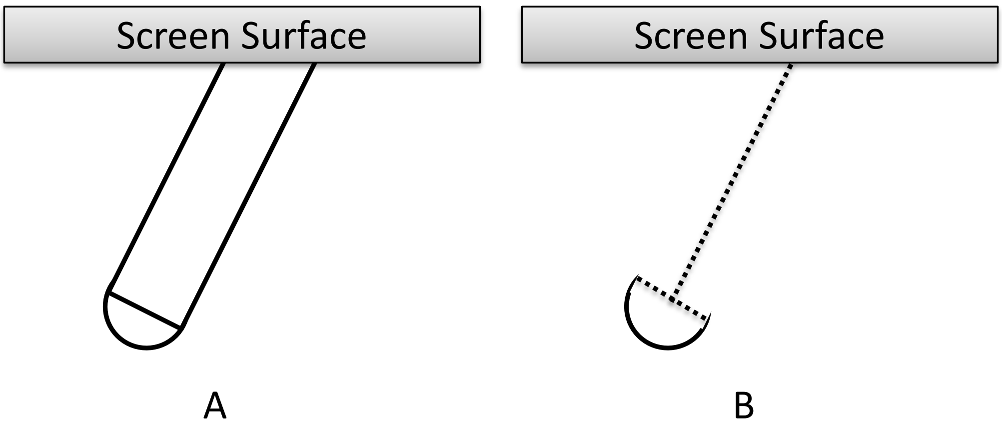

Figure 3.9: Haptic and visual model of *ImpAct*. A: Visual Model, B:Haptic Model

Furthermore, object interferences on the cylindrical component of the virtual stylus is neglected and only tip is considered as the haptic sensitive area. This differentiates the visual model and haptic model of *ImpAct* virtual stylus. This is shown in the Figure 3.9, where (A) shows the visual model which is rendered inside the screen and (B) shows the model used to calculate forces from haptics interactions. Dotted lines are not considered as the surfaces where a force can be act upon. However, this does not interrupt the direct touch of objects but limits the touch point to the tip of the *ImpAct*.

3.2.3 Calculation of Forces

This section describes the method used to calculate forces to create haptic interactions with *ImpAct*. As we describe earlier, only the forces exerted on the tip of *ImpAct* is considered for making haptic feedback to the user. Three different kinds of force exerting surfaces are analysed to create haptic stimulations. Other complex shapes are not implemented in the current design. Figure 3.10 shows the three shapes considered to implement in the current prototype. They are (A) force exerted by a spherical object, (B) Force generated by a plane surface, (C) Force generated by an edge.

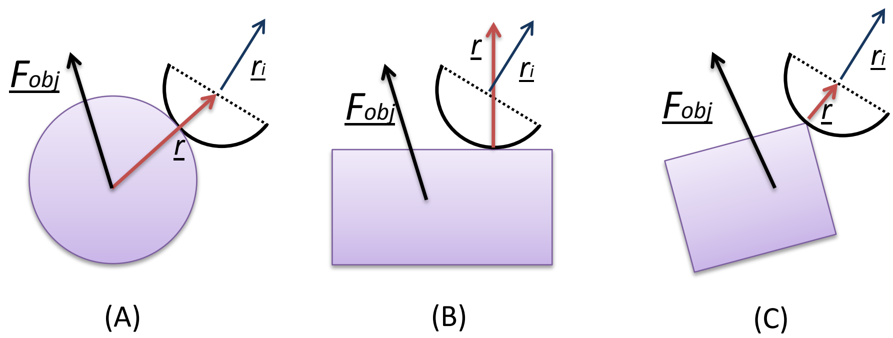

Figure 3.10: Calculating forces for haptic interaction. A: Force exerted by a spherical object, B: Force generated by a plane surface, C: Force generated by an edge.

Calculation of forces are done using vector mathematics. Every object implemented inside the virtual world contains a vector indicating the force excreted by it $\vec{F_{obj}}$. In case of a solid object, it will assume a normal force according to the place of collision or contact. Calculation of the haptic force is started by the directional vector $\vec{r}$ according to the surface in contact. In case of a spherical object, $\vec{r}$ is the vector from the centre point of the sphere to the centre or origin of the half sphere of the virtual stylus. In case of a planar surface, $\vec{r}$ is the normal vector to the surface. In case of an edge, $\vec{r}$ is equal to the vector drawn from point of contact between the colliding edge and stylus surface to the origin of the half sphere of the virtual stylus. Once the $\vec{r}$ is calculated, it will be normalized to form the $\vec{r_0}$, which is a pure indication of the activation direction of the force on *ImpAct*. If the force active on the *ImpAct* is $\vec{R}$, then,

$$|\vec{R}| = \vec{F_{obj}} \cdot \vec{r_0}$$

Since the direction of the force is $\vec{r_0}$,

$$\begin{aligned} \vec{R} &= |\vec{R}|\vec{r_0} \\ \vec{R} &= (\vec{F_{obj}} \cdot \vec{r_0})\vec{r_0} \end{aligned} \quad (3.2)$$

Similarly, since the force should be exerted in the direction of *ImpAct* axis, which is indicated by unit vector $\vec{r_i}$ (Figure 3.10), if the resultant force excreted by the *ImpAct* ram is $\vec{P}$,

$$\begin{aligned} |\vec{P}| &= \vec{R} \cdot \vec{r_i} \\ \vec{P} &= |\vec{R}|\vec{r_i} \\ \vec{P} &= \{(\vec{F_{obj}} \cdot \vec{r_0})\vec{r_0} \cdot \vec{r_i}\}\vec{r_i} \quad \text{eq. 3.2} \end{aligned} \quad (3.3)$$

Vector $\vec{P}$ can be used to send the commands to the DC motor in *ImpAct* to exert the relevant force on the moving shaft, so that user feels the haptic sensation. Same method is used to calculate responses for elastic impacts and calculate final velocities for each object after collision, however, other than the forces, velocity vectors are used.

Chapter 4

Implementation of *ImpAct*

In this chapter, we will describe the implementation of *ImpAct* haptic device and related software developments. Chapter is divided into two sections, first is for hardware implementation of *ImpAct* and second is to describe the software implemented to test and demonstrate the capabilities of *ImpAct*.

4.1 Hardware Implementation

This section describes the implementation of *ImpAct* prototype and design parameters considered with reasons for selection. Furthermore, section discusses how does the design decisions influenced and projected on its operation.

4.1.1 Designing the Outer Frame and Hardware Model

Outer model of *ImpAct* is made using 3D printing technology and acrylic sheet based block framework. 3D printing was used to make the grip and inner shaft of *ImpAct* where a complex and precise model was required. Precisely cut acrylic sheets were used to build the back box, since it will require lot of alterations with results from consecutive prototyping and testing. Acrylic based prototyping is cost effective compared to 3D modelling, therefore grip and moving shaft parts were built with the scalability to used independent of the back box configurations. Figure 4.1 shows the design model and assembled prototype of the *ImpAct*.
Initial design problem was to determine the form factor for *ImpAct*. Since it is a special stylus, our goal was to integrate the design so that it could reach a comparable dimensions to a commercial stylus. However, given the functionality required and due to the limitations in prototyping facilities,

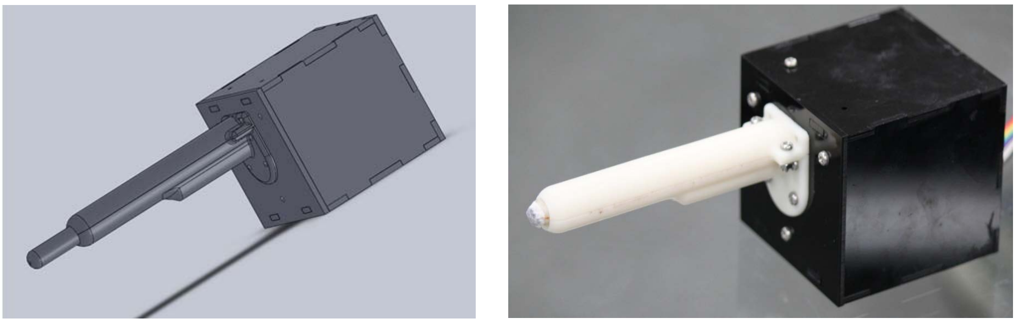

Figure 4.1: Model and Actual Prototype of *ImpAct*

it is a fairly impossible task in a lab environment. Since stylus is developed from the pen metaphor, we investigate the sizes of pens available. Average length of a pen is about 10cm and the diameter is less than 1cm. In the process, we also examined other pen types such as white board markers , highlighters and colouring pens.While length remains the same average at 10cm, these markers share a great variety of diameters ranging from 0.8cm to 2.5cm. With these observations, we finalized the length of the outer grip to be 10cm and the diameter to be 2cm, which is a form factor shared by many marker pens. Specially, since *ImpAct* has a co-centric shaft moving inside the grip, we had to keep the diameter at a higher value.

Another important decision is to select the span length for the inner shaft. At first, we thought it is better to implement the highest spanning length plausible. In order to determine a good spanning length, I tried to manipulate few thin cylindrical shafts with difference heights, which I was able to find in the lab. In this process, I noticed, with the increasing spanning length, manipulation of *ImpAct* would become hard and tiring. With higher spanning lengths, hand position goes to well above the screen. When someone is using a pen, usually they rest there hand on the table, however, if we create *ImpAct* with higher span length, user's hand will lift higher from the screen surface resulting no resting position, leading tiring and difficult manipulation. Lifting of hand is shown in the Figure 4.2.

Another limiting factor for span length of the inner shaft is that, when it is not spanned out, there should be space in the back side of *ImpAct* to fold it in. This also became a limiting factor for the span size. Considering all these factors, we decided to have a 5cm span length for the inner shaft of *ImpAct*. Though it seems small, 5cm covers fair percentage of the human

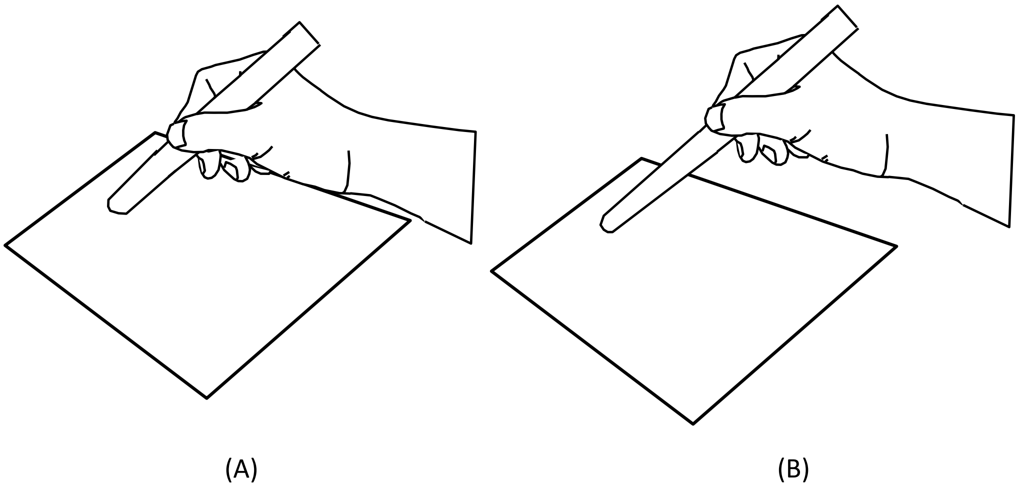

Figure 4.2: Effects of high span length. (A) at low span lengths, user can rest hand on the surface. (B) but in high span lengths, user's hand is lifted high and no resting provided.

wrist span along a linear axis.

Given the span length of the inner shaft, next important parameter is to determine the diameter. Obviously the inner shaft diameter should be less than 2cm to fit in to the outer grip and it should be thick enough to bear the tensions created during haptic interactions. Since inner shaft diameter is matched to that of the virtual stylus it defines size of the probe which is used for haptic interactions. This was explained in the [30] too. In other words, tip size of the virtual stylus is a very important factor in designing the amount of haptic details to be simulated using *ImpAct*. This is illustrated in the Figure 4.3. It can be seen that lines interconnecting the centre of the probe in (A) produce fine details compared to the same in the part (B). In general, one might think that smaller the diameter better the performance of the haptic display. However, in reality, haptics can not be presented just by probing, there has to be means to physically stimulate user. In early stage of the development, we were not certain about the haptic resolution of *ImpAct*. Therefore, we decided it is fair to chose the inner shaft diameter to be 1cm so that as far as *ImpAct* will be able to perform haptic interactions with resolution of 0.5cm, it would be able to simulate all the details picked up by the probe.

Since lot of electronics needed to be integrated, we had to attach a back

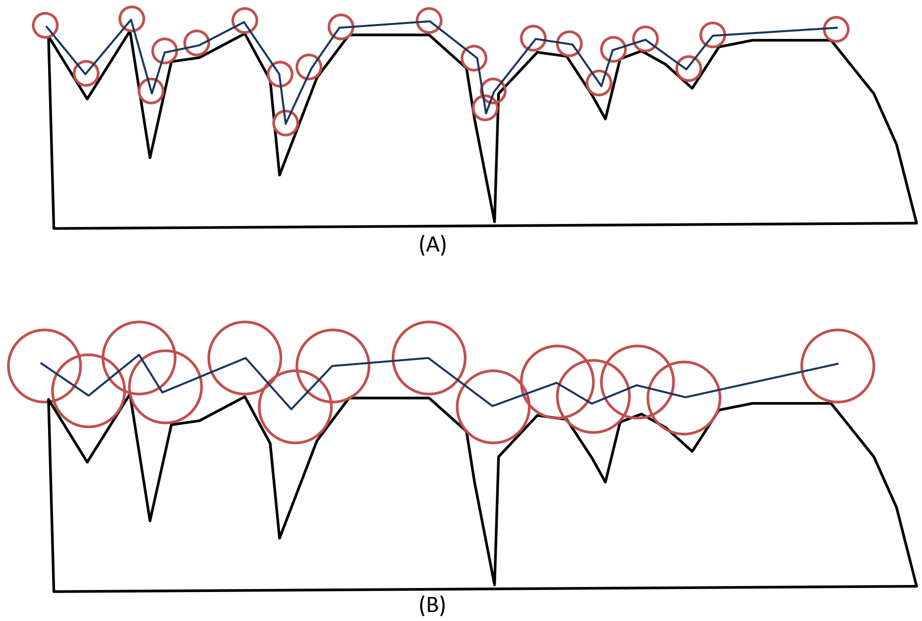

Figure 4.3: Relationship between probe diameter and haptic details. Same surface is probed using two probes with different diameters. (A) low diameter probe pick up fine details and (B) higher diameter filters fine details.

box to *ImpAct* in order to fit them in. Dimension of this back box is much higher than the diameter of the outer grip. Therefore, two alternative methods were considered as shown in Figure 4.4.

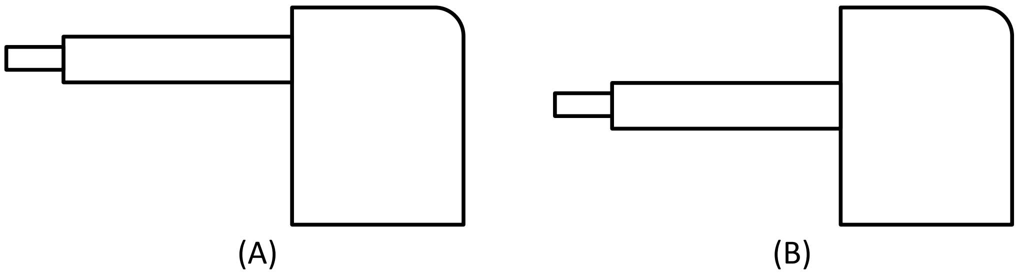

Figure 4.4: Alternative methods to attach back box to *ImpAct*. (A) Closer to the upper edge, (B) to the centre of the back box

Though it looks the center attachment to be a more balanced design, in practical terms, I thought if the outer grip is attached near the edge of the box, user can rest the weight of the back box on the back side of their palm. After implementation, this feature became very handy as shown in the Figure 4.5.

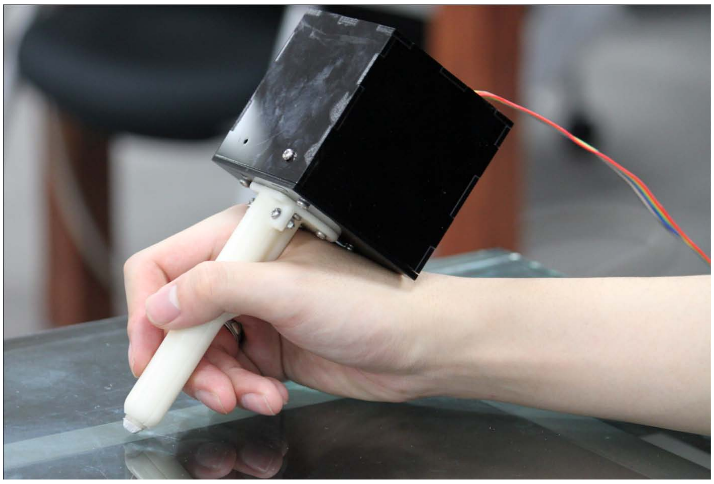

Figure 4.5: User resting the weight of back box on the back of his palm.

Furthermore, we wanted *ImpAct* to be able to work with multiple touch sensing technologies. *ImpAct* can be used in resistive touch surfaces without any modifications. However, capacitive touch surfaces are reluctant to register the contact of *ImpAct* as a valid touch since it can not stimulate capacitive effects. Therefore, we attached a thin layer of conductive form to the tip of *ImpAct* and grounded it to internal circuitry to enable capacitive touch.

Figure 4.6 shows working prototype of *ImpAct* in a user's hand.

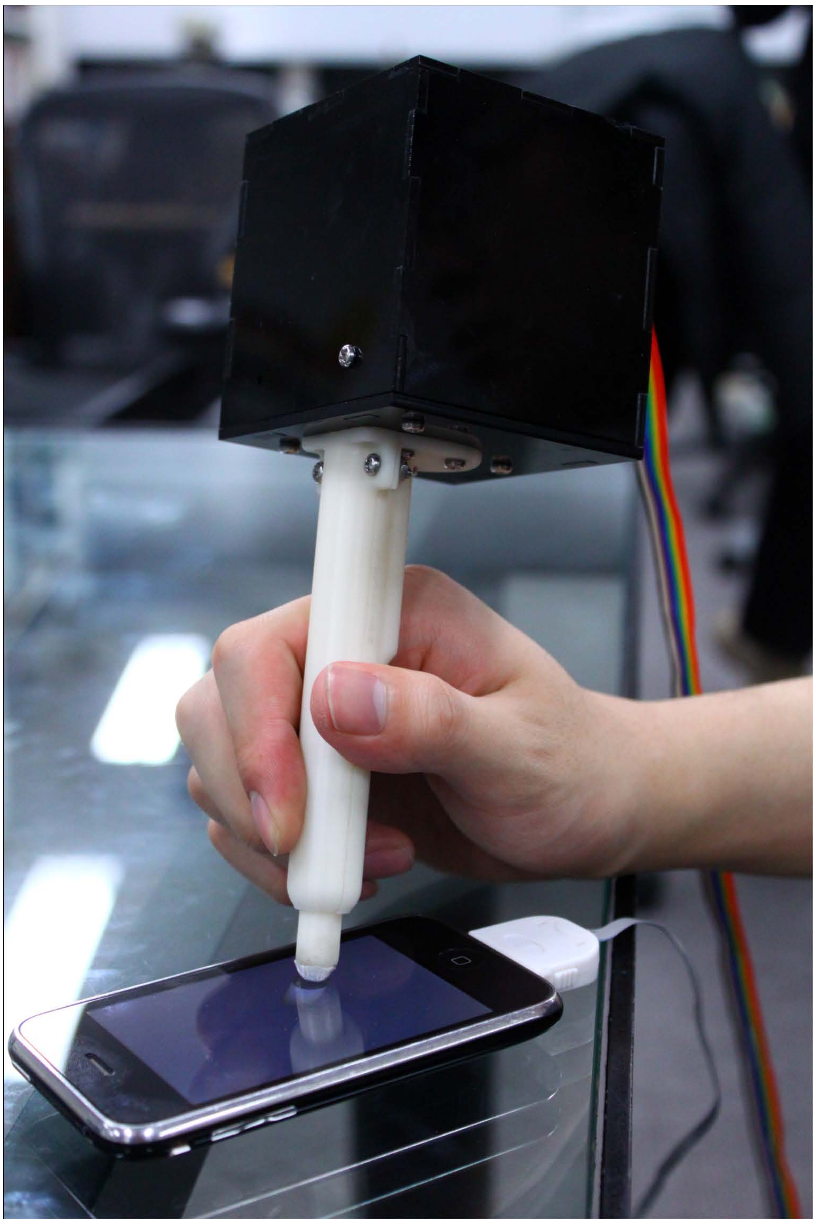

Figure 4.6: *ImpAct* at operation.

4.1.2 Electronics

ImpAct has a very limited space to setup all the hardware and electronic parts. Therefore, electronics built into the device is limited to vital functional components. *ImpAct* has both sensing and actuation built into it. Individual functions of *ImpAct* can be listed as below,

1. Measuring the length change of *ImpAct*.

2. Measuring the orientation (Yaw, Pitch and Roll).

3. Measuring the force exerted by *ImpAct* on user.

4. Driving DC motor to control the force exerted.

5. Communication with surface computing system.

These are the five basic functions implemented in *ImpAct* using electronics. *ImpAct* has a collection of electronic sensors and actuator along with an embedded microprocessor to control their functions. Figure 4.7 demonstrates individual functional blocks of internal electronics and their controlling authority of micro processor.

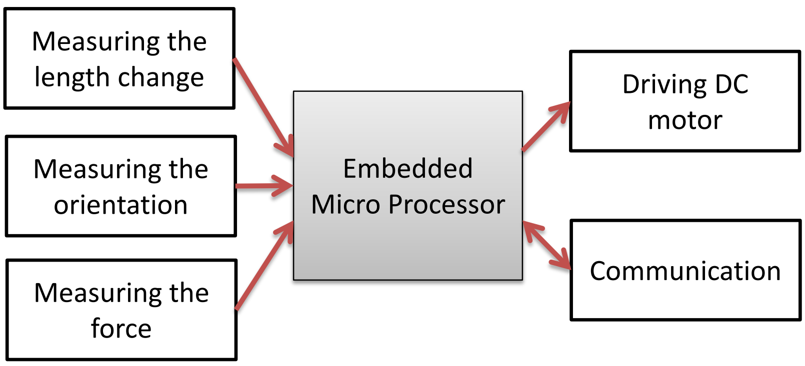

Figure 4.7: Functional block diagrams of internal electronics

Each of the individual boxes shown in the functional bloc diagram consist of combination of electronic sensors, actuators, transducers, coders and processor. Some of the inbuilt electronics transducers are accelerometer and magnetometer module, DC motor and motor driver, linear encoder and current sensors. Figure 4.8 shows the internal structure and layout of some of the visible components inside *ImpAct*.

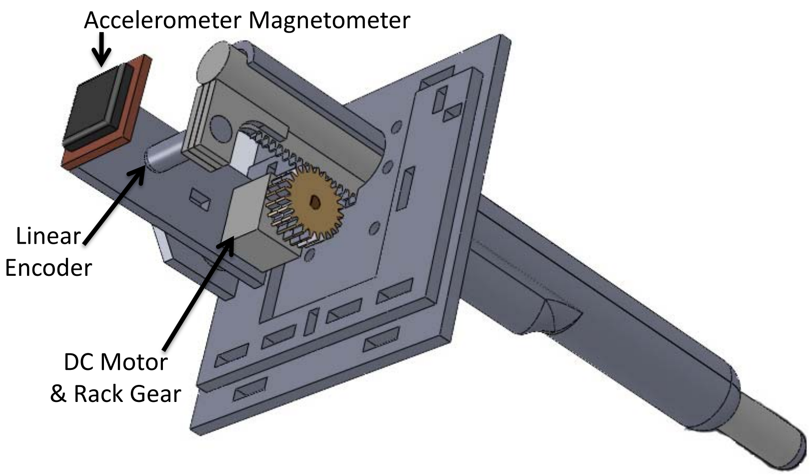

Figure 4.8: Structure of the electronics components in *ImpAct*

Span length of *ImpAct* is measured using a linear potentiometer with a pressure sensitive actuation. A wiper actuator is attached to the moving shaft of *ImpAct*, which is used to activate the linear potentiometer. Potentiometer has an active length of 5cm with $10kOhm$ resistance. Change of this resistance is measure using voltage divider bridge and analog to digital (A/D) conversion. Quantization rate of A/D conversion is $10kHz$ and digitization is done using 8bits. Digitized value indicates the current position of the moving shaft.

Orientation is measured using a combined accelerometer and magnetometer sensing device (mounting is shown in Figure 4.8). We use Honeywell HMC6343 type sensor for orientation sensing, which gives 10Hz update rate at a 0.1^o resolution of angular measurements in 10bits long data words for each angle.

Actuation force of the shaft is generated using the torque τ generated by the DC motor. And thes torque is directly proportional to the current flow in the motor. Therefore, measurement of current flow can be taken as an indication to the force exerted. We use Honeywell's CSLW Series miniature, open-loop current sensor to measure the current flow into motor. Frequency of measurement is $10kHz$ at a 8bit resolution.

Driving DC motor is controlled by the embedded processor using Pulse Width Modulation(PWM) based DC motor driver, *intersil* HIP4020. It is a Half Amp Full Bridge power driver and it is driven at a PWM frequency

of 5kHz.
One of the most important component of *ImpAct* is the DC motor used. Motor used is HS-GM21 SD, small form factor motor with max loading torque 300gcm with gears. Its average current rating is 65mA and loading current is 200mA. Figure 4.9 shows the actuation mechanism of moving shaft using the DC motor. Embeded processor uses a damping mechanism to power the moving shaft in order to prevent fluctuations and instabilities.

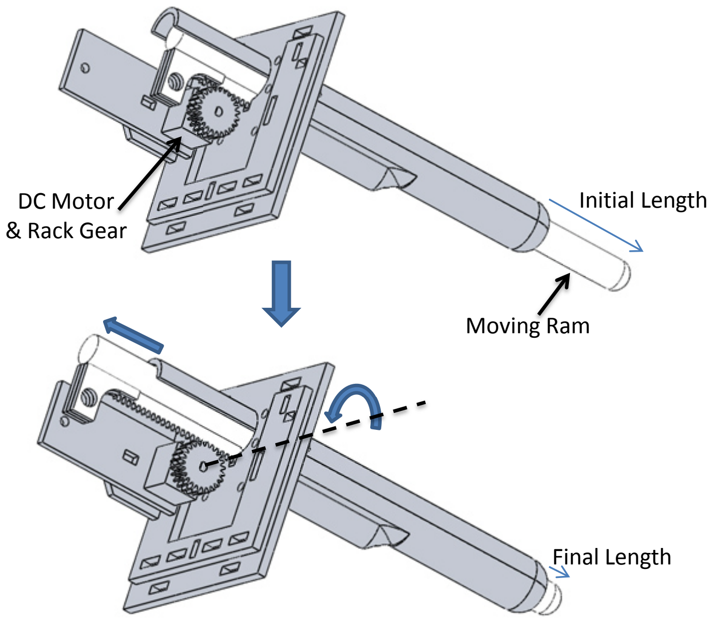

Figure 4.9: Actuation of moving shaft using DC motor

Communication with the surface computing platform is done via a RS232 serial communication protocol. *ImpAct* uses the baud rate 38400 to communicate with the computer. In order to keep the firmware operation as simple as possible, micro processor reads the raw data from sensors and directly send them to the computing platform. Data sent from *ImpAct* are yaw, pitch, and roll angular measurements, potentiometer meter measurements and the current measurement as a string of readable data. Information update rate can be controlled by the computing platform.
In the current prototype *ImpAct* uses an external power supply due to the space limitations. However, it can be powered by an integrated battery because of its low power consumption (max 250mA, average 60mA (active), 5mA(idle), 5V).

4.2 Physical Specifications of ***ImpAct***

This section states different physical parameters of *ImpAct* such as dimensions, weight, force feedback parameters, etc.
Physical dimensions of *ImpAct* is shown in the Figure 4.10. Image shows an instance where inner shaft of *ImpAct* is spanned out to its maximum range.

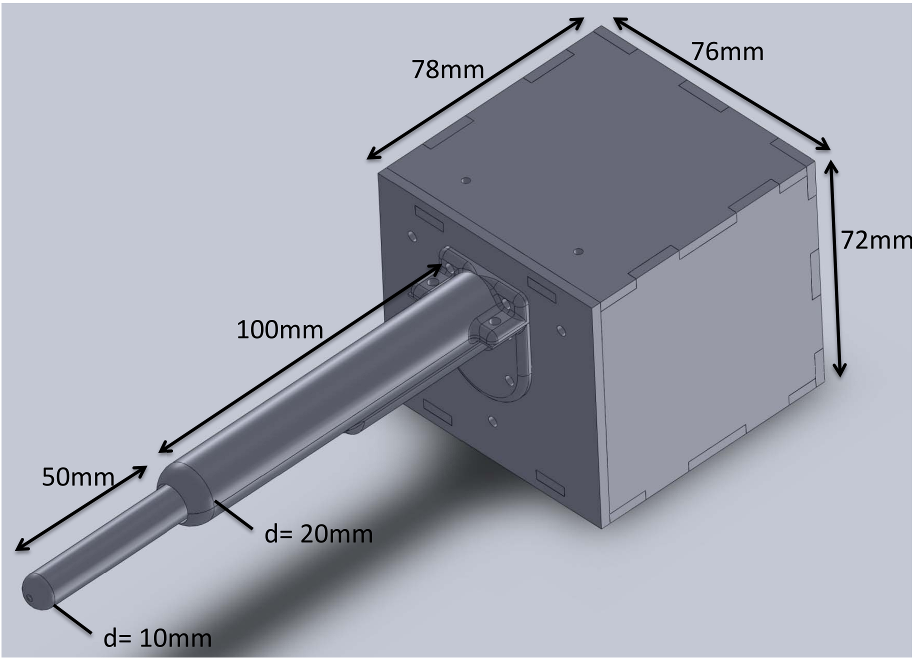

Figure 4.10: Dimensions of *ImpAct*

Other specifications are listed in the Table 4.2.

Specification	**Unit**	**Value**
Weight	Kg	0.243
Ram Span (Min)	mm	3
Ram Span (Max)	mm	50
Voltage	V	5.0
Current (Idle)	mA	50
Current (Max)	mA	250
Residual Friction	N	3.58
Max. Force	N	10.8

Table 4.1: Hardware specifications of *ImpAct*

4.3 Software Implementation

We developed two software interfaces for ImpAct, first is for iPhone (3G 8GB) and second for a tablet PC(SlateDT, Inte Core Duo 1.8Ghz, 1GB, WIndows XP). Both application uses OpenGL library for graphics. iPhone application is written in Objective-C language and tablet PC version uses visual C++ and Java 3D. In order to properly render the projection, it is important to know the position of the viewers eyes. In this prototype system, we assume a predefine position for users head since we have not implemented head tracking. We discuss this in future work section further. Initial software development including all the mobile applications for iPhone is written by my colleagues. I developed a manipulation program which will be further discuss in the applications section. In this section, we will discuss the basic features of fundamental software platform rather than specific applications.
Software system is responsible for two basic functions. First is to render the 3D visualization according to the sensor data acquired from *ImpAct* and second is to transmit the haptic information to *ImpAct*. Rendering the 3D environment is straightforward as described in the Section 3.1.1. Haptic information is generated by detecting collisions of virtual objects and stylus inside the 3D environment and calculating the resultant forces. Then these forces are used to make transformation of virtual objects inside the screen and send information to *ImpAct* to exert the relevant force. Forces are expressed by two parameters, span length of the *ImpAct* and amount of force should be applied. Span length parameter determines the next position of the *ImpAct* ram. This is very important in case of *ImpAct*'s moving shaft has to be forcefully moved to a new location. Force value indicates a 8bit data word indicating the magnitude of the force should be applied. These two parameters are continuously updated at the frame rate of display system.

4.4 Applications

In this section, we will describe some of the prototype applications we developed to explore the capabilities of *ImpAct*. Section introduce selected applications, which can describe *ImpAct*'s capability to provide better manipulation, probing of virtual objects and free form creation environment.

4.4.1 Billiard Game

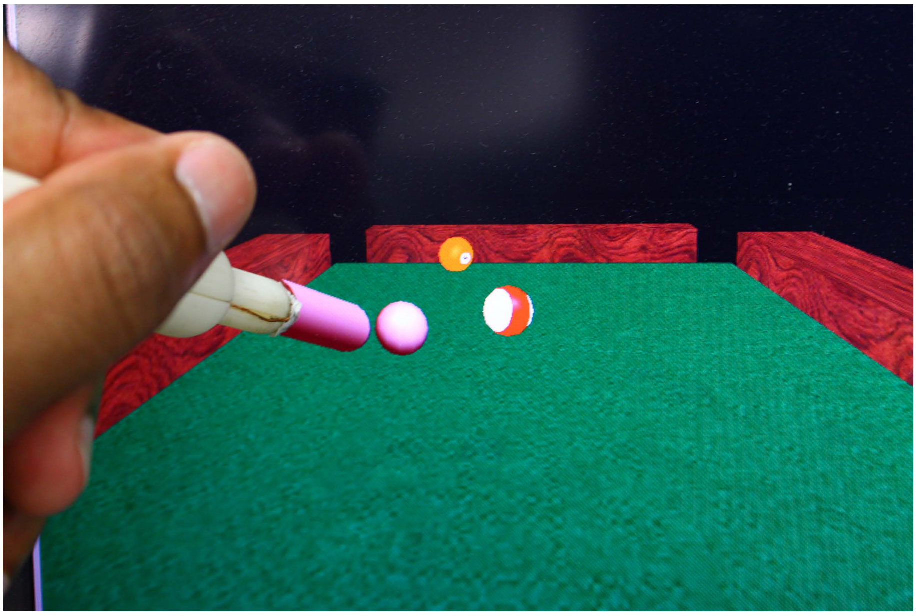

Figure 4.11: Using *ImpAct* to play a billiard game

Billiard game is an application we developed where a user can play billiard on a touch screen based computer using *ImpAct* as the cue. When user hits a ball with the virtual stylus, it calculates impact forces and move along the table. In existing billiard games, users have to instruct the power level using a slider like GUI controller and give the direction of hit separately. In case of *ImpAct*, playing is superfluous since all the parameters are calculated using the orientation of *ImpAct* and the speed user hits the cue ball, exactly similar to how one plays it in real life. Since, *ImpAct* gives sensation of impact forces between the cue and the ball, user has a good feedback to make decisions on the hitting speed according to the ball placements and distances. We believe this helps user to learn the game fast and add considerable enhancement to the game experience. Figure 4.11 shows an image of a user playing billiard using *ImpAct*.

4.4.2 Probing Applications

In this section we will present two probing applications, first one to demonstrate the simulation of static force, second a dynamic force using *ImpAct*.

Shouji

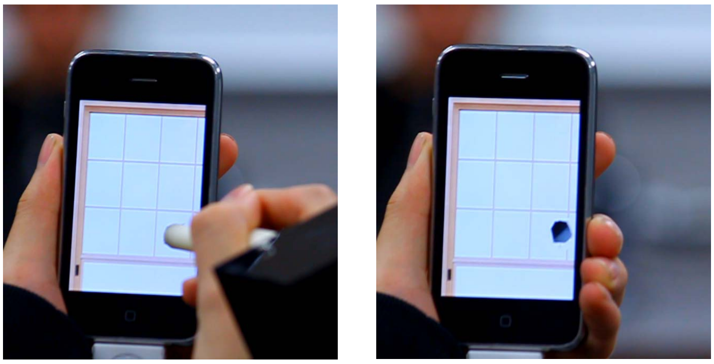
(A) (B)

Figure 4.12: Using *ImpAct* to play Shouji

Shouji is a simple game my colleague Kakehi Gota created to demonstrate the effect of a static force, using *ImpAct*. Shouji is developed on iPhone as a mobile game, where a user can tear a Japanese style paper window to see through to the other side of the window. As shown in the Figure 4.12 (A), user can push *ImpAct* against the paper window to break it. At first, user will feel the stiffness, once the force reaches the breaking point, iPhone application will command *ImpAct* to remove the force making user to feel an impulse and *ImpAct* will go through the paper window. After breaking, user can see the other side of the window via the video captured from iPhone camera as shown in Figure 4.12-(B). We gave this application to some of our colleagues in the lab and they commented they could feel the sensation of tearing the paper window. Such true force-feedback haptic sensations were not possible in previously implemented haptic tools for screen based displays. This haptic simulation includes restricting user's hand to generate the effect of strength of the virtual window paper and impulsive force release to simulate the tearing effect. Two features that enables *ImpAct* to generate these effects are that it can use the screen as the grounding and its capability to actuate, monitor and recalculate the forces using a high speed closed loop controller.

Heart Beat

Figure 4.13: Probing the heart beat of a frog

Heart Beat is an application created by my colleague Makoto Kondo for *ImpAct* to demonstrate dynamic forces. It is a simple iPhone application, which shows a 3D model of an animal. By pointing the tip of *ImpAct* near the heart of the animal shown, user can feel the heart beat of that animal. Figure 4.13 shows an image of probing a frog's heartbeat. In addition to frog, this application can demonstrate humans and a horse heartbeat.
Addition to restricting forces and impulsive releases we described in *Shouji* application, Heart Beat exerts active forces on user's hand making it to forcefully move away from the screen. This haptic simulation can create a active energy transition between user hand and the device. When the device pushes the hand, energy is transferred to the hand so that it perceives the push sensation while when the force is released, device absorbs energy creating a pulling sensation.

4.4.3 Free-form Drawing

Free-form drawing application is developed by Makoto Kondo to demonstrate the expression capability of *ImpAct*. It is not a perfect drawing

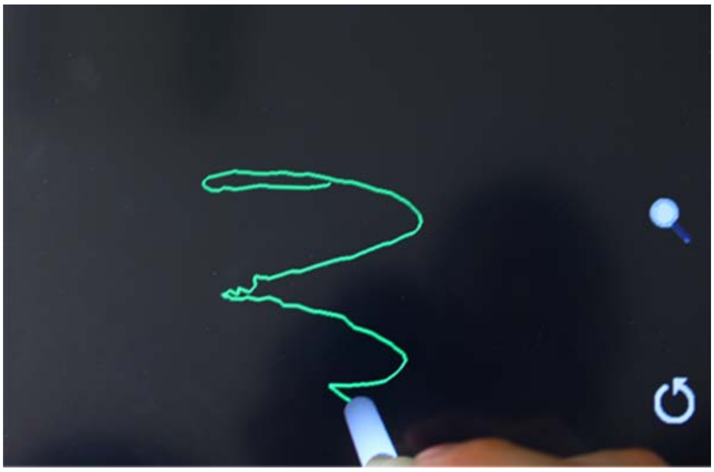

Figure 4.14: Free-form drawing with *ImpAct*

application, however, it is a proof of concept application where a user can draw three dimensionally using *ImpAct*. In general, if a user draws using a generic input device, he/she has to change to each dimension to create 3D sketches. however, in the introduced free-form application, user can utilize z axis movement of *ImpAct* to create 3D drawings directly. Figure 4.14 shows an image of 3D drawing with this application.

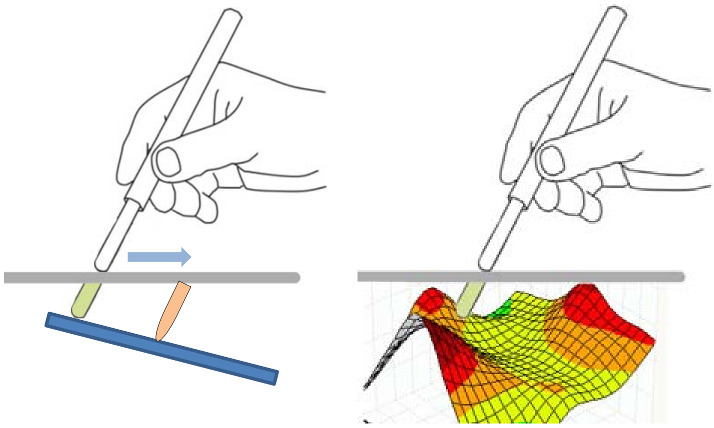

Figure 4.15: a) Restricts the end of virtual stylus to a surface. b) Using haptic informations to draw on an irregular surface.

Furthermore, we can use the haptic features of *ImpAct* to improve the 3D drawing application. We can restrict the virtual tip of *ImpAct* to a given surface by on the surface making user can not move beyond it. Force-feedback can used to restrict user's hand from pushing *ImpAct* beyond the surface limits (Figure 4.15). For an example, a user will be able to draw on an irregular 3D graphics surface as if he/she draws on a similar wall on the real world. We call this haptic assisted drawing.

Chapter 5

Technical Evaluation of *ImpAct*

We conducted a study to evaluate the accuracy and operability of *ImpAct*. Intention of this evaluation is to find how well a user can control the device. Study is divided into two main parts. First part is to examine the errors present in *ImpAct*, which we call device errors. In the second part, we will examine the accuracy of orientation measurements and span length (z axis controllability) of *ImpAct* when a user is asked to achieve a given orientation and depth on the visual display, which we call the controllability of the device by a user. However, we will not measure the accuracy of X,Y dimensions since they are calculated using the existing touch technologies and independent of *ImpAct* measurements.

5.1 Evaluation of Device Errors

Device errors can be further classified into two categories. First are the errors exist in individual sensors due to their sensitivity and stability. Second is the combined errors exist in the system after integration.
Individual sensor errors are already calculated and presented in technical manuals of respective manufacturers. Orientation of *ImpAct* is measured using a combine 3-axis magneto-resistive sensors and 3-axis MEMS accelerometers from manufacturer Honeywell with model name HMC6343. According to the data sheet, typical heading (yaw) and tilt (pitch and roll) accuracies are $\pm 3^o$ and $\pm 2^o$ respectively. Since we are using a analog potentiometer for span length measurement, resolution or the expected error depends on the A/D converter. Since we use 8 bit A/D conversion, expected accuracy for span length measurements is $0.02cm$. These errors exist in the absolute measurements of the sensors. Since we use them for a relative calculations, effect of these errors can be minimized via initial

calibration.
Combine errors exist in the system is calculated by analysing raw sensor measurements presented by the device while it is kept in a steady rest position without any contact to a moving object. We place *ImpAct* on top of a table in a stable position and collected the data presented by the inbuilt micro controller for $10s$ time interval. We calculated the errors of the system measurements, yaw, pitch, roll and span length measurements compared to the *mode* of the dataset. This error value indicates the relative stability of the overall system measurements. Yaw, pitch, roll and span length measurement had average errors of 0.07^o, 0.00^o, 0.05^o and $0.00cm$. Respective standard deviation values were 0.51, 0.11, 0.43 and 0.00. Therefore, we can assume the combine system stability is well enough compared to the absolute errors of individual sensors.

5.2 Evaluation of Controllability

This section describes the evaluation tasks and procedures along with the user groups selected for evaluation of the controllability of *ImpAct*. Goal is to examine combine effect of user errors and system errors in carrying out a specific task with *ImpAct*.

5.2.1 Evaluation Setup

User study software equipped tablet PC was placed on a table and users were given a chair to sit. Additional, since the projected graphics can be changed according to the viewing angle (perspective angle of 3D graphics) a head rest is given to users so that all the users will look into the display from the same position. This chin supporter helps them to keep their head steady at the OpenGL projection camera position so that the user can properly see the rendered scene. And also, users are able to rest there hands on the table between experiments. Figure 5.1 shows the configuration of the head rest to crate a consistent viewing angle for the users.

5.2.2 Users

We evaluated the system with 13 (3 female) voluntary participants with mean age of 29.5 (min 22, max 47) years. All the participants were college

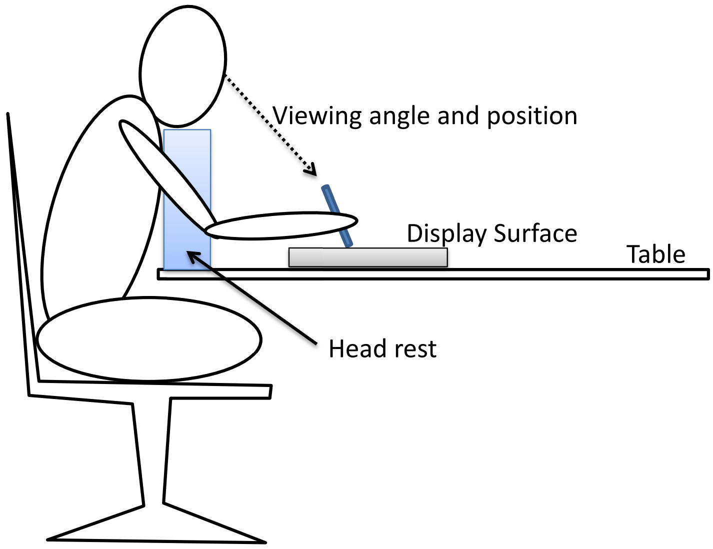

Figure 5.1: Evaluation setup showing configuration of a head rest to create a consistent viewing angle for the users

students (no relationship to the project) and everyday computer users. 10 out of 13 students have had considerable amount of experiences with 3D computer graphics applications and others has little experience in the field. Non of the participants were used *ImpAct* prior to this evaluation and given a basic introduction the operation of *ImpAct* prior to the study. Evaluation took approximately 10 minutes per person and participants were given some snacks and refreshments as a gratitude. After the completion of all the tests, a small verbal discussion was done by the conductor with the study subjects. In this discussion, users were asked to give there feedback and suggestion for *ImpAct*.

5.2.3 Tasks and Procedure

We conducted 3 tests with each user. Three tests are,

1. Calculate involuntary errors of *ImpAct*.
2. Calculate orientation errors of *ImpAct* .
3. Calculate z axis control errors of *ImpAct*.

Calculate involuntary errors

Involuntary errors are the errors occur in measurements without the knowledge of user. For an example, system could trigger a change in span length of *ImpAct* while user did not intentionally push or release it. And also system could measure a change in orientation while user believes he/she is holding the *ImpAct* steady at the same orientation. In this test, users were asked to push and hold *ImpAct* steady for $5s$ time period on the screen. No visual feedbacks were provided what so ever. 10 iterations of angular and radial variables are recorded during this time period to calculate involuntary errors of the system. 130 iterations were recorded for all 13 users.

Calculate orientation errors

In this test, we are calculating angular errors occur in the system when a user tries to orient *ImpAct* according to a given visual guide.

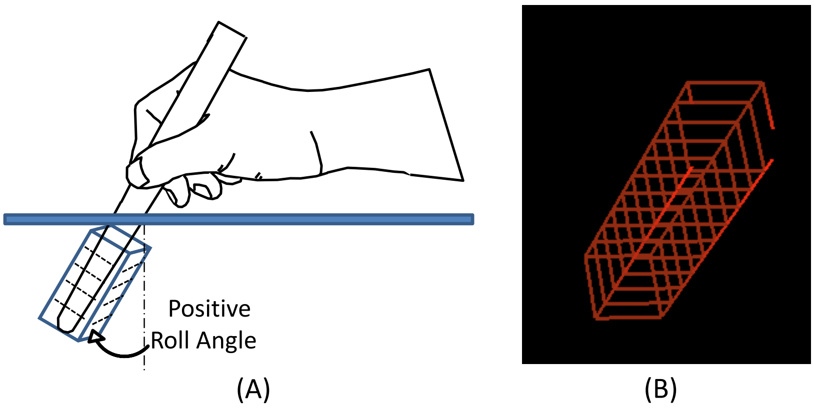

Figure 5.2: Orientation test. (A) concept of the test application, (B) actual guide shown in the display to align *ImpAct*

Figure 5.2 shows a sketch of the task and the actual guide shown to the user on the screen. Once a user become confident that the physical and projected stylus is aligned with the guide, they were ask to press a button on the keyboard (space bar) using the other hand to confirm the test. Guide is placed according to a randomly selected roll values between $\pm 30^o$ with steps of 5^o excluding the angle 0^o. Since the same technology is used, without loss of generality, we only conducted the angular accuracy for roll angles. However, we are hoping to conduct a proper study for pitch angle in future. In this test, per user, 40 iterations are carried out. At each iteration user's alignment angles, guide angles and the time to complete is recorded. Total of 520 iterations were recorded for all 13 users.

Calculate z axis control errors

This test is design to evaluate the controllability of *ImpAct* in the z axis direction by changing its length. Users were given a 3D slider with a highlighted block in it as shown in Figure 5.3 and ask them to locate the end of the rendered stylus within the highlighted area.

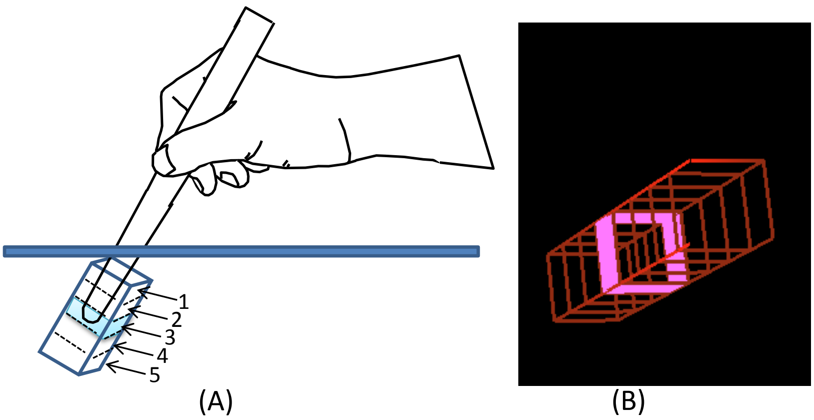

Figure 5.3: z axis control test. (A) concept of the test application, (B) actual guide with highlighted block shown in the display

Test was conducted with 4, 6, 8 and 10 levels per slider and highlighted block was randomly selected. Per each different level, one user was carried out 10 tests, summing to 40 iterations per user and 520 iterations for all 13 users. At each iteration, difference from tip of the projected stylus to middle of the highlighted area is recorded as the radial error in controlling the span length.

5.3 Results

Results of the first test

From the first test, we calculated the involuntary errors in the measurements of *ImpAct*. Average error of yaw, pitch and roll are 0.11^o, 0.22^o and

0.35^o respectively with standard deviation 1.33, 0.63 and 0.53. Error of the calculated span length is 0.37mm with standard deviation 0.177.

Results of the second test

Average completion time for each iteration of the second test was 4.9 seconds. From the total of the 520 iterations, we calculate the average of absolute angular error from the guide roll to the measured *ImpAct* roll. Average error was 5.6^o with a standard deviation of 6.2. Full span of roll angle is $\pm 90^o$ from the z axis of the display. Compared to the full span, error is 3.1%.

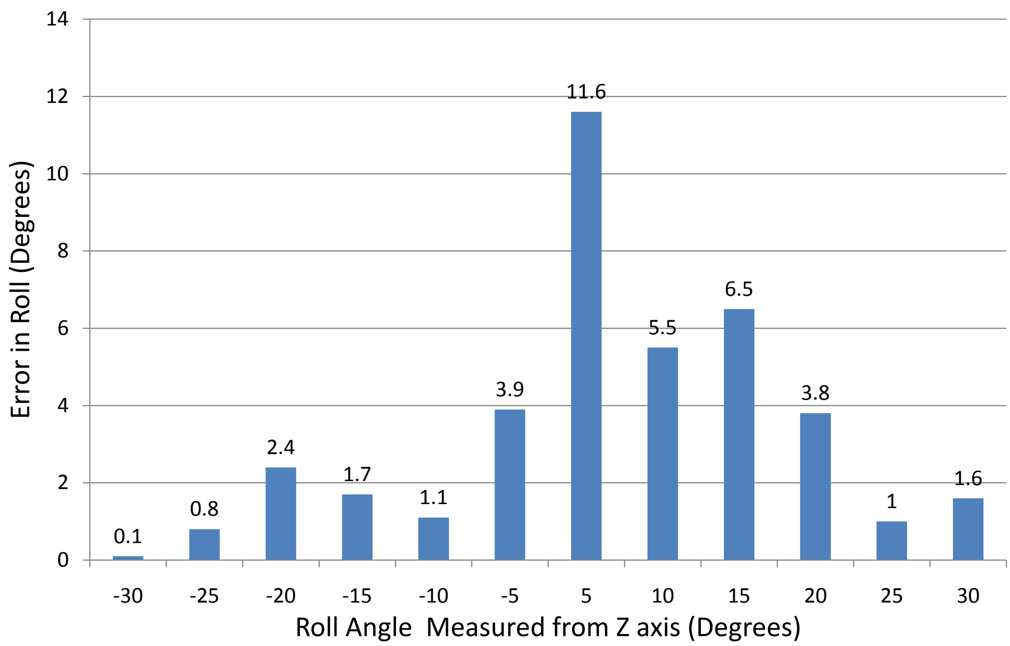

Figure 5.4: Average roll error against the roll angle.

We further analysed the results to find out existence of any relationship between roll angle and the error in roll. Figure 5.4 shows the variation of average absolute error against the roll angle used in guide. According to the graph, we can see the pattern that higher the deviation from the z axis higher the accuracy. And also we can see that when the guide roll value is negative (i.e. guide is tilted inward the user), orientation has much better accuracy than when it is positive.

Results of the third test

Average completion time for third test was 4.46 seconds. We found that the average error of the span length to the actual highlighted area of the given guide is 1.47cm with 0.75 standard deviation. This is a 29.7% of error compared to the full span of *ImpAct*, 5cm. And also, we noted 98% of the time, error made is negative. This means user pushed beyond the required target length.

5.4 Discussion

From the results of the first test, we can conclude that *ImpAct*'s involuntary errors are considerably smaller compared to the full effective measurement range. And a major portion of these errors are contributed by the system stability errors as presented in Section 5.1. Therefore, existence of such errors can be neglected in the operation.
Results of the second test indicates an error about 3.1% (5.6^o) compared to the full range of the roll angle measurement. As mentioned in the Section 5.1, sensors could contribute to $\pm 2^o$ error in roll value resulting possible 3.6^o user error. This is a significant error if *ImpAct* is used for precise operations. We believe that contributing factors to this error is mainly come from the heaviness and bulkiness of the prototype. In the after discussion, users commented that, because of the weight of the prototype made it hard to orient it properly and the back box reduced the handling capability.
Rather interesting finding of the second test is that error get significantly lower at high tilt angles. This is probably because of the fact that it is very easier to visualize the orientation of the guide at high angles. This effect is shown in the Figure 5.5, where (A) has higher roll angle giving it high visual clarity than (B).

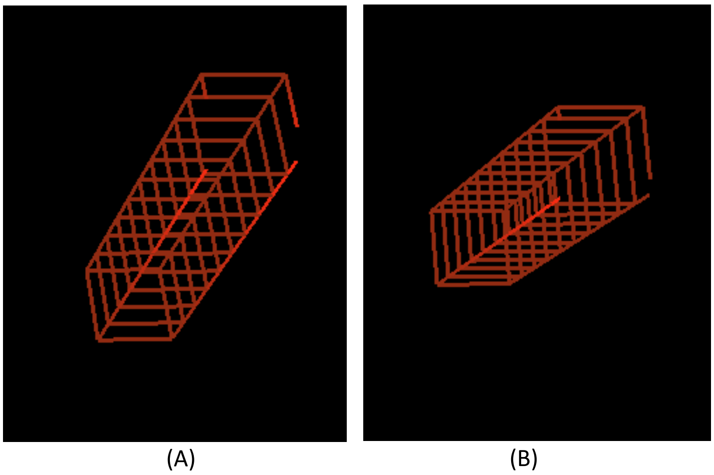

Figure 5.5: Visual clarity at high tilt angles

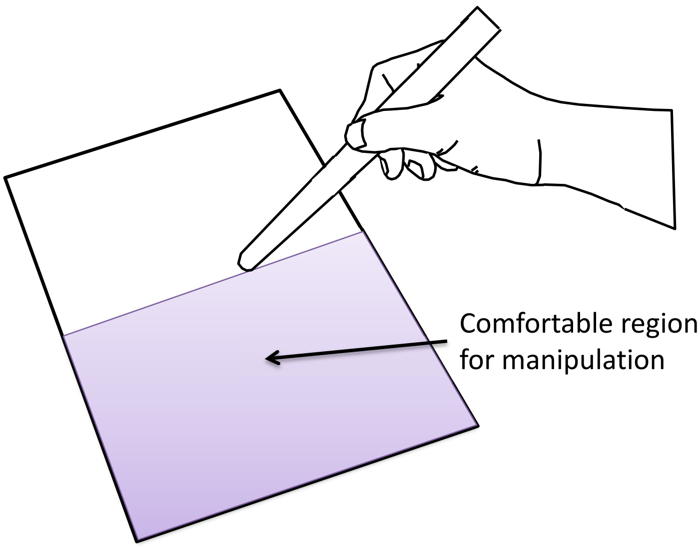

Figure 5.6: Lower part of the screen is more comfortable to interact using *ImpAct*

Another important observation is, at negative angles, accuracy is higher than the positive angles. This means user is comfortable at controlling the *ImpAct* in the region below the touch point than the above area of the screen. This is shown in the Figure 5.6.
Third test results indicate a significant flaw in *ImpAct* in terms of z axis controls. It has a very significant error and we study the reason causing this errors. First cause for this errors is that *ImpAct* moving shaft has a significant redundant friction. This makes it user to hard to move along the z axis and control it smoothly. Second cause is the weight and bulkiness of the prototype makes the user's hand tiring in z axis controls and it leads to considerable amount of human errors. Third cause is that users are not familiar in using tools that are directly move on a z axis. Therefore this concept of precisely controlling the span length is partially new to users.
In the study, addition to the technical factors, we learnt many human factors which holds great importance to the *ImpAct*. In future, we are planning to introduce solutions to existing weaknesses and further study the usability of *ImpAct*.

Chapter 6

Discussion

First part of this chapter describes some design challenges we faced in the process of designing *ImpAct* from the concept of *direct touch*. Latter part of the chapter describes some limitations in the developed *ImpAct* prototype and plausible solutions as future works.

6.1 Design Challenges

We have identified some design challenges in the concept of *ImpAct* in the perspective of direct touch and human computer interaction in general. In this section we discuss some of the selected.
First challenge is that the user's reach to the depth of the virtual world is limited by the maximum spanning length of the *ImpAct*. It is normal to 3D virtual environments to have considerable amount of depth which is well beyond the reach of *ImpAct*. One possible solution is to attach a scaling factor to the virtual stylus so that elongation is multiplied by this factor compared to physical length change. However, user reactions to such abnormalities are unknown and elongating the virtual stylus could lead to instability.
Next important limitation is *ImpAct* is unable to provide the sensation of forces which attracts user's hands towards the screen. *ImpAct*'s force feedback only works for the forces emitting from the surface and not towards the surface. One possible solution is to generate some sort of a magnetic attraction towards the screen. However, these forces are not considered in the implementation of *ImpAct*.
Another important limitation we identified in *ImpAct* is its limitations in giving haptic feedbacks for forces parallel to the screen. Controllability of such forces are highly dependant on the friction between the screen surface and the tip of *ImpAct* and maximum executable force could be very small. Furthermore, as described in the Section 3.2.1, *ImpAct* is unable to provide

movements in the X-Y plane, which is partially related to the limitation described here. One solution would be to attach a small actuator, possibly a rotatable ball to the tip of *ImpAct*.

One of the other limitations we found in the concept of *ImpAct* is the complexities it could cause in a multi-touch and multi-user environment. Currently, software system can identify the touch point of *ImpAct* and render its projection. However, in a multi-touch system, *ImpAct* touch point could be confused between touch by human fingers. A possible solution is to analyse the footprint of the touch and resolve the confusion. It could be further complex in a multi-user environment where two users are using two *ImpAct* devices. One resolution is to *ImpAct* to convey some sort of an identification to the touch surface such as a code.

In conclusion, translation of virtual world characteristics to the haptic simulations are limited by above mentioned challenges. Therefore, in current version of *ImpAct*, haptic force simulations are done only for a limited depth ($10cm$ max using virtual scaling factor). And also the projected haptic sensations are created only for the surfaces directly facing the user. In other words, *ImpAct* does not simulate the forces or impulses created from collisions with surfaces which are perpendicular to the screen surface. And it does not simulate any forces directed towards the screen. Therefore, considerable amount of haptic information are suppressed in the conversion process.

6.2 Limitations and Future Work

In this section, we identify some existing limitations in the *ImpAct* prototype and possible solutions to overcome them.

Most obvious and significant limitation of *ImpAct* is the bulkiness of the prototype and its weight. It greatly reduces the operability of *ImpAct*. Specially the back box used for electronics causes handling a little tricky at high tilt and lower angles. Weight causes users to get tired in short period of time. And also, comfortableness of a haptic device is very important design factor. Weight could cause an uncomfortability to *ImpAct* both in haptics and controlling. Furthermore, bigger form factor could occlude the display screen and also considerably reduces the attractiveness. We are planning to implement the scaled down version of *ImpAct* by moving the processing components and some of the electronics to an external box and only keeping the vital components inbuilt.

Second limitation is the residual friction exist in the ram. As described in the Section 5.4, it causes low controllability of span length of *ImpAct*. And also, residual friction makes users to feel a force all the time, which is undesirable in a haptic display[30]. This friction component is made by gear mechanism used in the motor, wiper actuator used to actuate the potentiometer. In future, we are going to use tension cables to transmit the energy from motor to the moving shaft and eliminate mechanical contact for encoding. We believe this will significantly reduce the residual friction.

Another limitation in current prototype is that the existence of perspective visual impurities in the rendered projection due to unavailability of head tracking. In order to properly display 3D content, it is important for rendering system to know the viewers eye position. Since we have not implemented head tracking, rendered projection of *ImpAct* could not purely align with the physical one. We are in the process of implementing head tracking for *ImpAct*.

Another addition to *ImpAct* in future will be an activating button. Currently, *ImpAct* does not have any buttons, therefore, when there is a need for something similar to *clicking* in mouse, users has to use gestures or keyboard. Therefore we are planning to integrate a button to *ImpAct*.

Chapter 7

Conclusion

In this thesis, I presented the concept of *Direct Touch* and *Manipulation* techniques for surface computing environments and introduced *ImpAct* as a tool and a proof of concept for implementing direct touch. Direct touch is meant to provide a spatially coincident haptic and visual display system along with free-form interaction within a given digital space. *ImpAct* is designed to adhere to direct touch concept by providing a visually continuous haptic tool for surface computers using a scalable stem and a projected virtual stylus. I believe direct touch implements more realistic and meaningful haptic interface than existing technologies and it provides better means to express user's mind to a computer in the context of 3D applications.

I explore the challenges and barriers of implementing a direct touch interface through *ImpAct* and described the concept design and implementation process in this thesis. However, there are many unsolved challenges of direct touch concept which I could not implement in *ImpAct*. I present some of those challenges in this thesis, so that, probably a reader or I myself in future can come up with solutions.

At this stage, considering *ImpAct* as stylus for surface computers may seem far realistic. However, as Moore's law says, in future *ImpAct* could be integrated into a form factor of the size of an average pen or a pencil. Actuation technologies are advancing to provide high power in a small package. Sensing technologies are getting integrated and computing power for unit space is doubling rapidly. Power usage of each of these technologies become smaller while batteries are built to provide longer energy life.

It is interesting to think about the future perspective of *ImpAct* as a human computer interaction tool to enable direct touch on surface computers. First application comes to my mind is artistic creation environments which need multiple degrees freedom and realistic feedback. For an example, there will be computer applications where a user can sculpt a 3D model using *ImpAct* as sculpturing tool, artist can carve a 3D model or create textures

on surfaces as now they do using *ZBrush*[1]. More recent application would be to manipulate 3D models in CAD/CAM applications.
ImpAct can be used as a gear for gaming as I presented in the billiard application in Section 4.4. *ImpAct*'s manipulation features are well suited for games such as ball games, first person shooter games and weapon manipulation games. Not only *ImpAct* can provide multiple degree of freedom, but also it will provide realistic direct touch sensation for user.
ImpAct would be good tool for medical field as a remote operation tool. Doctors could use *ImpAct* as a probing tool for diagnosis applications such palpation. Furthermore it can be used for surgery as a remote invasive tool such as scalpel.
It is clear that the direct touch technology will be a promising interaction strategy for future computers. As we presented using *ImpAct*, *direct touch* is plausible to implement, can enable number of non trivial interaction possibilities and has a clear path forward with potential future applications. I believe, direct touch concept and *ImpAct*'s design implications will open the door to a new direction in human computer interaction.

[1] A software tool used to create textures on 3D models. For more information `http://www.pixologic.com/home.php`

Bibliography

[1] Pär-Anders Albinsson and Shumin Zhai. High precision touch screen interaction. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 105–112, New York, NY, USA, 2003. ACM.

[2] Hideyuki Ando, Takeshi Miki, Masahiko Inami, and Taro Maeda. The nail-mounted tactile display for the behavior modeling. In *SIGGRAPH '02: ACM SIGGRAPH 2002 conference abstracts and applications*, pages 264–264, New York, NY, USA, 2002. ACM.

[3] J. J. Batter and F. P. J. Brooks. Grope-1: A computer display to the sense of feel. In *Information Processing, Proceedings of IFIP Congress*, pages 759–763, 1971.

[4] Xiaojun Bi, Tomer Moscovich, Gonzalo Ramos, Ravin Balakrishnan, and Ken Hinckley. An exploration of pen rolling for pen-based interaction. In *UIST '08: Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 191–200, New York, NY, USA, 2008. ACM.

[5] D.L. Bitzer and R.L. Johnson. Plato: A computer-based system used in the engineering of education. *Proceedings of the IEEE*, 59(6):960 – 968, june 1971.

[6] Jon Driver and Charles Spence. Attention and the crossmodal construction of space. *Trends in Cognitive Sciences*, 2(7):254 – 262, 1998.

[7] George W. Fitzmaurice, Hiroshi Ishii, and William A. S. Buxton. Bricks: laying the foundations for graspable user interfaces. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 442–449, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[8] J. J. Gibson. *The Theory of Affordances.* Lawrence Erlbaum, 1977.

[9] James J. Gibson. *The Ecological Approach To Visual Perception.* Lawrence Erlbaum Associates, first edition, September 1986.

[10] Rob Gray and Hong Z. Tan. Dynamic and predictive links between touch and vision. *Experimental Brain Research*, 145(1):50–55, 2002.

[11] Jefferson Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118, New York, NY, USA, 2005. ACM.

[12] Jefferson Y. Han. Multi-touch sensing through frustrated total internal reflection. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Sketches*, page 145, New York, NY, USA, 2005. ACM.

[13] Paul M. Hildreth and Chris Kimble. The duality of knowledge. *Information Research*, 8, 2002.

[14] Matthew Hirsch, Douglas Lanman, Henry Holtzman, and Ramesh Raskar. Bidi screen: a thin, depth-sensing lcd for 3d interaction using light fields. In *SIGGRAPH Asia '09: ACM SIGGRAPH Asia 2009 papers*, pages 1–9, New York, NY, USA, 2009. ACM.

[15] Takayuki Hoshi, Masafumi Takahashi, Kei Nakatsuma, and Hiroyuki Shinoda. Touchable holography. In *SIGGRAPH '09: ACM SIGGRAPH 2009 Emerging Technologies*, pages 1–1, New York, NY, USA, 2009. ACM.

[16] M. Ishii and Sato. M. A 3d spatial interface device using tensed strings. *PresenceTeleoperators and Virtual Environments*, 3(1):81–86, 1994.

[17] T Iwamoto, T. Maeda, and H. Shinoda. Focused ultrasound for tactile feeling display. In *ICAT 2001: Int Conf Artif Real Telexistence*, Tokyo, Japan, 2001.

[18] H. Kajimoto, N. Kawakami, Maeda T., and Tachi S. Tactile feeling display using functional electrical stimulation. In *ICAT'99: Int Conf Artif Real Telexistence*, Tokyo, Japan, 1999.

[19] Hiroyuki Kajimoto, Naoki Kawakami, Susumu Tachi, and Masahiko Inami. Smarttouch: Electric skin to touch the untouchable. *IEEE Comput. Graph. Appl.*, 24(1):36–43, 2004.

[20] John Kammersgaard. Four different perspectives on human-computer interaction. *Int. J. Man-Mach. Stud.*, 28(4):343–362, 1988.

[21] Sho Kamuro, Kouta Minamizawa, Naoki Kawakami, and Susumu Tachi. Pen de touch. In *SIGGRAPH '09: ACM SIGGRAPH 2009 Emerging Technologies*, pages 1–1, New York, NY, USA, 2009. ACM.

[22] Steffan Kennett, Marisa Taylor-Clarke, and Patrick Haggard. Noninformative vision improves the spatial resolution of touch in humans. *Current Biology*, 11(15):1188–1191, 2001.

[23] Myron W. Krueger, Thomas Gionfriddo, and Katrin Hinrichsen. Videoplace—an artificial reality. In *CHI '85: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 35–40, New York, NY, USA, 1985. ACM.

[24] Ki-Uk Kyung and Jun-Young Lee. wubi-pen: windows graphical user interface interacting with haptic feedback stylus. In *SIGGRAPH '08: ACM SIGGRAPH 2008 new tech demos*, pages 1–4, New York, NY, USA, 2008. ACM.

[25] Paul Lapides, Ehud Sharlin, Mario Costa Sousa, and Lisa Streit. The 3d tractus: A three-dimensional drawing board. In *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 169–176, Washington, DC, USA, 2006. IEEE Computer Society.

[26] Johnny C. Lee, Paul H. Dietz, Darren Leigh, William S. Yerazunis, and Scott E. Hudson. Haptic pen: a tactile feedback stylus for touch screens. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 291–294, New York, NY, USA, 2004. ACM.

[27] SK Lee, William Buxton, and K. C. Smith. A multi-touch three dimensional touch-sensitive tablet. *SIGCHI Bull.*, 16(4):21–25, 1985.

[28] J. C. R. Licklider. Man-computer symbiosis. *IEEE Ann. Hist. Comput.*, 14(1):24, 1992.

[29] Thomas H. Massie and J. K. Salisbury. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME Winter Annual Meeting*. Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 1994.

[30] Thomas Harold Massie. Initial haptic explorations with the phantom : virtual touch through point interaction. Master's thesis, Massachusetts Institute of Technology, 1996.

[31] N. Metha. A flexible machine interface. Master's thesis, Department of Electrical Engineering, University of Toronto., 1982.

[32] Margaret Minsky, Ouh-young Ming, Oliver Steele, Frederick P. Brooks, Jr., and Max Behensky. Feeling and seeing: issues in force display. *SIGGRAPH Comput. Graph.*, 24(2):235–241, 1990.

[33] Margaret Diane Rezvan Minsky. *Computational haptics : the Sandpaper system for synthesizing texture for a force-feedback display.* PhD thesis, Massachusetts Institute of Technology, 1995.

[34] Ikujiro Nonaka. *The Knowledge-Creating Company (Harvard Business Review Classics).* Harvard Business School Press, December 2008.

[35] Renaud Ott, Daniel Thalmann, and Frederic Vexo. Haptic feedback in mixed-reality environment. *Vis. Comput.*, 23(9):843–849, 2007.

[36] Michael Polanyi. *Personal knowledge : towards a post-critical philosophy by Michael Polanyi.* Univ. Press, Chicago :, 1959.

[37] Gonzalo Ramos, Matthew Boulos, and Ravin Balakrishnan. Pressure widgets. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 487–494, New York, NY, USA, 2004. ACM.

[38] Jun Rekimoto. Smartskin: an infrastructure for freehand manipulation on interactive surfaces. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 113–120, New York, NY, USA, 2002. ACM.

[39] Toshiki Sato, Haruko Mamiya, Hideki Koike, and Kentaro Fukuchi. Photoelastictouch: transparent rubbery tangible interface using an lcd and photoelasticity. In *UIST '09: Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 43–50, New York, NY, USA, 2009. ACM.

[40] K. B. Shimoga. A Survey of Perceptual Feedback Issues in Dexterous Telemanipulation: Part II. Finger Touch Feedback. In *Proc. of the IEEE Virtual Reality Annual International Symposium*. Piscataway, NJ : IEEE Service Center, 1993.

[41] Michael Sinclair. The haptic lens. In *SIGGRAPH '97: ACM SIGGRAPH 97 Visual Proceedings: The art and interdisciplinary programs of SIGGRAPH '97*, page 179, New York, NY, USA, 1997. ACM.

[42] Christopher M. Smith. Human factors in haptic interfaces. *Crossroads*, 3(3):14–16, 1997.

[43] E. So, Hong Zhang, and Yi sheng Guan. Sensing contact with analog resistive technology. In *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, volume 2, pages 806 –811 vol.2, 1999.

[44] I. E. Sutherland. Sketchpad a man-machine graphical communication system. In *25 years of DAC: Papers on Twenty-five years of electronic design automation*, pages 507–524, New York, NY, USA, 1988. ACM.

[45] Yu Suzuki, Kazuo Misue, and Jiro Tanaka. Stylus enhancement to enrich interaction with computers. In *HCI'07: Proceedings of the 12th international conference on Human-computer interaction*, pages 133–142, Berlin, Heidelberg, 2007. Springer-Verlag.

[46] S. Tachi and T. Sakaki. Impedance controlled master slave system for tele-existence manipulation. In *Proceedings of the 1st International Symposium on Measurement and Control in Robotics (ICMSR '90)*, pages B3.1.1–8. Houston, Texas, USA (1990.6), 1990.

[47] Feng Tian, Lishuang Xu, Hongan Wang, Xiaolong Zhang, Yuanyuan Liu, Vidya Setlur, and Guozhong Dai. Tilt menu: using the 3d orientation information of pen devices to extend the selection capability of

pen-based user interfaces. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1371–1380, New York, NY, USA, 2008. ACM.

[48] Feng Wang, Xiang Cao, Xiangshi Ren, and Pourang Irani. Detecting and leveraging finger orientation for interaction with direct-touch surfaces. In *UIST '09: Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 23–32, New York, NY, USA, 2009. ACM.

[49] Feng Wang and Xiangshi Ren. Empirical evaluation for finger input properties in multi-touch interaction. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 1063–1072, New York, NY, USA, 2009. ACM.

[50] Pierre Wellner. The digitaldesk calculator: tangible manipulation on a desk top display. In *UIST '91: Proceedings of the 4th annual ACM symposium on User interface software and technology*, pages 27–33, New York, NY, USA, 1991. ACM.

[51] R.L. Williams II. Cable-suspended haptic interface. *International Journal of Virtual Reality*, 3(3):13–21, 1998.

[52] Andrew D. Wilson, Shahram Izadi, Otmar Hilliges, Armando Garcia-Mendoza, and David Kirk. Bringing physics to the surface. In *UIST '08: Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 67–76, New York, NY, USA, 2008. ACM.

[53] Jeffrey A. Wolfeld and Aravind K. Joshi. Real time control of a robot tactile sensor, 1981.

Appendix A

Additional Images

Figure A.1: Using *ImpAct* as a mobile stylus

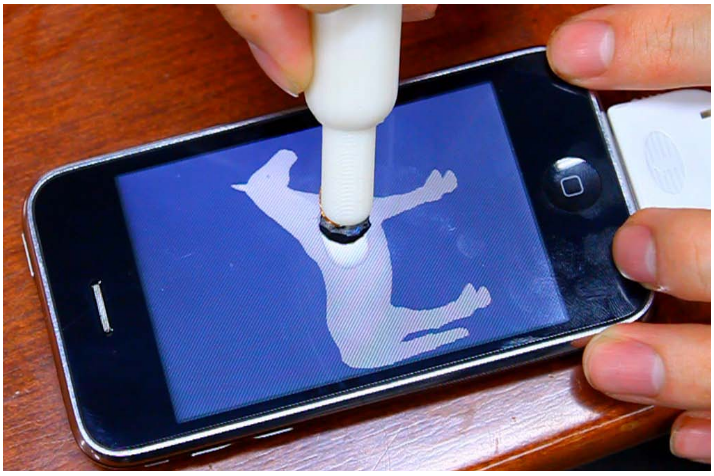

Figure A.2: Horse model in Heart Beat application