Master Thesis

Design of Tracking and Interaction Techniques for Touch-sensitive Tangibles in Tabletop Environments

（テーブルトップ環境におけるタッチ認識可能なタンジブルのためのトラッキングおよびインタラクション手法の設計）

Saphyra Vânia Marina Amaro – 学生番号 37-116882

Supervisors: Prof. Dr. Takashi Chikayama

Prof. Dr. Masanori Sugimoto

Department of Electrical Engineering and Information Systems
Graduate School of Engineering
The University of Tokyo

Submitted on February 6th, 2013

To my Mother Thelma,
who always supported me,
especially in my life's hardest decisions.
No matter what happened,
she always is there for me,
even if we are physically a world apart.

To my Grandmother Therezinha,
who would always be worried
about me, even when
I tell her I'm OK.

I love you, and I would not be here if it were not for you.

ACKNOWLEDGEMENTS

Before starting all the tech talk in this master thesis, I would like to express my gratitude for all the help that I got, ever since I was a research student. Space is limited, though, so it is impossible to list all the people, whom I regard to as friends, that I met and helped me, directly or indirectly, to be where I find myself now. I thank you all from the bottom of my heart.

First of all, words cannot express my gratitude to Professor Masanori Sugimoto. Was it not for him, I would not even be in University of Tokyo, or not even in Japan, for he has accepted me as a research student in his laboratory, the Interaction Technology Laboratory, back in 2010. Professor Sugimoto has supported me even in my hardest times as a student, especially when our research does not work as expected. Also, he taught me how you are supposed to think and investigate as a researcher, and how to have critical sense when it comes to new ideas. Sugimoto sensei also showed me and our lab members that it is possible to be a strict but very kind supervisor, even though that sounds like a paradox. I feel deeply indebted to him, and I feel I could not and will never be able to repay him for all he has done to me. It is a pity, but I feel I may not have learned all the things that I could from him, and I would like to able to learn more from him in the future.

I am deeply thankful to Professors Takashi Chikayama and Yoshimasa Tsuruoka. Thanks to them, I was able to proceed in my research, even though our tabletop took much space of their room.

I would like to thank my classmate Yusuke Takeuchi. I really believe that he changed our laboratory's atmosphere, with his cheerful personality. My years during the master course were really fun also thanks to him. Also, we went through the same problems and dilemmas that I did, so we could discuss a lot about several topics in a very informal and frank way. I could learn a lot from him – he is very kind, considerate, focused on his objectives, and I am sure that this will bring him much success and happiness in his professional and personal life in the future. I hope that I make friends with the ease that he does someday.

I also am very grateful to Carlos Espinoza, my only South American pal in the lab and currently almost my neighbor, that supported me so much. It was very fun being in the same lab as he was, even though it was for so little time. I hope that I can help him as much as he did to me in his academic life – and I also hope that we have the chance to watch a Timão match together.

My thanks also to Jun Miura, a very kind classmate from the Electrical and Electronic Engineering Course. I am yet to find someone so conscious and mature as he is for his age. I have very nice memories of the times that I could hang out with him and Yusuke, especially in the last weeks before the thesis submission.

I cannot thank enough my Thai seniors Nuttapol Sansugsuriyachot and Natsuda Laokulrat. Nuttapol gave me a lot of advice about the student life at Todai and that discussed so much about interaction techniques in tabletop systems with me, besides sharing the same general interests that I did. Natsuda was my role model for a girl studying Engineering, because of her dedication and knowledge of the area, without never forgetting her kind side. She was always willing to help even if the task took a while to finish.

I would like to thank my senpais Kyohei Mizutani, who was my tutor and that had to keep up with my broken Japanese from my earliest days as a research student, and Shigeki Nakamura. They helped me a lot with general doubts about the University, especially about the entrance examination, and Japan in general. I had them as my role model of how to be a good person – how to be serious in your studies, but without being dull.

I am very thankful also to the PhDs, now Professors, that graduated from ITL, Natapon Pantuwong and Mi Haipeng, who I really looked up to, as they are extremely talented and serious researchers. Beside being very nice people, they were always ready to help a junior even if they were busy with their own affairs. Their students will be very fortunate for having such excellent and kind professors by their side.

Of course I would not forget my Chinese juniors Lin Runzhi and Zhang Conying. I am very grateful to them for cheering me up and helping me in my experiments. And also for all the nice friendly talks over nabe we had in our old laboratory, along with all the other members.

I also need to mention my gratitude to the Interaction Technology Laboratory's Robogroup members Thitirat Siriborvornratanakul, Masanori Inoue, Tomoki Fujita, Dao Vinh Ninh, Eiji Seki, Hiroki Narukawa and Noriaki Kakiuchi. I was very fortunate to be in such a talented research team. I cannot forget my thanks to the Usonic team – Tomohiko Sato, Koutaro Terabayashi, Keita Furui, Yuuki Inoue and

Noriyoshi Kanie. ITL was like a family to me. Thank you all very much.

From outside the ITL but still in the University, I would show my appreciation especially to my Peruvian friends Eduardo Paucar and Harold Suarez; classmates Lin Li, David Gutierrez and Wing-Ho Ko. It was very fun being around such nice and intelligent students. It makes me sad that it will not be possible to meet everyday all these people that I used to in the University. I will really miss that.

Now, from outside the University, I really need to thank Aleksander Krzywinski, with whom I had very serious discussions on tabletop interfaces and tangibles when coming back from ITS 2011. That talk surely inspired me in keeping going on with my touch-sensitive tangible idea, which I had not much faith in back then.

I cannot forget to express my gratitude to Masahiro Kozaki, who encouraged me so much to study hard and not let myself down in spite of bad results, especially before the entrance examination. I hope that I can encourage him to pursue his dream of becoming a Mathematician.

I cannot thank Diego Vasques enough, as he always patiently listened to my doubts regarding my research, as well as personal and student life rants over donuts and questionable taste Brazilian music videos that made us laugh so hard in Shimokitazawa. He is always there for anyone that needs help, is very kind, and very fun. He is undoubtedly one of my best friends. And he has an awesome taste for games.

Also, my sincere thanks to Felipe Pepe, who came to visit me and had the patience of reading this whole dissertation even though his major is Marketing, just to check my text, point out contradictions and paragraphs that did not make sense. And also for all the serious game industry discussions, which are really interesting.

Special mention to Pedro Marques and Roberto Carapeto, my Brazilian older brothers in Japan, despite the fact that I'm older than them. They always supported me ever since I got to Japan, encouraging me to study Japanese and to give my best in the examinations. I really miss living in Komaba, where we could meet everyday and have dinner together, along with Murilo Roggia, Marcos Persici and Julio Worman. We had a place to call home and a family to talk to, even though we were in Japan. My sincere thanks to all of them.

I cannot forget to mention and thank also João Ota, Letícia Kato, Yul Otani, Takuya Keima, Sonsu Sakai, Andre Euphrasio, Bruna Cavassani, Waleska Giordano and Édison Aparecido Jr. – friends who indirectly contributed to this research, by always being by my side.

Last, but not least, I would like to thank my mother Thelma Rezende with all my heart, as were it not for her, I would not be anything in life. Even being in the other side of the planet, she encourages me and gives advice, no matter what time of the day. I am a very fortunate being for having such a nice and intelligent person as a mother. I admire and love you with all my heart. I also cannot forget my grandmother Therezinha Nobrega, who always is cautious of calling me and waking me up, even though it is daytime in Japan. I cannot thank you two enough for all you have done and borne because of me.

In the end, I would like ask the people that are important to me but do not figure in this list to accept my gratitude along with my sincere apologies. It is very hard to thank all people involved, one by one. The list became very big, but it is not as big as the appreciation that I have for all the kindness you have shown toward me.

Thank you all for your support, care and company. I hope I live up to your expectations and that I can return double everything you have done for me.

Saphyra Vânia Marina Amaro
February 6th, 2013.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Design of Tracking and Interaction Techniques for Touch-sensitive Tangibles in Tabletop Environments
by
Saphyra Vânia Marina Amaro

Supervisor: Masanori Sugimoto

This work's goals are to propose a tracking method and interaction techniques for a touch-sensitive tangible, to be utilized in combination with a tabletop interface. The proposal is done having in mind specifically a tabletop CAD application as a target application for the methods here discussed.

The interaction techniques combine the touch-sensitive tangible and touches on the tabletop to assist 3D manipulation. The tangible is used as an avatar of the virtual objects. The user is supposed to interact with the tangible – rotating and translating – as if s/he was actually performing the action with the 3D model.

The tracking system requires the attachment of an active marker to the tangible. The marker is tracked by employing the same camera utilized for touch tracking. The method combines computer vision algorithms to detect the infrared light-emitting diodes that form the tangible's active markers, and also does the multiplexing of the tabletop's infrared lighting and camera's signal using a time-division strategy, allowing both trackings to be performed simultaneously.

The main contributions of this work are two. The first one is the proposal of a touch-sensitive tangible and novel interaction techniques employing the tangible in combination with a tabletop interface for 3D manipulation. The second is the proposal of a tangible tracking method method that allows mid-air tracking and also provides estimates of the tangible's distance from the table, employing the same camera utilized for touch recognition.

I

Introduction

1.1 Background

The first personal desktop computers as we know – in other word, that were fully programmable and that fit on the top of a desk – were the HP 9800 series, which were commercialized in the 1970s. These early desktops had LED displays, keyboard and could read magnetic cards containing programs. Depending on the model, a thermal printer also be connected and utilized for printing the results obtained from the programs. The first mouse prototype was built in 1963, by Douglas Engelbart, and the first computers to utilize a mouse as input appeared in 1970.

In the following 30 years, displays, mice, keyboards and printers evolved and new peripherals were invented. But, in spite of all these technological advances, the interface utilized to operate desktops or laptops had not changed much ever since. Though, ever since the launch of iPhone and iPad by Apple, which made popular the utilization of touch-sensitive interfaces, the computer interface paradigm is finally changing towards more intuitive interfaces. As an example of that, recently released laptops come with touch-sensitive displays, whose touch input is also supported by the operating system. But, still, the computers that we use to browse through information – the smartphones and tablets – are the ones whose interface is changing more rapidly. The computers we actually use to work, namely desktops and laptops, still utilize the same interaction paradigms, adding only touch-sensitive displays. And, because of that, they still do not incentive or facilitate collaborative activities, requiring that each member of a certain group work on his/her share of the task in his/her personal device, updating or making available his results upon completion of the task for all the group. The members will visualize the results in their own personal device. In a typical working environment, there is no common-use computer in which all members can perform a task simultaneously, while discussing.

In that sense, tabletop interfaces were born from the idea of creating a big touch-sensitive computer for encouraging collaborative tasks. Currently, all collaborative tasks are done individually on each team member's PC but, in the future, they will probably be done in such collaborative common spaces, and it will be possible to seamlessly download the task or its result to your own personal space as well, as proposed in MobiSurf *Seifert et al.* (2012).

The initial sketch of what could be called a Tabletop Interface (TI) was made by Dr. Vannevar Bush in 1945 (*Bush* (1945)). It was called a "Memex". This device would allow its owner to store all of his books and records, and would support annotations, providing flexible and fast access to all these data by indexing it. It would also have means of communicating to other devices of the same nature, allowing content sharing between researchers. The Memex would be integrated in a large desk, equipped with levers, buttons, microfilm cameras and readers for storing and retrieving data, a place in which the user could write notes with his/her own calligraphy to be stored as an image into the microfilm, and also two translucent screens in which information could be displayed. Bush himself described it as a "a sort of mechanized private file and library". A drawing of his idea can be seen in Figure 1.1.

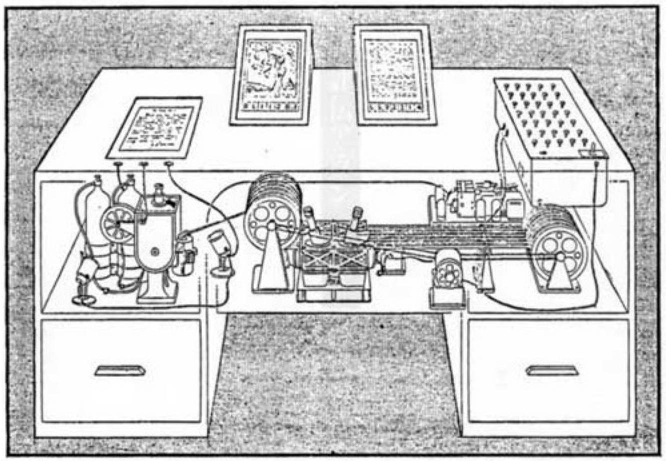


Figure 1.1: Memex, as envisioned by Dr. Vannevar Bush in 1945.

But it was, at that time, a concept. The first real implementation of a TI was done in 1993 (*Wellner* (1993)), and TIs have been largely researched since then. Tabletops are being used especially for educational purposes or in exhibitions, but in an experimental fashion. Even though Microsoft PixelSense[1] (*Microsoft PixelSense 2013*) is a TI commercialized as a product for normal consumers, tabletops have not been put into real work environments. Besides practical reasons such as price – currently a PixelSense by Microsoft costs around US$7500 – and the space it requires, one possible explnation for tabletops to not have been largely adopted yet lies in the article "The Long Nose of Innovation", by Bill Buxton (*Buxton* (2008)). In his article, Buxton states that even though we are living in times of fast technological advances, it takes time – sometimes 10 or 20 years – for a certain technology to reach a major adoption stage. According to him, an innovative idea has to mature a lot before it is largely utilized (Figure 1.2).

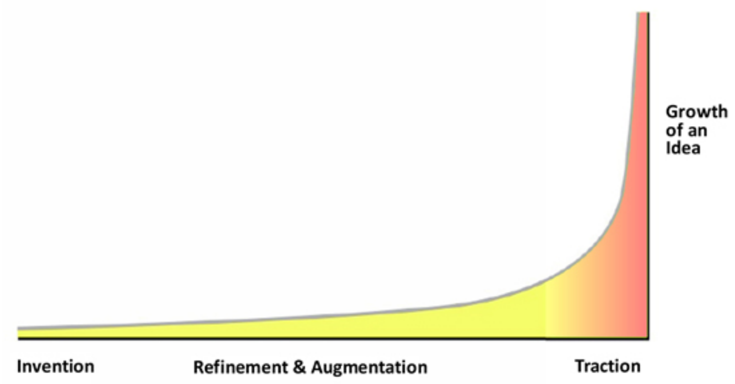


Figure 1.2: Growth x maturity stage of an idea, according to Buxton.

Until we reach a "major adoption" stage for tabletop interfaces, researchers will keep improving both the technology involved in building a TI as well as developing new interaction methods.

In that sense, this dissertation contains ideas and results that were obtained as results of a research focused in the combination of Tangible User Interfaces and Tabletop Interfaces, therefore bringing a small contribution to the future adoption of Tabletop Interfaces in our everyday lives.

As results of the research conducted in the University, this dissertation contains the proposal of a touch-sensitive tangible to be utilized in a tabletop CAD application and also the proposal of a novel tracking method for tangibles. A contextualization about each of these two proposal's problem space can be found below.

---

[1] Previously called Microsoft Surface.

1.1.1 Touch-sensitive tangible

Touching is one of the simplest and most intuitive ways of interacting with objects – but, somehow, this kind of interaction is still rarely employed when it comes down to Tangible User Interfaces.

In the literature involving tangible tabletop interfaces, two modes of interaction that occur frequently are tangible translation and tangible rotation. These two actions are easily recognizable by the technologies employed in finger and fiducial tracking on tabletop interfaces. Some researchers have explored other techniques for tangible interaction, such as pressure sensitivity or shape changing. However, interacting via touch still remains underexplored, and the majority of implementations of tangible tabletop interfaces cannot detect touch. Detecting such an event could invoke special features in a tabletop application. One of the goals of this work is to discuss how touch-sensitive tangibles can enrich the interaction experience and suggest some ideas for applications utilizing such tangibles.

1.1.2 Tangible tracking

In tabletop tangible interfaces, tangible tracking is usually achieved using the interface's own camera, which detects the position and orientation of a tangible by processing images of its markers. This method is limited because tangible markers need to be on the tabletop surface so the camera can identify them. Thus, this tracking method cannot be applied if the interface includes tangibles that can be removed from the surface or tangibles specialized for mid-air interactions. Therefore, one of the goals of this work is to propose a novel method for tangible tracking that does not require tangibles to be above the tabletop's surface to be tracked and permits the simultaneous tracking of touches and tangibles using the same camera for each task.

1.2 Goals

The main goals of this dissertation are to describe in detail the proposal and also the implementation of a tracking method and interaction techniques for a touch-sensitive tangible. It is combined to a tabletop interface with the purpose of achieving better 3D manipulation. It also contains discussions regarding issues observed during research that justify certain design decisions or results obtained during experiments. It is important to note, though, that all considerations written in this work are done having a tabletop CAD application as a target application for the utilization of the proposed ideas. Thus, some of the considerations may not be valid for all kinds of tabletop applications.

1.3 Contributions

This work has two main contributions that distinguish it from the related works in the same field.

The first one is the proposal of a touch-sensitive tangible and novel interaction techniques employing the tangible in combination with a tabletop interface for 3D manipulation. There are similar researches that apply tangibles for performing 3D manipulations in tabletop environments, but few of them utilize tangibles that can be actually lifted up from the table, and none of the combine mid-air tangible manipulation with touch-sensitive capabilities.

The second contribution is the idea and implementation of a tangible tracking method that is based on processing the images of the active markers attached to the tangible, which are obtained from the same camera utilized for touch recognition. This method also allows the user to perform mid-air gestures with the tangible and also provides estimates of the tangible's distance from the table by only processing the images from the camera. There are several works in active marker tracking, but this is the first work to combine active marker tracking with tangible tabletop interfaces and also height estimation, utilizing just a camera as a data source.

1.4 Thesis outline

In Chapter II, some of the most relevant works related to our proposals are introduced and discussed. Then, in Chapter III, we describe all ideas related to our touch-sensitive tangible proposal, including details of the implementation for both interaction techniques and tracking framework. The main experiments conducted during the implementation of the proposals and the conclusions obtained from them can be found in Chapter IV. Next, Chapter V contains the discussions related to the proposals, including general issues, design decisions and facts observed during experiments. Finally, Chapter VI, Conclusion, brings a panorama of this whole dissertation and also the intended future work.

II

Related Work

There are several research fields related to the proposals in this dissertation. This chapter introduces some of these researches, along with brief descriptions, limitations and discussions.

The first research to be introduced is the one considered to be the first research in tangible user interfaces, which is Urp, by *Underkoffler and Ishii* (1999). It is the proposal of an urban planning application, in which tangibles are utilized to adjust manually some parameters, such as the position of some buildings, the position of the light source, wind direction and others.

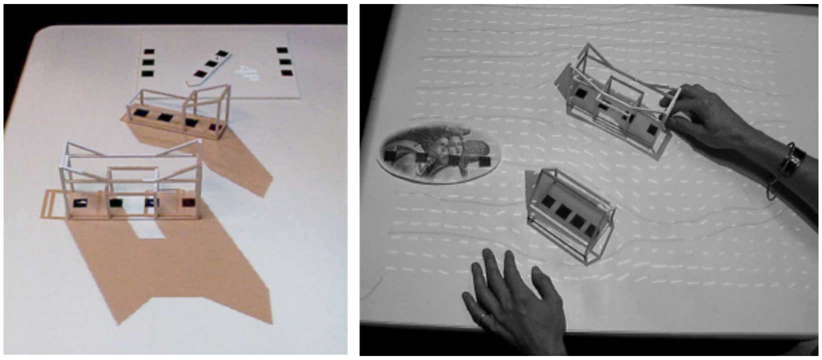


Figure 2.1: Urp (1999).

Then, talking especifically about related work in touch-sensitive tangibles, there are not many researches addressing this tangibles of this nature. One that can be cited here is Pulse, by *Gupta* (2010), which is a concept interface. This concept is based on the idea that a material responsive to proximity was created by scientists. So, Pulse is an interface that responds to the user's finger movements by raising a part of its surface creating buttons that can be pressed on demand. The material retracts once the user's fingers move away, going back to its initial form.

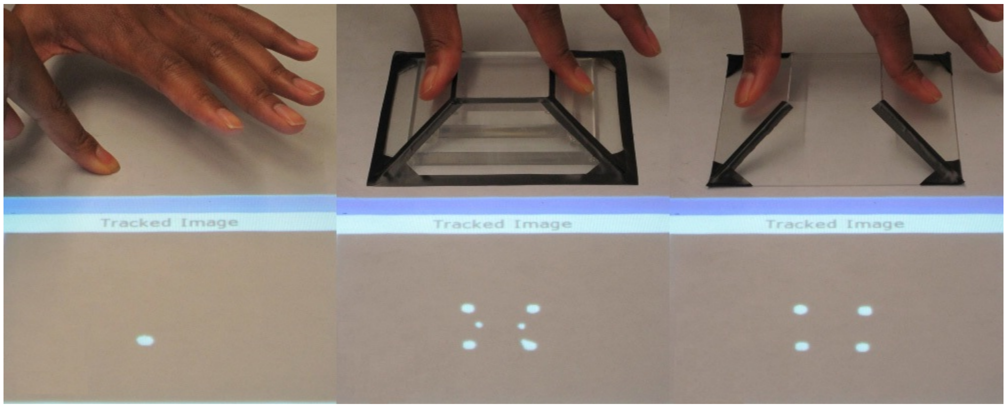


Figure 2.2: TZee (2011).

Another research about a tangible that can detect touch is TZee (*Williams et al.* (2011)). TZee is a transparent tangible that uses Diffused Illumination (DI) tabletops and light reflection to detect user's touches and gestures on its surface. Though, because it depends on the infrared light that comes from inside the table, its usage is restricted to DI tabletops. Also, if taken away from the table's surface, TZee cannot recognize userfs touches anymore, differently from the touch-sensitive tangible prototype built and described in the present work. TZee can be seen in Figure 2.2.

On the other hand, there are several researches that are related to the proposals in this dissertation in the fact that they try to address the 3D manipulation limitation problem in tabletops using many different tangible strategies. *Wu et al.* (2011) employed a tangible in combination with a tabletop to navigate a 3D world, which could be seen in an environment that consisted of two wall displays and a tabletop interface, as seen in Figure 2.3. The tabletop surface showed a top view of the whole world, while a first person view could be seen from the wall displays. The tangible's usually is utilized on or over the tabletop surface. Its position is determined by fiducial tracking when sitting on the tabletop's surface and by a camera mounted over the tabletop when lifted in the air – therefore, this method is susceptible to the occlusion that may be caused by the user's hands.

Another idea for 3D navigation using tangibles is the idea proposed by *Aguerreche et al.* (2010). They employed extensible tangibles built from camera tripods to perform 3D virtual object manipulation by associating them with the tangibles, as shown in Figure 2.4. Three optical markers are attached in each tangible's joints to allow position and orientation tracking, which is done by infrared cameras.

Related specifically to tabletop CAD applications, there is the research by *Au et al.* (2012). As opposed to the proposal in this dissertation, instead of employing

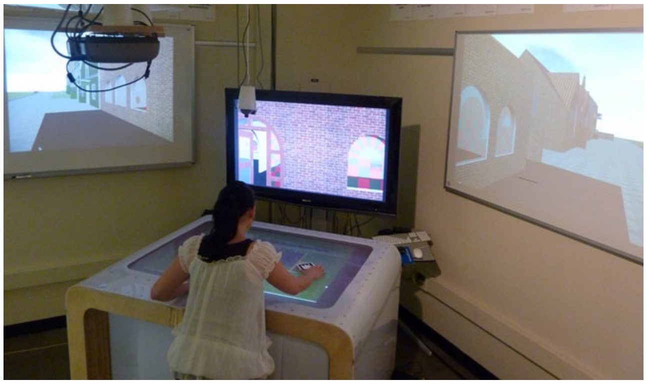


Figure 2.3: Tangible Navigation and Object Manipulation in Virtual Environments (2011).

tangibles for performing 3D manipulations, they proposed a set of gestures to be done on the surface of the table. Some of the gestures can be seen in Figure 2.5.

Finger Walking in Place (*Kim et al.* (2008)) is a proposal of a set of interaction techniques that achieve 3D navigation by finger gestures. They are based on the idea of walking simulation using fingers. The fingers' walking and steering motion are mapped to actions such as walking forward, backwards and rotating, which are restricted to a certain area on the tabletop. The gesture for walking forward is shown on Figure 2.6.

tBox (*Cohé et al.* (2011)) is another proposal of 3D manipulation for tabletops which does not employ tangibles. Though, the set of proposed gestures is simpler than the one proposed by *Au et al.* (2012). A widget which consists of a wireframe box is drawn over the object. The user interacts with it by doing simple gestures – such as sliding his/her finger over the edges of the box or over the face of the cube

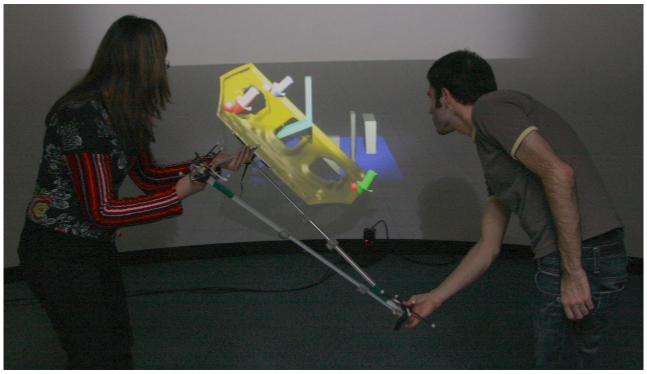


Figure 2.4: Reconfigurable Tangible Devices for 3D Virtual Object Manipulation by Single or Multiple Users (2010).

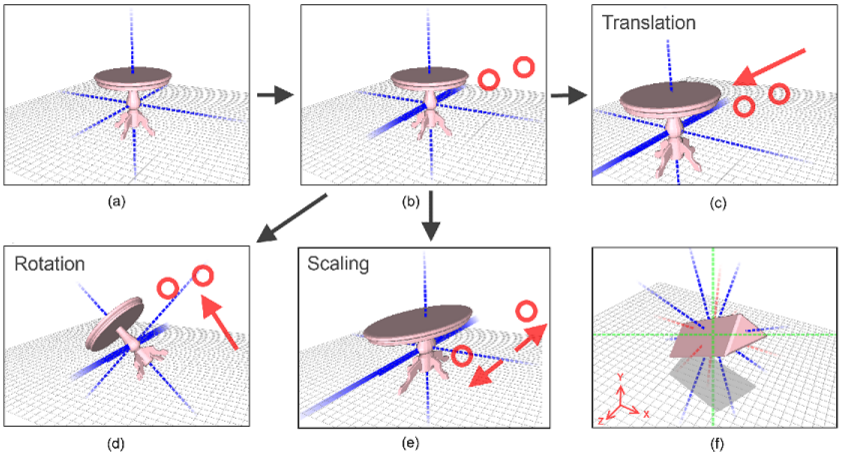


Figure 2.5: Multitouch Gestures for Constrained Transformation of 3D Objects (2012).

for rotations. An example of the gesture for rotations can be seen in Figure 2.7.

On the other hand, tracking tangibles is a hot topic in Tangible User Interface (TUI) researches. One of the most utilized tracking methods is to utilize computer vision to process images and detect fiducials on the table surface, which is the one employed by reacTIVision (*Kaltenbrunner and Bencina* (2007)). On the reacTIVision official site[1], both the source code and the binary for the tracking framework are available to download, as well as a set of asymetric fiducials that permit determining uniquely ID, position and orientation of a tangible. An extension to this method is the one proposed by *Nishino* (2010), which adds also angle information.

There are other tracking strategies, apart from the ones based on fiducials. Sensetable (*Patten et al.* (2001)) utilizes the electromagnetic field variation to do tracking of wireless tangibles. This method is robust against occlusion and lighting variation,

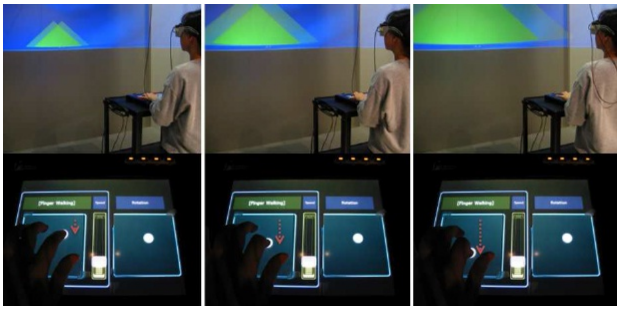


Figure 2.6: Finger Walking in Place (FWIP): A Traveling Technique in Virtual Environments (2008).

---

[1] Site: `http://reactivision.sourceforge.net/`

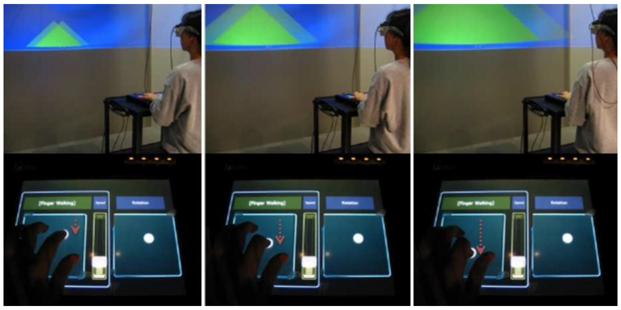


Figure 2.7: tBox: A 3D Transformation Widget designed for Touch-screens (2011).

since the electromagnetic field is not susceptible to such elements. Though, this method can be applied only to this kind of tabletop. Also, the setup of tabletop that supports electromagnetic sensing is very expensive and time consuming, because of its complexity. Sensetable is shown on Figure 2.8.

Another method that is unobtrosive is the one proposed and utilized in SurfaceFusion (*Olwal and Wilson* (2008)), that combines computer vision algorithms and RFID technology to sense the tangible, identify it and determine its position. Their tracking algorithm is quite robust due to the frame algebra that they utilize, which has very few assumptions about the tracked objects, making it applicable in a broad range of situations. SurfaceFusion can be seen running in Figure 2.9, and it is possible to notice a few tangibles being recognized on the table surface. The information about each tangible – or in this case associated to each RFID – must be previously registered into the system.

Regarding researches about active marker tracking, there are InfrActables (*Ganser et al.* (2006)) and QualiTrack (*Hofer et al.* (2009)). Both utilize active markers capable of transmitting bit-coded information that are attached to the tangibles.

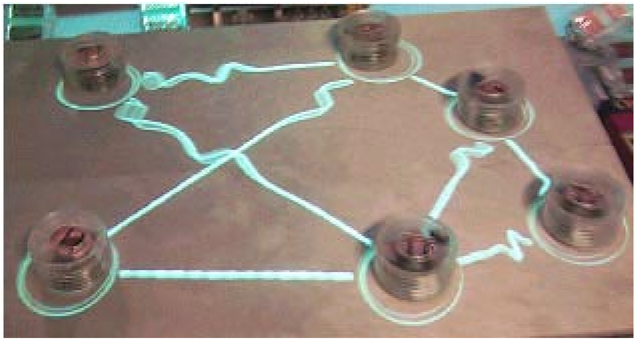


Figure 2.8: Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces (2011).

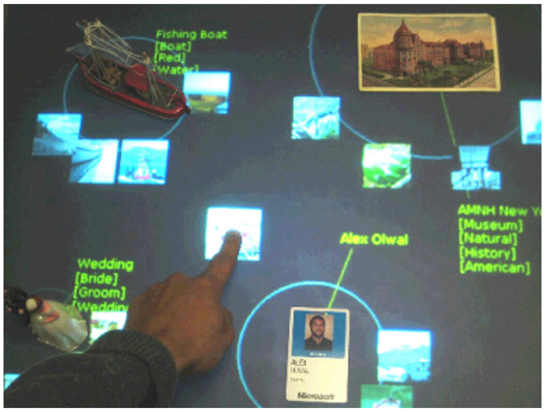


Figure 2.9: SurfaceFusion: Unobtrusive Tracking of Everyday Objects in Tangible User Interfaces (2008).

This way, the tracking system is able to detect position, orientation and state of each tangible. However, these systems require high frame rate cameras, which are expensive, and also the addition of a dedicated hardware to synchronize the camera and tangibles.

Finally, as a related work that also justify the methods introduced in this dissertation, the use of direct touch versus tangible interfaces for 2D and 3D manipulation in tabletops was evaluated by *Hancock et al.* (2009). Dummy data for the experiments were organized in two menus: one that was navigated by using direct touch on the TI and the other which was navigated through by utilizing a special tangible. After analyzing the results obtained through the user evaluations, they suggested that for coarse interaction methods, direct touch is better than using tangibles, but for precise manipulation, tangibles are better suited for performing the task.

In the following chapter, two proposals – one for making interactions methods in tabletop CAD applications more intuitive and one to track tangibles utilizing the already existing hardware – will be introduced and discussed.

III

Proposed Method

3.1 Introduction

In current Tangible Tabletop Interface (TTI)s, 6 Degree of Freedom (DOF) manipulation is a task that still cannot be accomplished by using intuitive interaction methods. As mentioned in chapter II, most researches either restrict the manipulation to less DOF or define a set of finger gestures to be done on-screen. When it comes to complex applications such as a CAD, restricting the manipulation is not acceptable, and requiring the user to memorize a set of gestures decreases the intuitiveness and learnability of the application.

In this sense, with the final objective of making 3D manipulation tasks more intuitive having in mind a tabletop CAD application, this research brings about two contributions: a proposal of a more intuitive way of performing manipulation tasks on tabletop interfaces utilizing a touch-sensitive tangible, along with a proposal of a new tracking method for tangibles employing active markers and computer vision algorithms. The first proposal depends on the second one, as the proposed tracking method is utilized to determine position and height of the touch-sensitive tangible during interaction. Both proposals will be introduced and discussed in this chapter.

In the following sections, the first proposal of this dissertation will be detailed.

3.2 Touch-sensitive tangible for 6 degree of freedom manipulation on TIs

For achieving intuitive 6 DOF virtual object manipulation in TIs, an interface which employs a touch-sensitive tangible and specific interaction techniques in combination with the table is proposed. The overall idea of this interface is that the user transfers the control of an on-screen virtual object which s/he wants to manipulate to a tangible by touching both at the same time, and then moving the tangible in the air, as if s/he was manipulating the object in the real world. The tangible can be considered in this case, then, an avatar of the virtual object.

The design of the interface and its requirements are discussed below.

3.2.1 Design requirements

For accomplishing or 6 DOF manipulation in TIs, the surface of the table itself imposes a barrier for direct on-screen gestures. Therefore, it is only possible to perform manipulations with up to 3 degrees of freedom: considering a Cartesian plane with its origin on the left-most bottom corner of the table, positive oriented, it would be possible to perform translations on the X and Y-axis and rotations on the Z-axis.

To overcome that physical constraint, two solutions are possible: to implement on-screen gestures or to not perform manipulations on the surface of the table. Because of the negative impact on the usability and learnability of the target application, it was preferable to find a way to perform manipulations away from the table surface.

When considering interactions off the TI's surface – in other words, mid-air interactions –, it is possible to implement recognition for such interactions by detecting the user's hand movements. But, again, it would require the user to learn how the interface works and what kind of hand/arm gestures would produce the desired result. Actually, the problem of this kind of implementation is exactly the same as adopting finger gestures on the table surface. The only thing that would change would be the place where the interaction actually occurs: in the air or on the screen. Though, one could argue that if Kinect™(*Microsoft Kinect 2013*) is utilized for capturing the user's hand and arm movements and intuitive gestures are adopted – such as moving one hand in the air for translations or turning his hand for rotations – the user would not have problems with learning the interface. That is correct, but, another problem would arise, and it is that the user cannot actually touch or feel the target of his/her manipulations, would make the interaction awkward. Thus, bare hand mid-air interactions may not be the most appropriate way of implementing 3D interaction in tabletops.

To improve the experience of bare hand mid-air interactions, it would be interesting to add the affordance factor to the experience – in other words, offer the user tactile feedback, which only a real world object could offer, and, at the same time, the capacity that a hand-sized object has to be held in one hand and touched, besides allowing being moved around in the air. For that, one possible option is to add a tangible for the user to hold and track its position and movement. Then, the user would move the tangible as an avatar of the virtual object which s/he wants to manipulate, and that can be done without focusing on the interaction, but on the application shown on the interface instead. The tangible solves the problem of the user's focus and, at the same time, due to its affordance, provides tactile feedback, even though limited. We believe that the property of affordance provided by the tangible during interaction improves the 3D manipulation experience because the user feels that s/he is actually holding and manipulating something, even though the target of the manipulation is a virtual object. It is known, though, that adding a device limits the flexibility of the interface, since the number of existing devices limits the number of simultaneous users that can utilize the interface.

Because of the nature of this method, which includes mid-air manipulation, some guidelines for the interface design were decided before actually building the necessary devices, to support future decisions:

- *Tangible size:* since the user is going to manipulate the tangible in mid-air, it should be small enough to allow holding with only one hand without causing any discomfort.

- *Tangible weight:* for the same reason stated above, the tangible should be of low weight so it does not tire the user's arms.

- *Accuracy:* the interface should allow users to manipulate accurately 3D objects on tabletops.

- *Cost:* the implementation cost should be low.

These guidelines also will be used to evaluate the method when it reaches its final stage, to measure how much of our initial goal was actually achieved.

Then, the discussion regarding the type of tangible chosen for the proposed interface – a touch-sensitive tangible – can be found in the next section.

3.2.1.1 Touch-sensitive Tangibles

Touching is one of the primordial ways that humans have to interact with the environment. Thus, it is considered a very intuitive way of interaction – such as that, when discussing the advantages of tabletop interfaces when compared to other interfaces, the possibility of interacting directly by touching it is promptly mentioned as an advantage. Surprisingly, though, there are very few researches involving touch-sensitive tangibles – therefore, the whole potential of this kind of interaction is yet to be explored.

Touching a tangible is an event that the majority of TTIs cannot detect. Detecting this event could allow special processing such as the tangible reacting to the users touch in some way as if it were alive and also it could enable finger gestures on the surface of the tangible, what could lead to the development of new interaction techniques. With that in mind, we present below some ideas of how touch-sensitive tangible could be utilized in tabletop applications. The examples are classified in terms of the effect of the touch interaction inside the tabletop application.

Activation and deactivation of tangibles. In a general application, one tangible could be deactivated in the interface by touching it. Another touch on it could make it active, toggling between these states. This would allow the user to simply inactivate tangibles temporarily without taking them out from the table surface.

Tangible selection. Touching a tangible on an application can be used for selecting it and then, afterwards, assigning it a special task. Selecting can be used for grouping tangibles as well the after a create group signal, each touched tangible would become a member of the group until the finish group input is done. Also, selecting a tangible by touch can be used to display information related that specific tangible on the tabletop surface.

Changing the function of a tangible. In an application where a tangible has two or more different exclusive functions, it would allow the user change the current function of the touched tangible. In a drawing application, a tangible that represents ink color can be touched in order to change its color or, if it represents a round-tip pen for drawing on the screen, touching it could change it to an air spray, for example.

Annotation tangible. A tangible could allow annotations in special visualizations of the tabletop contents that can only be seen by the tangible. As an example, *Koike et al.* (2009) proposed a new type of tangible which can also be used as a magnifier over a satellite image map. Adding touch recognition to this tangible, it would be possible to add hand-drawn annotations to a certain zoomed area. In another example application, a weather map with satellite photos could be displayed on the table and the tangible would allow visualizing the street map associated for a specific region. Then, upon identifying areas that would be hit by heavy rain, it would be possible to check which streets could be affected by flood or mudslides and annotate them. With this, it would be possible to quickly identify potential heavy traffic or danger spots and warn the population beforehand.

Visual feedback conditional on touch. If a tangible offered visual feedback to the user - for example, if it had touch-screen LCDs -, the tangible could display different contents conditionally to the position of the touch over its surface. For example, if an active tangible with a touch-screen LCD displays a characters face, a touch on one of the eyes would make the tangible react to that interaction by moving away from the user, as if it really had its eye poked and tried avoiding being poked again. This would make the tangible a special character and this kind of interaction with physical response can be applied to storytelling tabletop applications. This may get the childrens attention and involve them even more in the application.

User identification by touch. With some extra hardware in the tabletop, it could be possible to identify users who touched tangibles. With that information, the application may display contents related to the user who touched the tangible or even associate one user to one tangible, such in, for example, a tangible chess game. This way, it would be easy to detect who was the last user that played and also to identify illegal move, such a player trying to move the opponents pieces. The idea of user identification by touch is similar to the one proposed in by *Dietz and Leigh* (2001).

These are examples of how to use touch detection on tangibles in a tabletop application. However, the utilization of this feature is not restricted to just these applications. Additional applications of touch-sensitive tangibles have yet to be found, as this is still an under-explored field. In this sense, one of the expected contributions of the present work is to also incentive future works in touch-sensitive tangibles.

3.2.2 Interaction Techniques

The user is expected to interact with the tabletop application by concurrently using touch on the tabletop screen and manipulation/touch on the touch-sensitive tangible.

The recommended method for holding the tangible is as it is shown in Figure 3.1. The thumb should be used for gestures and touches. The user may hold the tangible with either hand.

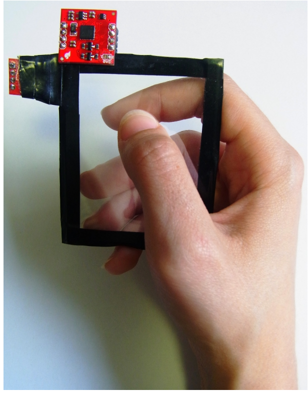


Figure 3.1: Recommended method for holding the touch-sensitive tangible.

In our application, the user will be able to translate objects along the X and Y axes, and perform rotations about the Z axis using only the tabletop. The first prototype of the tangible allows X-, Y- and Z-axis translations and X- and Y-axis rotations. Z-axis rotations are not performed in this first prototype because of the nature of the accelerometer.

The main idea of the interaction techniques developed for manipulating 3D models with the touch-sensitive tangible is to transfer the control of an on-screen component to the tangible itself. This is achieved by touching a specific component on the tabletop's surface and on the prototype at the same time. This procedure is illustrated in Figure 3.3.

After control is transferred to the touch-sensitive tangible, the following six actions can be realized:

3.2.2.1 Translation of a selected component

To translate the object along the X-, Y- or Z-axis, the user has to touch the tangible, hold the touch and move the tangible in the desired direction. To end the translation, the user just has to stop touching the tangible.

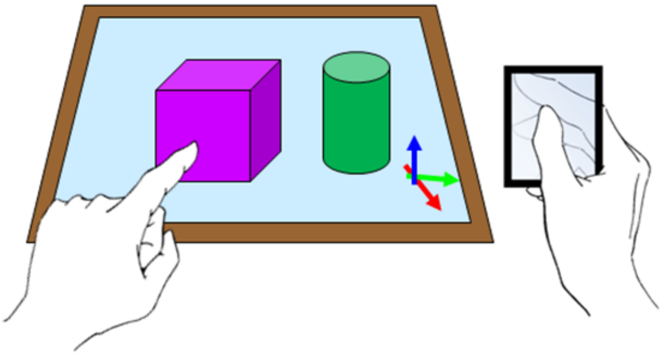


Figure 3.2: Transferring control of a 3D model from the tabletop to the tangible by touching them simultaneously.

3.2.2.2 Rotation of a selected component

To perform a rotation about the X-axis, the user makes a horizontal scroll gesture on the tangible's surface. For the Y-axis, a vertical scroll gesture is used. This action is illustrated in Figure 3.3. Alternatively, if tilting is supported in a future version of the prototype, the user can perform rotations by touching the tangible, holding the touch and then rotating the tangible about the desired axis. The reason for implementing alternative methods for rotating elements is to investigate which method the user feels is more useful and more comfortable.

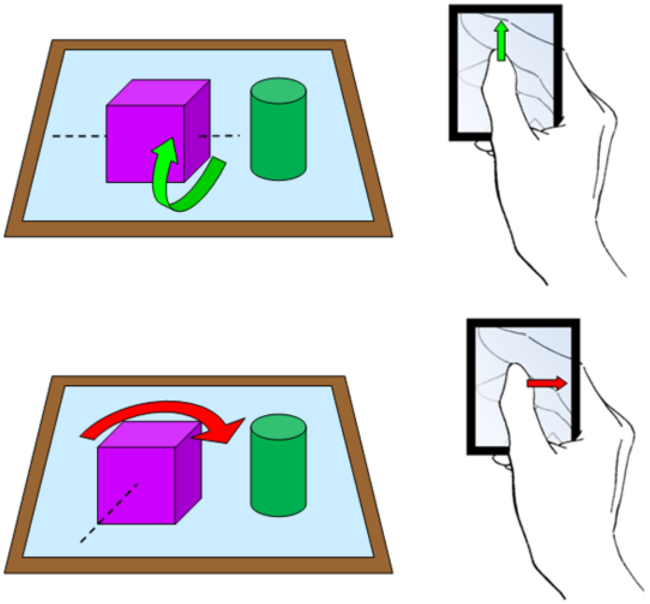


Figure 3.3: Rotations are performed by a scrolling gesture on the tangibles surface. Top: rotation about the X-axis. Bottom: rotation about the Y-axis.

3.2.2.3 Translation and rotation of the world-coordinate axes (changing the point of view)

The user touches the world-coordinate axis symbol on the tabletop surface, transfers its control to the tangible, and performs translation or rotation as for any selected 3D model.

3.2.2.4 Extrusion

The user can also create 3D models by extrusion. A finger is used to draw a closed line on the tabletop and to select it (by transferring its control to the tangible). Then, s/he moves the tangible up and down while touching its surface. Upon reaching the desired size, the user simply releases the touch on the tangible.

3.2.2.5 Scaling

To increase or decrease the size of a 3D model maintaining its proportion, the user transfers the control of the target object normally, as if s/he was going to translate or rotate it. Then, the user touches on the tabletop on the same object, this time holding the touch. Then, by moving the tangible up or down, s/he changes the size of the model.

3.2.2.6 Canceling the control of a component by the tangible

To cancel the manipulation of the currently selected 3D model by the tangible, the user can undo a selection by touching another model or by touching twice on an empty area on the tabletop. This action also implies that a touch on the tangible's surface means keeping the previously selected model active. Alternatively, s/he simply has to wait 10 seconds without touching the tangible.

3.2.3 CAD Tabletop Application

In this section, a CAD tabletop application will be described as well as how the interaction techniques here proposed can be utilized.

CAD applications, sometimes also referred to as CADs, are applications that are used for 3D modeling. They are extremely employed by design professionals – automobile designers, architects etc – and also for entertainment purposes.

A simple CAD application, such as Sketchup (*Trimble Sketchup 2013*), supports a set of basic operations, listed in Table 3.1. For this proposal of a CAD tabletop application, it would be desirable to implement initially these basic functions and, once they are implemented in a usable way, proceed with the implementation of advanced features, such as the ones found in Blender (*Blender 2013*).

The tabletop CAD application employs the tangible in interaction techniques that require 3D movement inputs. So, for the following operations, the input would be solely done on the tabletop, by touch:

- Draw lines;
- Import / export a 3D model;
- Apply texture;
- Move text.

For all the other operations listed, the tangible can be utilized. An overview of how the following operations can be found below, but some of them were already detailed in section 3.2.2, Interaction Techniques.

Extrusion. A polygon is drawn on the screen and the tangible is used to extrude it to the desired height. Already mentioned in section 3.2.2.4.

Table 3.1: Basic operations of a CAD application.

Type	**Operation**
Object creation	Draw lines (and pre-defined forms such as rectangles, ellipses, arcs, closed polygons) Create an object by extrusion
Model file operations	Import a 3D model Export a 3D model
Model operations	Translate an object (in the X, Y, Z axes) Rotate an object (about the X, Y, Z axes) Scale Perform Boolean operations on solids (addition, subtraction)
Decoration operations	Apply texture Add text Translate text Rotate text
Modifying the point of view	Translate the world coordinate axes Rotate the world coordinate axes

Translation and Rotation. Basic 3D manipulation techniques, they are performed after transferring control of the target model to the tangible. Details can be found in section 3.2.2.1 and 3.2.2.2.

Scaling. As written in 3.2.2.5, to perform scaling, a user should, after transferring control to the tangible, hold the touch on the 3D model that s/he wants to rescale. By moving the tangible up or down, the size of the model would increase or decrease, respectively, maintaining its proportion.

Boolean operations. Once both objects are created, it is necessary to position them as desired to perform the operations correctly. This way, it is only necessary to perform normal translations and rotations, and then choose the desired Boolean operation to be applied to the models.

Translate / rotate text. The text is selected as a 3D objected would be, transferring its control to the tangible, and then performing translations and rotations as one would with a virtual model.

Translating / rotating the world coordinate axes. A more detailed explanation can be found in section 3.2.2.3.

Finally, an image of how the tabletop CAD application would actually look like during its utilization is shown on Figure 3.4.

The sections below describe in detail the implementation of this proposed method.

Figure 3.4: Tabletop CAD application combining the tabletop and the tangible.

3.2.4 Implementation

3.2.4.1 Hardware

The hardware necessary for implementing this proposal are a touch-sensitive tangible and a tabletop interface. For evaluation purposes, a prototype of a touch-sensitive tangible was built and the already existing University of Tokyo Interaction Tecnology Laboratory (ITL)'s tabletop interface was utilized. The following sections introduce the two pieces of hardware needed for this implementation.

Touch-sensitive tangible prototype Initially, the touch-sensitive prototype was built using only a 3-axis accelerometer breakout from SparkFun Electronics (*SparkFun 2013*)attached to a Nintendo DS$^{\mathrm{TM}}$(*Nintendo DS 2013*) touch screen (2.2″ x 2.75″). It can be seen in figure 3.5. Both were connected to an Arduino (*Arduino (2013)*) Mega board, which was connected to the desktop computer of the tabletop interface by the serial port.

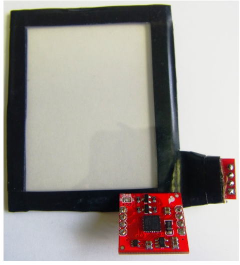


Figure 3.5: The touch-sensitive tangible prototype.

The X and Y coordinates from the touch screen and the X, Y and Z coordinates from the accelerometer are sent to the tabletop in the form of a character string with a baud rate of 115200 bps.

It was thought that only an accelerometer could achieve a satisfactory accuracy level for manipulation techniques, but, through a preliminary study (described in chapter IV), it was stated that the low accuracy of the prototype at that moment compromised the overall usability of the interface. As of now, the accelerometer is not integrated in the tracking system.

Then, to allow tracking by the tabletop camera, an active marker was attached to the back of the tangible. The marker is described in section 3.3.3, Active Marker, and the tracking system utilized is the one proposed in section 3.3.

Some modifications that were done is that the voltage source and ground for the marker were respectively connected to the 5V and GND pins that can be found in the

Arduino Mega shield on the cable connecting the tangible to the board. This way, the same cable can be shared for both power source and also data retrieval, eliminating the need for a new cable. Ideally, the tangible should be wireless, but, as of now, the current prototype is cabled.

Tabletop interface The tabletop interface utilized for implementing this proposal is the same as the one described in section 3.3.2.1. The only addition that is done to the tabletop computer is the connection of the touch-sensitive tangible prototype, via the serial port.

From this point below, the description of the software layer utilized on the implementation of the current proposal can be found.

3.2.4.2 Software

The 3D manipulation method here proposed requires the simultaneous tracking of the user's touches as well as the tangible's position. To achieve that and the implementation of the interaction techniques, several pieces of software had to be integrated. In this section, each piece will be described and how they communicate with each other will be explained.

Gestures The touch gestures supported for now are scrolling gestures. When touch is detected, the algorithm determines if the touch is being held or if it is the start of a gesture. The touch points are approximated by a straight line and its slope determines whether it was a horizontal or a vertical scroll.

Tangible tracking The tangible tracking utilizes the tracking method proposed in 3.3, which utilizes the images obtained by the tabletop camera and applies computer vision algorithms to detect position and estimate the tangible's height.

As a future work, the information from the accelerometer could be combined to the marker-based tracking proposal, to achieve better accuracy levels than utilizing either method alone. With the accelerometer, movement could also be detected by determining the peaks of acceleration using derivatives and the relative amplitude of each peak could be used as a multiplicative factor for changing the velocity of the translation/rotation.

This tracking method is the second contribution brought about by this dissertation, and it will be discussed further below.

3.3 Tangible tracking method for Tabletop Interfaces using Active Markers

The second contribution of this dissertation is the proposal of a method for tangible tracking by attaching an active marker on an already existing tangible and utilizing the tabletop camera. It is possible, with this method, to track both touches and tangible at the same time even though the camera employed for such task is the same.

To introduce this proposal, first the requirements for the utilization of this method will be listed, and then each part of the implementation will be detailed.

3.3.1 Design Requirements

The main goal of this proposal was to develop a tracking method for tabletop interfaces to track touches and tangibles using active markers. To achieve this aim, our system had to meet the following requirements.

- Tangible tracking must be performed using the tabletop's own infrared camera.
- The tangible must have an active marker attached and the geometrical characteristics of the marker should be known in order to calibrate the tracking algorithm.
- The application must be able to detect touches and the position of the tangible simultaneously.
- The system must be able to estimate the tangible's vertical height above the tabletop surface.

This method is based on processing images obtained using an infrared camera that is assumed to be below the surface of the table. Thus, the application is limited to tabletop interfaces using touch recognition techniques such as rear diffused illumination (DI), frustrated total internal reflection (FTIR), and diffused surface illumination (DSI).

It is necessary for the tangible to have an active marker attached and the Active Marker section (3.3.3) describes the construction of this type of marker. Our prototype is shown in Figures 3.10a and 3.10b.

3.3.2 Environment

This section introduces the environment in which the implementation was done. In section 3.3.2.1, the TI used will be discussed and the details necessary for understanding the implementation will be explained. Then, in section 3.3.2.2, the software required for development and their respective versions will be listed.

3.3.2.1 Tabletop interface

Figure 3.6: A picture of ITL's tabletop interface.

The tabletop interface utilized can be seen in Figure 3.6. It is comprised of a normal desktop computer connected to an infrared camera, which is placed in the center of a rectangular box-shaped wooden structure, without its cover. Four elongated infrared LEDs are placed around the camera, parallel to the sides of the box. This arrangement can be seen in picture 3.7. The wooden box is covered by a translucent light diffusing screen. This configuration is a typical implementation of a tabletop built for touch and tangible detection utilizing the rear DI technique, which schematic can be seen in Figure 3.8[1].

The rear DI technique is based on detecting changes in the infrared light diffusion pattern. The table surface is illuminated from below by setting infrared LEDs inside the tabletop structure. The cover of the tabletop – the surface which will be touched by the user – has to be either made of a light diffusing material or a transparent material with a diffuser glued to it (in the particular case of the tabletop utilized in this work, the first option). When there is no touch on the surface, the infrared light is completed diffused and the camera can capture only black images. Once a finger touches the diffusing cover, the light diffusing pattern changes and part of the previously diffused light is reflected back to the inside of the table structure, and that is captured by the infrared camera. Figure 3.9 shows a raw image from the camera obtained with this technique.

The raw images are processed by a touch recognition framework, which operation will be explained in the Software section.

---

[1] Image source: http://sethsandler.com/multitouch/reardi/

Figure 3.7: Inside view of the tabletop interface. The camera is positioned on the center of the bottom of the box and it is surrounded by infrared LED tubes.

The camera used for capturing the images to detect touch and also the tangible's position is a Point Grey Research infrared camera, model Firefly FMVU-03MTM. The settings used for optimal touch and tangible tracking can be found below on table 3.2.

Table 3.2: Camera settings

Video		**Image**	
Mode	0	Brightness	132
Width	576	Exposure	24
Height	360	Gamma	0
Packet size	2304	Pan	56
		Shutter	2.108ms
		Gain	0ms
		Frame rate	60.651 fps

The accuracy and continuity of touch recognition is very susceptible to the settings above, so they have to be carefully chosen and utilized. The process for choosing such values is completely experimental, based on the observed performance of the touch recognition framework.

A wide-angle lens is mounted on the camera to allow the interface to have a lower height, enabling kids to use the table and also that users can operate it while sitting. Because of that, the captions from the camera contain a distorted image of the surface and unnecessary regions (such as the sides of the wooden structure). So,

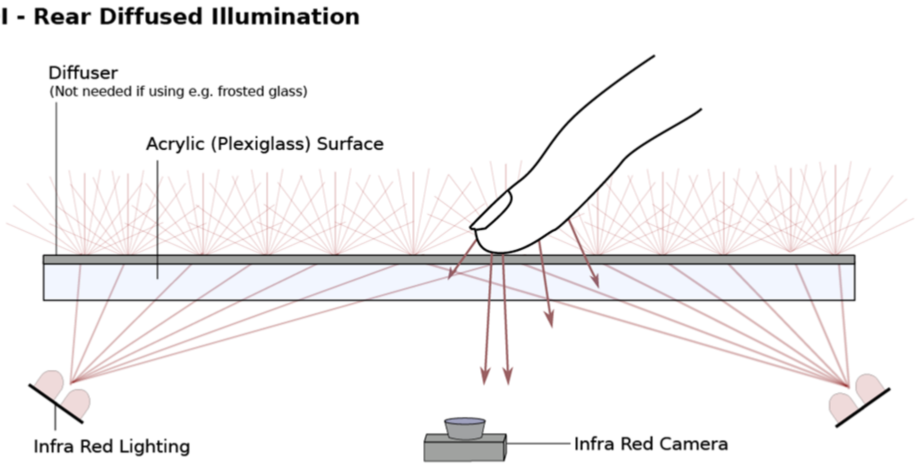


Figure 3.8: A scheme of a rear DI tabletop interface.

even though the maximum resolution captured by the camera is 752 x 480 pixels, after cutting the unnecessary information off the screen (such as the sides of the wooden box), the final resolution becomes 576 x 360 pixels – a few pixels from the corners are left in order to keep the captured images' aspect ratio as 16:10, to match the tabletop's projector. Finally, the camera is connected via a common USB 2.1 port to the tabletop computer.

The projector utilized is a Hitachi projector, model CP-AW100N. The settings utilized are as seen on table 3.3.

Again, these settings are very important, especially for the tracking algorithm, which calibration depends heavily on the camera and projector parameters.

3.3.2.2 Software environment

The development was done utilizing Microsoft Visual Studio 2010 Service Pack 1, running on a machine with the Windows 7 operating system. For processing

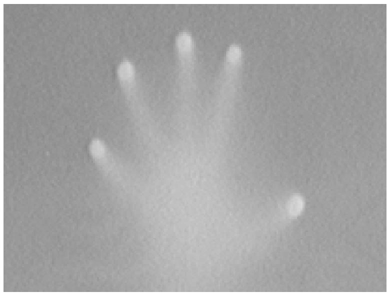


Figure 3.9: A touch as seen from the tabletop camera when the rear DI technique is utilized.

Table 3.3: Projector settings

Settings	
Digital zoom	276
Digital shift (vertical)	-27
Digital shift (horizontal)	+2
Image positioning	Center
Keystone	+29


(a) Front view

(b) Rear view

Figure 3.10: Pictures of the current tangible's marker.

the images captured by the camera, the OpenCV (*OpenCV 2013*) computer vision framework version 2.4.3 was utilized, in combination with the blob extraction library cvBlobsLib (*cvBlobsLib 2013*), both open source, and also the FlyCapture^{TM}SDK (*PGR FlyCapture 2013*), by Point Grey Research, necessary for operating the camera. The programming language utilized for the implementation is C++.

From the next section on, the implementation of this tangible tracking proposal will be explained. For implementing this method, three things are necessary: an active marker (whose building process is discussed on 3.3.3) and the tracking framework (introduced in 3.3.4).

3.3.3 Active marker

The active marker utilized for this implementation comprises four infrared LEDs placed on the vertices of a 2.2″ x 2.75″ rectangle that matches the size of the frame of the touch-screen that will be utilized as part of the prototype of a touch-sensitive tangible, the second proposal in this chapter (section 3.2). The four LEDs utilized in the prototype were model TLN231[2], by Toshiba. They were separated in two circuits of two LEDs each, and each pair was put in series with two resistances of 51Ω, resulting in a current of approximately 21.5 mA flowing through each circuit. A schematic of the arrangement can seen below, in figure 3.11:

For these two circuits, the voltage source and ground were respectively connected to the 5V and GND pins of an Arduino Mega shield just for simplicity, instead of utilizing voltage converters. Afterwards, voltage converters will not be needed anyway, as the marker will be attached to the back of the touch-sensitive tangible prototype described in section 3.2.4.1 and the power pins will be shared between them.

The LEDs employed, TLN231, are high-intensity LEDs made for space-optical transmission. So, when analyzing the image captured from the tabletop camera especially when the tangible prototype was in mid-air, the blobs generated by the LEDs would all overlap in a way that was not possible to identify their geometry, not even for a human. That happens because LEDs emit light in all directions – so, to limit light emission to the direction perpendicular to the tabletop surface and make blob recognition possible, each LED was wrapped in insulating tape.

For the first version of the marker for the prototype, the four LEDs were mounted on a Styrofoam frame with the same size as the touch-screen. The second version of the marker is much more compact, being made of small breadboards tiles on which the LEDs are soldered on, and the tiles are glued to the back of the tangible by using fusible glue. By practicality, the Styrofoam marker was utilized in all the experimental evaluations, since it has the same geometrical properties as the compact version.

One thing that was not considered in these versions of the prototype was their

---

[2] Datasheet: `http://www.semicon.toshiba.co.jp/eng/product_detail/opto/sensor/1263811_15032.html`

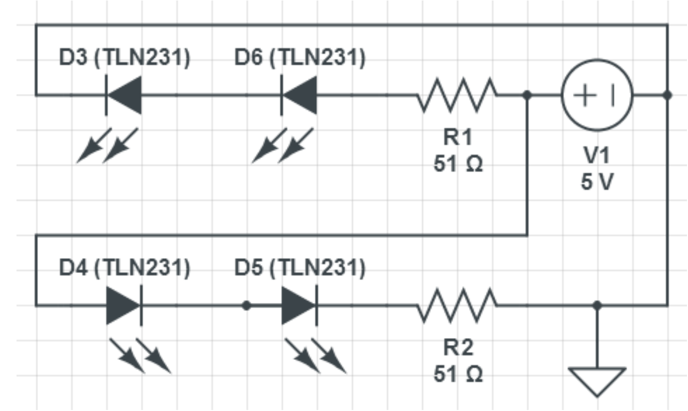


Figure 3.11: A scheme of the circuits for the marker on the back of the prototype.

symmetry. For real applications, to determine absolutely both position and orientation of the tangible which the marker will be attached to, it is necessary for the marker to be asymmetrical. Though, in these two versions, it was desired to create a proof of concept, thus, do a evaluation of the overall idea. For proof of concept purposes, the determination of the tangible's position is enough. If non-ambiguous orientation is necessary for the application, then the marker has to be redesign to an asymmetric shape and the tracking algorithm must also be changed.

Another item that has to be considered when designing a marker (especially an asymmetrical one), is the occlusion problem. Again, for proof of concept purposes this problem was not taken into account, but, for real applications, if the tangible is expected to be held by an user (such as the one proposed in section 3.2), chances are that one or more LEDs of the marker will not be detected by the camera. Then, the marker has to be asymmetric in such a way that the tracking algorithm is able to, with the visible LEDs it can track, reconstruct the original marker, guessing the position of the missing LEDs and determine orientation even in that situation. Of course, there is a limit for that capacity, but in a real application, the tracking algorithm has to be tolerant to a certain extent to cases of incomplete information acquisition like this. In the case of the implementation done for this work, orientation is not determined but the algorithm can do positioning even if only one LED is occluded. For two occluded LEDs, the algorithm does not work properly anymore.

In the next section, the tracking algorithm will be introduced and explained in detail.

3.3.4 Tracking framework

In this section, we describe the proposed touch and tangible framework. It contains three software layers. A scheme of their relation and how they communicate can be seen in Figure 3.12. Each layer is described in detail below.

3.3.4.1 Initialization Layer (IL)

First, the algorithm processes the initialization statements, which set up the camera and calculates the distortion matrix, which is used to compensate the distortion effect from the wide-angle lens.

After the setup is finished, a frame is retrieved from the camera.

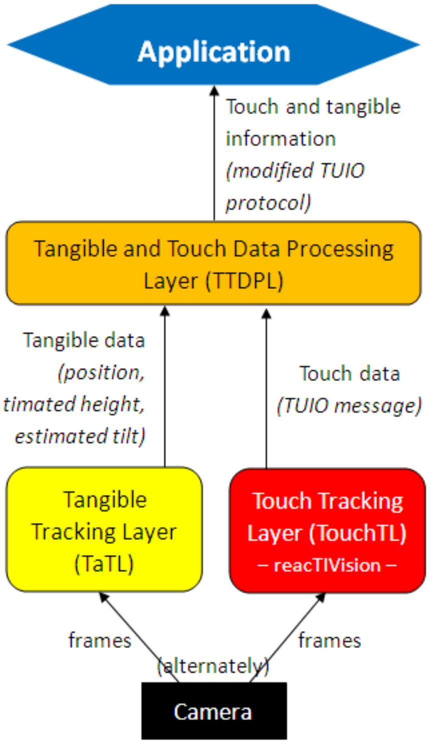


Figure 3.12: Software layers for the tracking framework proposal.

3.3.4.2 Tangible Tracking Layer (TaTL)

The tracking algorithm processes the image obtained from the tabletop camera and it estimates the position and height of the tangible relative to the table surface based on the calibration data and the known geometry of the tangible.

Each frame obtained from the camera is thresholded and the resulting image is processed to obtain information about blobs present in the picture. If the number of blobs detected matches the number of light-emitting diodes (LEDs) on the active markers (four in the current implementation), then it proceeds directly to the next step, i.e., lens distortion compensation.

If the number of blobs does not match the number of LEDs, it is assumed that the blobs are overlapping. An image containing only the blobs' contour is generated and the Hough circle transform is applied to obtain candidate points for the centers of the LEDs. To determine their validity, the color of the pixel in the position that corresponds to the candidate center in the original picture is checked – the pixel is in the blob region if it is white, whereas the pixel is a false positive if it is not.

After the centers of the LEDs have been determined, it is necessary to compensate for the camera lens distortion, which is achieved by accessing the corresponding distortion-compensated point in a distortion matrix. It is calculated based on the differences between the real world coordinates and those in the distorted image for 63 points, and approximated using Catmull–Rom splines to approximate the pixels between them.

Next, all of the points are fitted into a polygonal shape that matches that of the marker. The shape is compared with the original marker's shape and the distance from the surface of the table and tilt angle are obtained based on the patterns determined during calibration. This information is passed to the TTDPL, which is introduced below.

In the next sections, the main parts of the this layer – thresholding, blob detection and calibration – will be discussed and further explained.

Thresholding and Blob Detection Thresholding is a common procedure in image processing. It consists on, from a gray scale image, create a binary (black and white) image. For that, a threshold – hence the name – is defined, and pixels whose value in gray scale are above the threshold become white in the resulting image, while the others are become black. There is also the inverse thresholding, in which pixels whose values are below the threshold become white in the resulting image. Since every pixel in a gray scale image has only one color channel and the value ranges from 0 (pure black) to 255 (pure white), the threshold is also defined in that range.

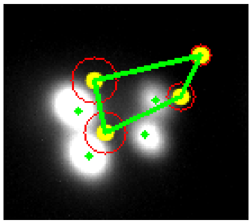
In this proposal's case, the gray scale image captured by the camera is first thresholded by a very high value, 190, and then processed by cvBlobsLib to detect blobs in the picture. In the case that less than the number of markers utilized (in this specific case, four) are detected, the original image is thresholded again with a lower value, 160.

Blob detection is done by cvBlobsLib in this tracking algorithm. It implements the algorithm proposed by *Chang et al.* (2004), taking a binary image as an input. As a result, it provides a list of blobs detected in the source picture, which are usually filtered by some criteria afterwards – in the case of this framework, to remove wrong results due to noise in the image, the blobs with total area less than 40 are discarded. Exactly because of this filtering that we realize a two-step thresholding. Figure 3.13 is an example of this. Figure 3.13a is a detail of the original image. After applying a threshold of value 190, the obtained picture is as seen in Figure 3.13b. As the marker in the upper right is discarded because of its small size (and that filtering cannot be removed due to noise in the image), thresholding is done again in the original picture,

(a) Detail of a frame captured by the camera.

(b) Frame after applying a value 190 thresholding (detail).

(c) Frame after applying a value 160 thresholding (detail).



(d) Center of the blobs after detection and distortion compensation (detail).

Figure 3.13: An example of the thresholding process.

but with a lower value. The detail of the result image is shown on Figure 3.13c. Now the upper right blob is overlapping the lower right blob – so the centers of the blobs cannot be determined by only using cvBlobsLib, but this situation is easily resolved by applying the Hough Circle Transform, as explained in 3.3.4.2. After the centers are obtained and the distortion compensation is applied, the obtained image is as in Figure 3.13d.

Overlapped blobs

If after the second thresholding all the markers have not been determined yet, then it is assumed that at least two blobs are overlapped. This happens quite often – when the tangible is resting on the table surface, the 4 markers are easily recognizable by only employing cvBlobsLib, but once it is lifted in the air, the circular image of each LED's light as seen from the camera becomes bigger, eventually overlapping each other. This happens following the LED's light emission model, called *light cone*.

Before actually solving the overlapping blob problem by software, there are two adjustments on the physical marker prototype that can be done in order to minimize that problem, or to produce overlapped blobs which are actually separable by image processing.

The first parameter that can be adjusted in order to produce blobs that can be separated in an easier way is the electrical current flowing in the markers. The LEDs utilized in this prototype were high-intensity LEDs, so for low currents their lights would be so strong that all the four blobs would overlap, being impossible to even a human to tell the LEDs apart. On the other hand, if the current is set too low, the maximum height from which the tangible is still trackable is minimized. So, in order to find a good equilibrium point for the trade-off that the current imposes on the detection of the markers, several experimental tests were done with different resistances to find an ideal value which allowed the tangible to still be tracked from a distance of around 20cm from the table surface, which is a height that allows the user to comfortably hold the tangible and manipulate it naturally, and, at the same time, did not emanate light bright enough to make more than two blobs overlap. For the Toshiba TLN231 utilized in this prototype, a current of approximately 21.5 mA flowing thorough each LED did fulfill these conditions. This value depends heavily on the model of the LEDs chosen for the marker, though.

The other physical parameter that can be adjusted in order to produce better images for the algorithm is to fix the LEDs on the marker prototype in such a way that they stay perpendicular to the table surface. This is justified again by the physical LED light emission model, which states that the LED light is more strongly emitted in the perpendicular direction to the plane where the LED is attached to. This may sound like an obvious observation, but when building the prototype it is very easy to misalign the LEDs with that direction, which affects the blobs in the image captured by the camera. In a situation where the marker is parallel to the table surface and all LEDs should produce blobs of the same size, the misaligned LEDs produce noticeably smaller blobs. When tilting or increasing the distance of the tangible from the table, the smaller blobs will become undetectable first than the others, which can produce wrong results in the tracking algorithm. So, it is necessary to be careful with this alignment when affixing the LEDs to the marker prototype.

Then, once these two physical adjustments have been done on the markers, most of the overlapping blobs obtained should be separable by software. For that, the Hough Circle Transform function is utilized. The idea of this algorithm and also its parameters, which are of extreme importance for the algorithm to succeed in obtaining the centers of the markers, are explained in the section below.

The Hough Circle Transform

The Hough Circle Transform is a generalization for the Hough transform proposed by *Duda and Hart* (1972).

In the Cartesian coordinate system, the equation of a circumference which radius R is known can be represented as

$$\begin{cases} x(a) = a + R\cos\theta & \text{for } 0 \leq \theta < 2\pi \\ y(b) = b + R\sin\theta \end{cases}$$

The center of the circumference is represented by $C(a, b)$ and it is what we want to determine. We have a set of (x_1, y_1) pairs from which we want to determine $C(a, b)$. So, rearranging the equations in terms of a and b:

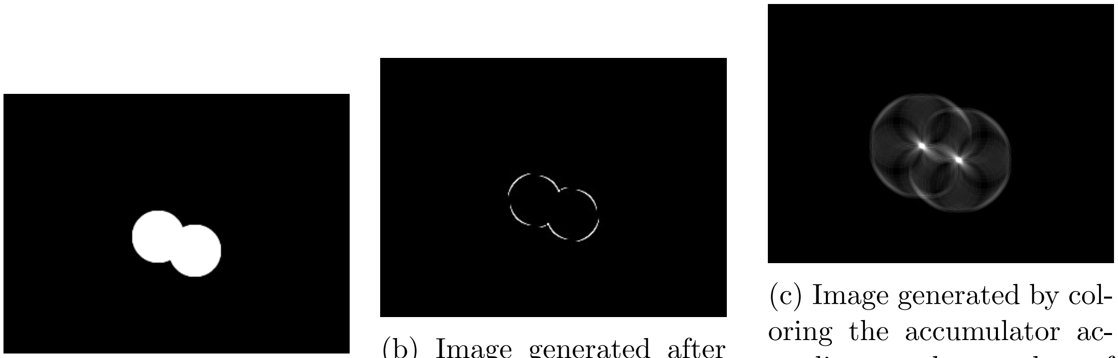
$$\begin{cases} a(x_1) = x_1 - R\cos\theta & \text{for } 0 \leq \theta < 2\pi \\ b(y_1) = y_1 - R\sin\theta \end{cases}$$

So, sweeping θ from 0 to 2π degrees, we have, for each (x_1, y_1), a circumference containing possible candidates for C. Therefore, the Hough Circle Transform consists on casting votes on candidates obtained for C for every (x_1, y_1) pair. The candidates are also called **accumulators**. The cells containing the greatest number of votes are considered the centers of the circles.

When the radius R of the circumference we are looking for is unknown (most common case), the algorithm defines three parameters: $minRadius$, $maxRadius$ and $minVotes$. $minRadius$ and $maxRadius$ define the range of the radius of the circumferences to be found and $minVotes$ defines the minimum of votes in an accumulator to be returned as a center of a circumference by the algorithm.

In other words, the step-by-step process is illustrated in Figure 3.14 and can be described as follows:

- Load an image, such as Figure 3.14a;
- Detect edges on that image and generate a binary image containing only its contour like Figure 3.14b;
- For $R = minRadius$ until $R = maxRadius$, cast votes in all accumulators which are at a distance R from every point in the contour;
- After sweeping the whole contour for all radiuses, the accumulators which



(a) Image containing blobs.

(b) Image generated after extracting the contour of the blobs.

(c) Image generated by coloring the accumulator according to the number of votes. The brighter the color is, the greater the number of votes.

Figure 3.14: Steps of the Hough Circle Transform.

$numVotes > minVotes$ are returned as the result of the algorithm. They are represented as the brightest spots in Figure 3.14c.

Once the centers of the LEDs are determined by the Hough Circle Transform, the TaTL fits the four markers into previously calibrated data and estimates its height from the table and also tilt angle. Finally, once this information is determined, the 4 markers' position, estimated height and tilt angle are sent to the TTDPL.

3.3.4.3 Touch Detection Layer (TouchTL)

In the proposed method, the touch detection task is handled by reacTIVision, which runs in parallel with the tangible tracking layer. The TTDPL listens for TUIO messages and processes them in the same way as a normal tabletop application.

3.3.4.4 Touch and Tangible Data Processing Layer (TTDPL)

This layer controls the infrared lighting of the table and also multiplexes the signals obtained from the TouchTL and the TaTL. Managing all of these data demands a time-division strategy for multiplexing data. Thus, time is divided into touch tracking time slots and tangible tracking time slots. During a touch tracking time slot, the program turns on the table infrared lighting and listens for TUIO messages. This time slot is followed by a tangible tracking time slot, during which the infrared lights are turned off on the table and the markers are switched on. This allows the tangible tracking algorithm to determine the marker's position without the user's fingers creating blobs in the images captured by the camera. For this process, a frame rate of about 25 frames per second (FPS) should be sufficient for touch/tangible tracking, taking into account that this tracking system will be utilized in a CAD application. Such applications do not require precise tracking of abrupt gestures, which permits utilizing a lower frame rate without affecting the usability.

This layer sends touch and tangible information to the application using a modified version of the TUIO protocol. Messages containing touch data only require an additional header so the application can identify the message as a touch information container. In addition to the header, messages containing tangible position information include the coordinates of the center of the marker, its estimated position, and its estimated tilt angle on each axis, which are also sent to the application. Thus, existing tabletop applications that utilize the TUIO protocol can readily be adapted to utilize our proposed tracking algorithm.

Video signal The video data has to be shared between the TouchTL and TaTL layers utilizing an application such as Syphon (*Syphon 2013*) or SplitCam (*SplitCamera 2013*), which allow sharing video between applications. This way, it is possible for both layers to access frames and process them to detect blobs. But, since the signal will be shared and not multiplexed, the TTDPL will be responsible to do the synchronism between both layers. Since the TTDPL will also be responsible for multiplexing the tabletop's IR lighting, the synchronism will be done by the layer itself, which will only listen and process the data received from each data according to the states of the tabletop IR LEDs.

Multiplexing the tabletop's infrared illumination The IR LEDs that illuminate the diffuser surface from below are necessary to identify touches on the table surface, but they affect negatively the detection of the tangible's infrared markers. In that sense, it is necessary to alternate the tabletop lighting between on and off states, and, according to that state, do either touch or tangible tracking.

The responsible for multiplexing the IR lights of the tabletop will be the TTDPL. The multiplexing will be done by an Arduino microcontroller connected to the computer and to the tabletop lights' power supply. The communication between the computer and Arduino will done by using the serial port. When instructed to turn on the tabletop's lights, Arduino will allow power to be supplied to the LEDs; Arduino will cut off the supply upon receiving messages to turn off the IR LEDs. Then, a time-division strategy will be implemented in the following way: in one time frame, the computer sends Arduino a message for turning on the lights, the TouchTL's messages are retrieved by the TTDPL and then it sends Arduino an instruction to turn off the table's lights. Then, in the next time frame, the messages from the TaTL are obtained and Arduino turns on the lights again, and so it goes, repeating the cycle.

The data received from each layer are rearranged in a modified TUIO protocol[3]: the messages have exactly the same format as the original, except by the fact that they have an extra field for every message, "type", to differentiate touch data from tangible data. This way, applications that already adopt the TUIO protocol can easily be adapted to utilize our proposed framework.

3.3.4.5 Application layer

The applications which employ the tracking system here need to utilize a modified TUIO client. It behaves exactly like a TUIO client, listening for messages on the port number 3333 and building the corresponding objects according to the information received, except by the fact that it is able to differenciate touch from tangible data and transmit that information to the target application.

3.4 Summary

This chapter introduced the two proposals for improving 6 DOF manipulation on tabletop systems: a tangible tracking technique based on computer vision algorithms to detect the active marker attached to the tangible, and the method for 3D manipulation utilizing a touch-sensitive tangible, including the prototype construction and interaction techniques.

---

[3] For comparison purposes, the latest TUIO protocol specification, version 2.0, can be accessed in `http://www.tuio.org/?tuio20`.

IV

Experimental Results

The experiments realized during the implementation of the touch-sensitive tangible and its tracking system can be categorized in three types: experiments for determining the best feature for estimating the height of the tangible, experiments for multiplexing the tabletop interface's IR light and a pilot study to evaluate some of the proposed interaction techniques proposed for the tangible. These three types will be explained below.

4.1 Determining the best feature for estimating height

Various experiments were performed in order to build the active marker tracking algorithm. From the results obtained, new experiments were realized, in a cyclic manner, with the objective of optimizing tangible tracking, and also determining the best features to be tracked in order to determine the tangible's position and height.

The tracking framework was created in a way that is possible to save the following information while it is running: the raw image from the camera, the absolute position of the center of each marker as determined by the algorithm, the measure for each side of the polygon determined by the four LEDs and the area of each blob. This information saving feature was added to allow data analysis on the images captured by the camera.

Since orientation is not being determined in this prototype, the convention adopted was that the vertex closest to the upper left side of the screen is vertex number 1, and the others are numbered in a clockwise order. Also, each side of the polygon determined by the four markers are named, respectively, `d12`, `d23`, `d34` and `d41`. The numbers correspond to the vertices that the edge connects.as seen in Figure 4.1.

To determine which feature from the camera capture would be best for determining

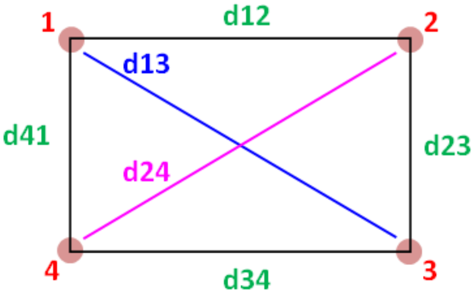


Figure 4.1: Naming convention for the vertices and sides of the image formed by the IR LEDs.

the height of the tangible from the table, data was collected using two lens that were available for use with the camera – a flat lens and a wide-angle lens.

Since the TI employed in the experiments has low height (total height of 60cm), the flat lens could capture only a part of the tabletop surface. Specifically, a 26cm x 19.5cm rectangle with its center aligned to the center of the tabletop surface. For the experiments using this lens, the resolution utilized for the video was 640 x 480 pixels. Therefore, the resolution in pixels/cm for experiments with the flat lens camera was of approximately 24.61 pixels/cm.

When the wide-angle lens was mounted on the camera, it was necessary to cut the borders of the video. The camera's maximum resolution is 752 x 480, and using that resolution the borders of the box were also unnecessarily captured. In order to obtain a cleaner image, the video borders were cut, lowering the resolution to 576 x 360 pixels. The video covered the whole tabletop area, which measures 87cm x 57cm. Thus, for experiments with the wide-angle lens, the resolution was of approximately 6.62 pixels/cm.

The captures were taken at the heights of 0cm (on the table), 5cm and 10cm (the maximum before the image of the markers would disappear) away from the tabletop. To keep the tangible marker stable at such heights and to not add unintentional tilting that a human holding the tangible during the experiments would cause, two supports made of cardboard boxes were prepared by cutting their bottom to fit the marker prototype and their sides were cut to match the desired distance of the prototype put on the support from the table was measured. One of them can be seen in Figure 4.2.

Then, the captures were analyzed to determine the best feature to track in order to obtain a good estimation of the height from the tabletop. The data obtained is presented below, separated by parameter analyzed.

(a) Box and breadboard with the marker attached on it during an experiment for height $h = 10cm$.

(b) Box and breadboard with the marker facing up.

Figure 4.2: Cardboard box to keep the prototype stable and parallel to the table surface during capture.

4.1.1 Length of each side

In this experiment, the length of each side of the tangible – d12, d23, d34 and d41 – was independently analyzed. For each height, 80 images were captured and processed. The graphs obtained can be seen below:

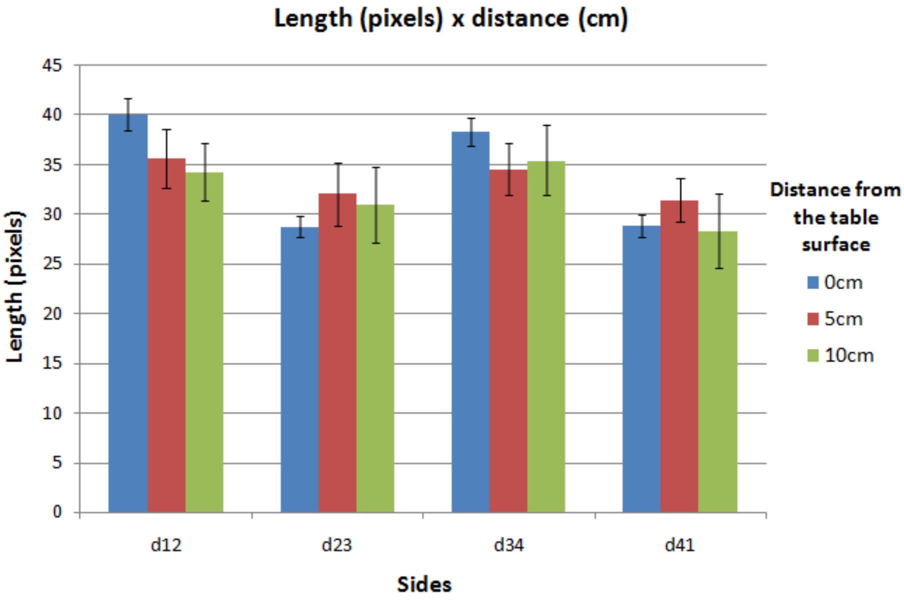


Figure 4.3: Graph: length (pixels) x distance (cm) for each side of the rectangle, obtained by using the wide-angle lens.

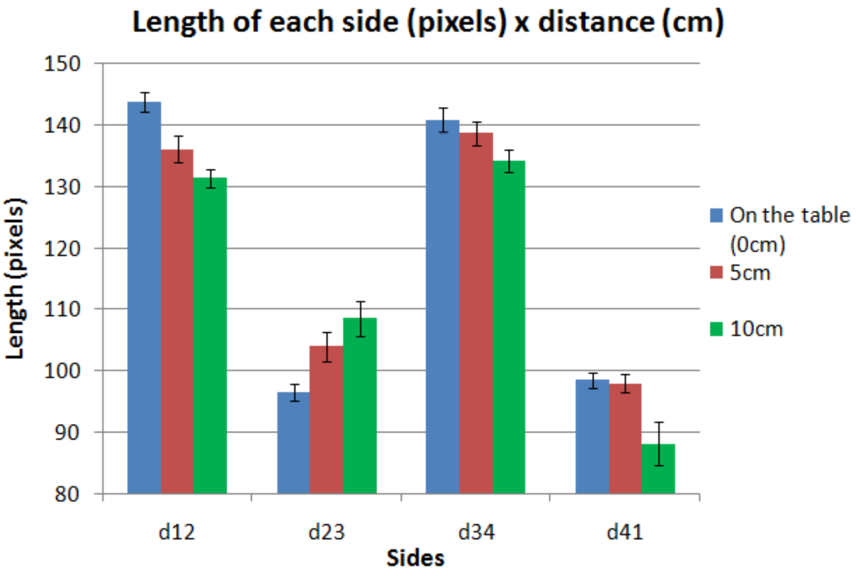


Figure 4.4: Graph: length (pixels) x distance (cm) for each side of the rectangle, obtained by using the flat lens.

4.1.2 Perimeter

In this experiment, the length of the four sides of the rectangle was summed up and the average value for each height was calculated. The calculations were based on 68 samples for each height. The graph with the results for the perimeter experiment can be seen in Figure 4.5.

4.1.3 Diagonals

The diagonals d13 and d24 of the rectangle were measured for each height in this experiment. They were compared individually and also averaged. The results

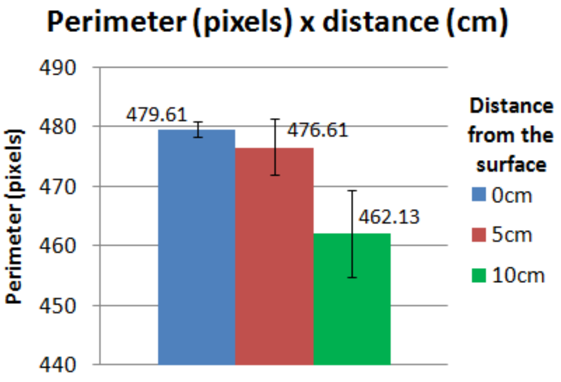


Figure 4.5: Graph: perimeter (pixels) x distance (cm) for the wide-angle lens.

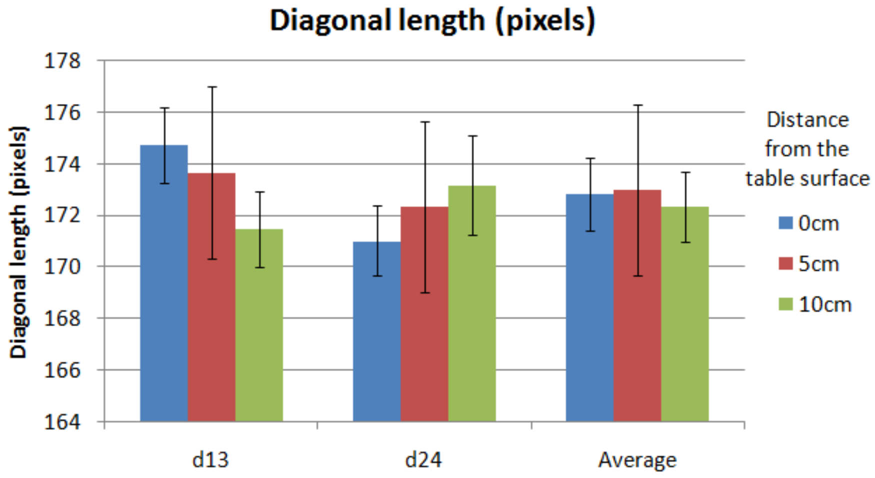


Figure 4.6: Graph: diagonal length (pixels) x side (d13, d24 or average) for the flat lens.

obtained can be seen in Figure 4.6. A total of 70 samples were obtained for this experiment.

Since at this point none of the linear parameters observed seemed to be satisfactory for tracking, we started analyzing the area of the marker's blobs in the camera captures.

4.1.4 Area for each blob

For this feature, the area for each of the four blobs was obtained utilizing cvBlobsLib after thresholding the image obtained from the camera. The calculations were based on 70 samples for each height. Figure 4.7 shows the results for this experiment.

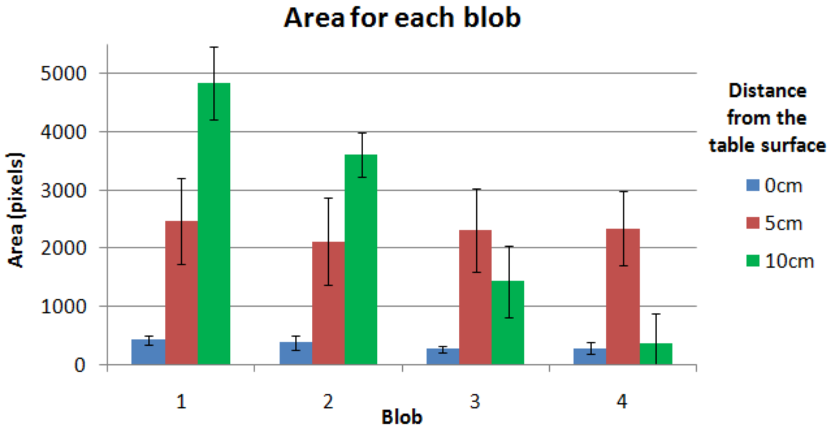


Figure 4.7: Graph: Area (pixels^2) x distance (cm) for the flat lens.

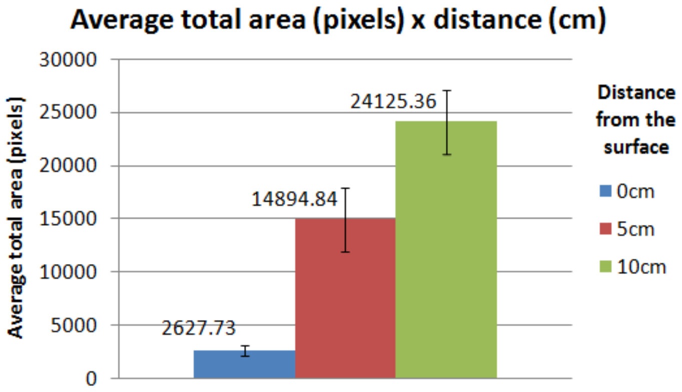


Figure 4.8: Graph: average total area of the blobs (pixels^2) x distance (cm) for the flat lens.

4.1.5 Total area of the blobs

For each sample, the area of the four blobs were obtained as in the previous experiment and added up, obtaining the total area for the blobs in such sample. Then, the average total area for each height was calculated. The graph is shown in Figure 4.8. Since the average total area of the blobs seemed to be a good parameter for tracking, we also did a linear fit for the obtained data, in order to obtain an approximate equation for calculating the height, as seen in Figure 4.9.

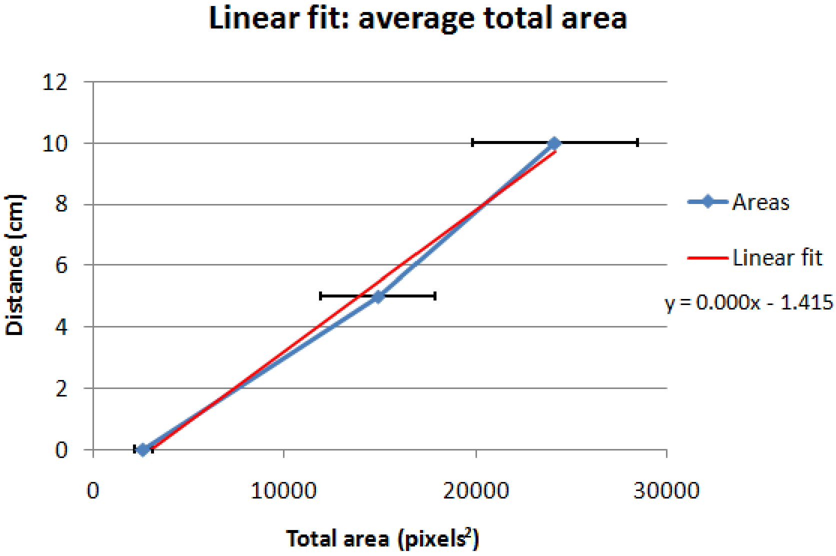


Figure 4.9: Linear fit of the data.

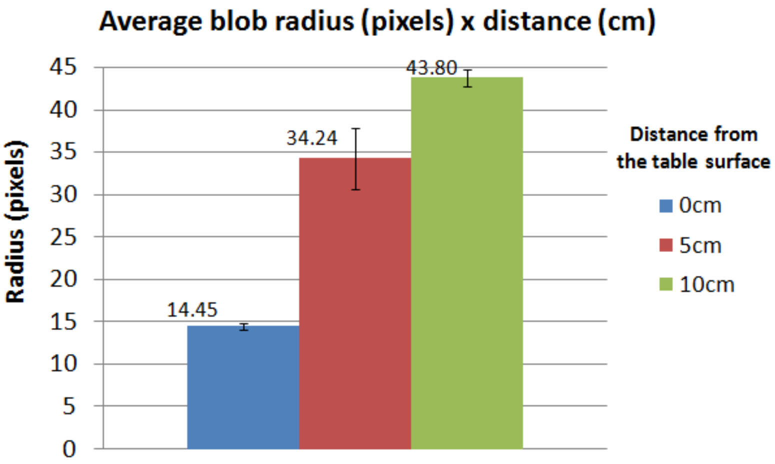


Figure 4.10: Comparative graph: average blob radius (pixels) x distance from the table for the flat lens.

4.1.6 Average blob radius

In this experiment, for each image, the total area for the blobs was calculated and, assuming that the four blobs are perfect circles of the same size, their radius is calculated with the formula $r = \frac{1}{2}\sqrt{\frac{A_T}{\pi}}$. The graph showing the results obtained is the one in Figure 4.10.

The radius proved to be a good feature for estimating the distance, we fitted the

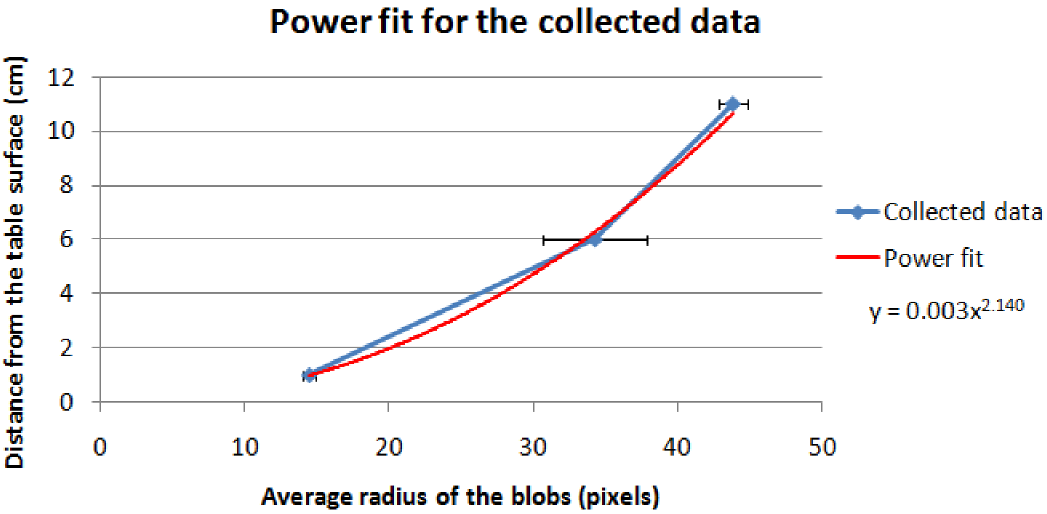


Figure 4.11: Power fitting of the data for the average blob radius for the flat lens. **Note:** Since it is not possible to fit the value for distance = 0cm, the graph was shifted 1 unit to the right. Thus, the resulting equation after fitting the data is is $y(x) = 0.003(x-1)^{2.140}$.

data to obtain an equation for height. Since radius is a function of the area, which, in turn, changes according to distance, the data obtained experimentally was fitted into a graph that can be seen below, in Figure 4.11.

4.1.7 Discussion about the experiments

Except for the total blob area and average blob radius, it is safe to affirm that it is not possible to determine the distance from the table surface utilizing the tested parameters. By observing the data obtained of the experiments, the standard deviation of the samples often overlap either the deviation for different heights or the average themselves, making it impossible to calculate the height with acceptable accuracy levels. The only two parameters that show an acceptable accuracy for such calculations are the total blob area and the average blob radius.

On the other hand, from these results, it was possible to conclude that the best parameter to be tracked in order to get a good estimation of the height of the tangible up to 10cm away from the table is the average blob radius. The average error for this parameter utilizing the equation obtained from data fitting is of about 18%, which means that it may be hard to determine a small variation of the height from the tabletop (from $d_1 = 3cm$ to $d_2 = 4cm$, for example). Though, for variations like 5cm or so, the accuracy of height determination also increases.

4.2 Multiplexing the Tabletop's Infrared Lights

The TI IR lights are powered by a power supply as seen in Figure 4.12a. It accepts a power input of 100V AC and its output is of 12V DC. Two ways of multiplexing the tabletop's lighting were considered: one was to control digitally the 12V DC output from the power supply and the other was to control digitally the 100V AC input of the power supply.

To avoid burning the IR lights, the experiment that was done was to control the 100V AC input of the power supply utilizing a relay. In a worst case scenario, the relay or the power supply would burn, and these can be more easily replaced.

Then, a microcontroller-controlled power plug was built, using a relay circuit board, a power plug, a power cable and an Arduino Mega 1280. The components can be seen in Figure 4.12b.

Then, the power supply was connected to the Arduino-controlled plug and the frequency of the output was changed according to the program written to the Arduino

(a) Power supply: 100VAC 50/60Hz input, 12VDC output.

(b) Arduino-controlled power plug.

Figure 4.12: The power supply for the tabletop's infrared lights and the Arduino-controlled power plug built for the experiment.

board. To verify the output, an oscilloscope was utilized, and its probe was connected to the power supply's output.

Two parameters were going to be evaluated with this experiment: the voltage level for the output (it had to be 12V to correctly power the IR LEDs) and the frequency – which is crucial for the touch and tangible information to not affect each other.

What it was possible to observe in the oscilloscope, though, was just noise, for an input at the frequency of 10Hz – it was not possible to control the tabletop's lighting by controlling the power source for the light's power supply. What was possible to observe in this experiment is that the capacitors inside the power supply would not allow the input to reach 12V nor would switch with the same frequency as expected.

Even though this experiment did not succeed, it does not mean that the tabletop's lighting cannot be controlled. If changing the frequency of the power supply is not a good idea, there is another point in the light's setup that can be switched, which is the power supply's output. The voltage at that point is 12VDC, which is a low voltage level and therefore easier to create switching circuitry. Though, since this part is directly connected into the infrared LED tubes, there is a possibility of damaging them. So, as future work, we plan on acquiring spare infrared LED tubes to proceed with the tabletop lighting multiplexing experiments, to finish the full implementation of our proposed ideas.

4.3 Pilot study – Touch-sensitive Tangible

A pilot study was conducted to investigate the validity of the idea of utilizing a touch-sensitive tangible for 3D manipulation and to evaluate the usefulness of some of the interaction techniques implemented utilizing the tangible.

The hardware utilized in this experiment were the touch-sensitive tangible prototype and the tabletop system. The prototype was comprised by only the touch-screen with an accelerometer attached to it, connected to the tabletop computer. The IR markers and the tracking system were not utilized, as this study did not utilize the information about the position of the tangible, only the information about its movements, captured by the accelerometer. So, we also wanted to investigate in this experiment, how well the interaction techniques would work by using only an accelerometer instead of the tracking system.

4.3.1 Description

The study was performed with 5 users, all men, aged 23-40, 4 of which were right-handed and one was left-handed. Each user was shown two tabletop applications for 3D manipulation - one that offered only a GUI and one that used the touch-sensitive tangible. All subjects were given as much time as desired for using the applications and verify the limitations of each interface. Afterwards, they answered a questionnaire, in which they should rate in a 0 to 10 scale their opinion on how easy each interface was to use, how well they worked for 3D manipulation and how satisfied they felt with the interface. The results can be seen in the Figure 4.13. The GUI application got an average of 3.4 points for ease of use, 3.2 for achieving is purpose and 3.0 for satisfaction with the interface. The application with the touch-sensitive tangible got 8.0, 7.2 and 8.4 points, respectively. Also, the users were also asked whether the idea of introducing a touch-sensitive tangible in a tabletop interface for 3D manipulation was valid, and if the tangible was useful for that purpose. All the participants answered yes.

4.3.2 Feedback and discussion

While the evaluation tests were taking place, it was noticed that even though we thought that handedness would affect interaction, all users felt more comfortable using the tangible with their right hand.

During the evaluation, we obtained very positive comments from the users, most regarding the fact that touch-sensitive tangible allows intuitive manipulation of the

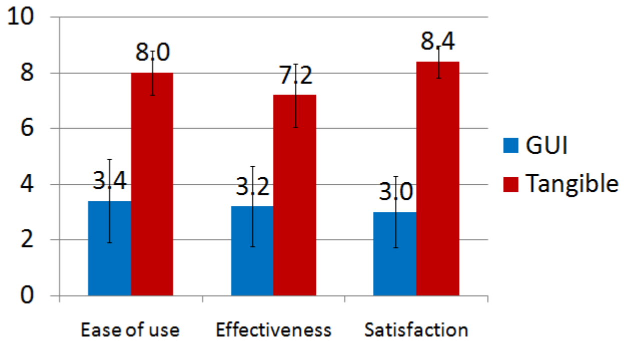


Figure 4.13: Results obtained from the user study. Y-axis shows the score obtained, while the X-axis shows the aspects of the interface evaluated.

models, without having to memorizing gestures, and the implemented gestures do not require exhausting arm movements, because the tangible did not require the user to move more than 5cm to do about a 50cm (on-screen measurement) translation. Also, some users commented that they could see that the tangible allows support for 3D gestures and that could be helpful for 3D manipulation. Finally, we also got comments on how this idea can also solve the problem of manipulating virtual objects in big screens, because using only direct touch can be tiring and require the user himself/herself to move along the table just for the sake of performing the task.

Regarding the negative feedback obtained about the system, the biggest problem pointed out was the accuracy of the system. The movements were detected only by processing the accelerometer's readings and not by utilizing the tracking system. Accelerometers are sensors that are naturally not very precise and their readings fluctuate even when put at rest and not touched at all, as seen in Figure 4.14. As a result, sometimes when a user moved the accelerometer to a certain direction, it would first respond in another direction and then translate in the desired way, or, in the middle of the translation, the object would stop, move a little bit in another direction, and then keep going following the previous trajectory. This is a problem that needs to be fixed in order to make the system actually usable, users said.

Another problem, less serious than the one stated before but still impacts usability, is the control-display ratio. Again due to the fluctuations in the accelerometer's readings, the peaks of accelerometer variation sometimes are different even though the user performed a movement with almost the same intensity. Then, because the amplitude of the peak serves as a multiplicative factor for calculating how much the object will be translated, if that peak does not have a fixed proportion rate with the intensity of the user's movement, then the control-display ratio is affected and the user

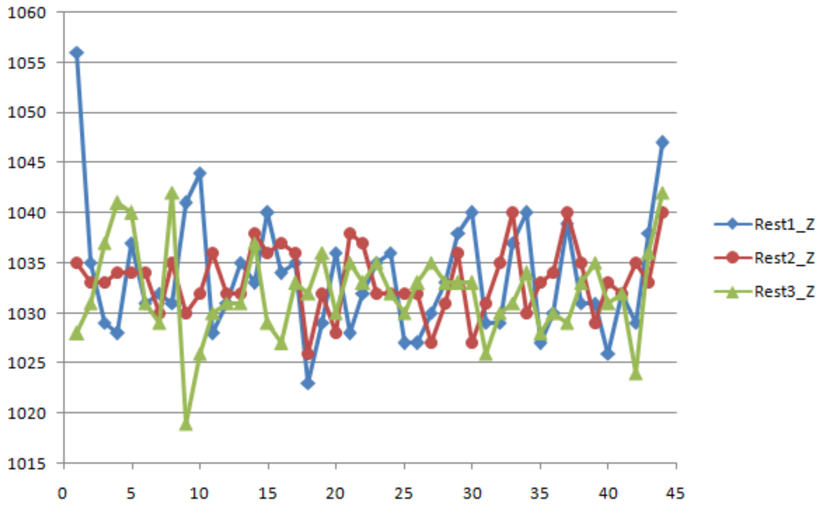


Figure 4.14: Graph showing the Z-axis accelerometer readings for the three experiments, with the sensor completely at rest. Y-axis of the graph shows the value of the reading after passing through the AD converter, and the X-axis shows the number assigned to each reading.

is not able to control with accuracy how much the objected will be translated every time that s/he moves the tangible by changing the intensity of her/his movements. This has to be corrected to improve the usability of the interface and obviously for making the 3D manipulation by using the tangible more intuitive.

Another usability issue pointed out by the users is that the current system requires the user to be aware of the position of object s/he wants to move. The system does not perform automatic axis transforms taking into account the user's point of view, besides mirroring the Y axis (because the user is always facing the table). So, if the initial state of the axes of the object to be manipulated are not parallel to the axes that represent the user's point of view, the user will have to look at the object and think of how to manipulate the tangible in order to perform manipulation. In other words, if the user performs a translation in the positive Y axis direction using the tangible, the object will do a translation in the negative Y axis independently of the user's current view of the object. This affects usability of the system negatively.

As for suggested improvements for the prototype, one was to implement the automatic transform of the tangibles input depending on the users view of the model so s/he does not have to be aware of the current views axis and its relative position to the tangible as explained above. Another suggestion was to implement tilt detection in the tangible, so it is possible to implement rotation guided by the applications current view, similar to the "trackball" rotation mode in Blender. Also, it was identified by one of the users the need of implementing gestures for scaling 3D objects.

In the next section, important facts observed with the experiments and during the implementation will be discussed, as well as general issues and limitations of the proposed methods.

V

Discussion

This chapter contains some issues related to the ideas proposed in this work, such as design decisions that require further discussion, interesting observations that deserve more attention, general ideas and future work.

5.1 Touch-sensitive tangible x Smartphone

One of the first questions that may arise upon reading/hearing about the work detailed in this dissertation is, why build a touch-sensitive tangible, which is a handcrafting process that is more time-consuming and which result itself is a very limited prototype, instead of using a smartphone?

Even though this prototype resembles a smartphone, we prefer this implementation because we want to develop interaction techniques that utilize the transparency of the tangible, such as transferring a pattern from a printed image to the application by drawing on the touch-screen. Even if the transparency can be emulated by displaying something on the smartphone, it is not possible to capture the video of things that are too close to the smartphone – as in, directly touching its back – because of the built-in camera's field of view. Therefore, the transparency effected emulated this way would be very artificial, and probably would cause strangeness in the user. It is better to display something else (like a trace in the touched area) than emulate transparency if a smartphone is used to replace the touch-sensitive tangible in our proposal.

Moreover, if the user knows that s/he's holding a device that can actually display something (or even emulate transparency), there is the possibility of the user unnecessarily focusing her/his attention on the screen to see if something actually is going to happen instead of focusing on the task.

5.2 Accelerometer and Tracking

From the pilot study, we could conclude that utilizing only the touch-sensitive tangible and the accelerometer would not produce a satisfactory accuracy. At the same time, we could observe during the implementation of the tracking framework, that tilting cannot be determined from the captures, requiring an additional device. This way, if it is desired to add tilt detection to the current tangible tracking framework, the accelerometer information must be utilized. In the case that tilting is not important, then the accelerometer is not necessary. Also, adding the acceleration information to the system is interesting not only because of tilting, but because combining the accelerometer information with the height detection from the tracking framework may increase the height determination accuracy, allowing the framework to be usable also for small height variations, such as 1cm or so.

5.3 Limitations of the Tangible Tracking Framework

The proposed tangible tracking method has a few limitations. As mentioned before, the current proposal does not work for tabletops that do not have in their physical structure a camera positioned under the surface of the tabletop and that utilize infrared light to do touch recognition.

Another limitation is that the calibration for estimating the tangible's height was highly dependent on the marker. Because of that, the implementation done only works with rectangular-shaped four-LED markers. If a marker of a different shape, size and/or with a different number of LEDs is utilized, the code for the implementation itself has to be changed. In future work, it would be interesting to create a user-friendly interface for registering markers to be tracked and also for doing height calibration. This way, users could customize the algorithm and utilize it in real applications.

Also regarding the marker, one thing that was not considered in this version of the prototype was its symmetry. In real applications, if it is a requirement to be able to determine the absolute position and orientation of the tangible to which the marker is attached, then the marker will need to be asymmetrical. Though, for proof of concept purposes, the determination of the tangible's position is enough. If non-ambiguous orientation is necessary for the application, then the marker has to be redesigned to an asymmetric shape and the tracking algorithm must also be changed.

Another item that has to be considered when designing a marker (especially an asymmetric one), is the occlusion problem. Again, for proof of concept purposes this problem was not taken into account, but, for real applications, if the tangible is expected to be held by an user, chances are that one or more LEDs of the marker will not be detected by the camera. Then, the marker has to be asymmetric in such a way that the tracking algorithm is able to, with the visible LEDs it can track, reconstruct the original marker, guessing the position of the missing LEDs and determine orientation even in that situation. Of course, there is a limit for that capacity, but in a real application, the tracking algorithm has to be tolerant to a certain extent to cases of incomplete information acquisition like this. In the case of the implementation done for this work, orientation is not determined but the algorithm can do positioning even if only one LED is occluded. For two occluded LEDs, the algorithm does not work properly anymore.

As mentioned in the experiment to determine the best feature to determine the tangible's height from the table, even though calculating the average blob radius based on the total area is the best option, small height variations cannot be detected easily. This way, the application has to adjust its control-display ratio in order to make up for that issue. For applications that require very precise height determination, the proposed tracking framework is not usable yet – it has to be improved in order to increase its accuracy. Though, as for the interaction techniques here proposed, the proposal is not to determine exactly the tangible's height, but instead its variation. The CAD application we have in mind utilizes the height variation to perform Z-axis translations. Since the translation is proportional to the height variation, it is possible to adjust the control-display ratio in the application for it to respond satisfactorily despite the current framework's low accuracy for small height variations.

Finally, as already mentioned before, the current marker prototype had a cable. In real applications, it would be better to have wireless markers because cables can restrict interactions and occlude parts of the markers, which could lead to incorrect tracking or no tracking.

5.4 Advantages of the Tangible Tracking Framework

Even though the proposed tangible tracking framework has limitations and despite the fact that our working prototype still has not achieved an accuracy level that enables its utilization in real world applications, we believe that it is worth doing further investigations regarding this tracking framework.

To justify our opinion on the usefulness of our method, let's assume that our proposal of an active-marker based tangible tracking system is fully working and the marker being utilized is asymmetrical. In this scenario, the proposed method would have important advantages in respect to usual tangible tracking systems. Below we list and discuss some of the most noticeable advantages our proposal would have in a more mature development stage.

Possibility of creating very small, unobtrusive markers. There are very small and high brightness infrared LEDs for sale. With them, it would be possible to create markers that are small (in height) enough to not protrude much from a tangible's surface. Also, they emit invisible light to the human eye, so if they cannot be felt by the user, then they certainly will not be seen, which would make them unobtrusive markers.

Robustness to occlusion. It would be possible, with very small LEDs, to create complex markers in a special shape so the tracking algorithm's robustness to missing information (due to occlusion) could be improved. Most current systems utilize printed marker-based tracking, which is not robust to occlusion: if part of the marker cannot be identified by the camera, it is not possible to estimate the tangible's position or orientation. This way, the algorithm could be able to reconstruct the occluded part in order to determine position and orientation of the tangible. Moreover, since the marker would be made of high brightness LEDs, some of the light could go through the user's hand, in the case of a handheld tangible, facilitating the tracking and reconstruction of the original marker shape.

Determine height from the table without extra cameras. This is a very important advantage in comparison to current tracking technologies, as current mid-air tracking algorithms utilize extra cameras, besides the one that is utilized to detect touches on the TI. Utilizing high brightness LEDs, it would be possible to increase the height in respect to the table surface from which the tangible is detectable, intensifying this advantage. Detecting the tangible from a height of about 20cm from the tabletop would be enough for the user to move the tangible around and performing our proposed interaction techniques, but the higher the identifiable height is, the better it would be.

Possibility of adding tilt estimation. As mentioned in section 5.2, an accelerometer could be added to the system to identify tilting. This way, our proposed tracking method would have another differential in comparison to other tracking systems, and would allow a richer, more intuitive interaction experience with the interface.

Easy to adapt existing applications. Since the messages sent to the tabletop application by our framework are based on the TUIO protocol, with very few and simple changes on a TUIO client, it is possible to quickly adapt existing applications to work with our tracking system.

For achieving all these advantages, improvements in our current implementation are necessary. But, once they are done and our proposed method reach satisfactory accuracy levels, our tracking system surely will support a new mid-air interaction style in tabletop systems.

VI

Conclusion

This dissertation introduced interaction techniques and a tracking method for touch-sensitive tangibles to be applied in combination with a tabletop interface, especifically for a tabletop CAD application.

Our proposed interaction techniques involve utilizing the tangible as an avatar of the 3D models which the user wants to manipulate. This way, the user transfers control of the model from the tabletop surface to the tangible by touching both the model on the tabletop surface and the tangible at the same time. Right after, the user can manipulate the model as s/he wants by moving the tangible around in the air or performing gestures on the tangible's surface. A set of interaction techniques are discussed in the Interaction Techniques section 3.2.2, previously mentioned in this text. A pilot study to verify the usefulness of the tangible in a 3D application was done, from which we could conclude that the tangible is useful for 3D manipulations, but it was necessary to improve the tangible manipulations' accuracy.

In that sense, to improve the manipulation accuracy utilizing the touch-sensitive tangible, the tracking system was proposed, also motivated by the possibility of utilizing the same camera for doing touch and tangible tracking on tabletop interfaces. For that, an active marker with known geometry was attached to the target tangible, which was tracked by processing the images captured using the camera. A time-division strategy was used where touch tracking time slots were alternated with tangible tracking time slots, which avoided the incorrect identification of blobs generated when the user touched the table and blobs detected by the camera when the tangible's markers overlapped. The tangible tracking algorithm facilitated the determination of its position and distance from the table surface, by calculating the average blob radius from the images captured by the camera.

6.1 Future work

As future work for the tracking system, it is necessary to include the touch detection layer on the tabletop. In our proposal, since reacTIVision will be utilized for such purpose, it means integrating reacTIVision to our framework. Afterwards, the time-division multiplexing strategy has to be implemented, allowing the synchronization between both tracking layers – touch and tangible – and the tabletop's infrared LEDs state. Finally, the information obtained from the tracking has to be reformatted into the modified TUIO protocol that we mention in this dissertation, which means just adding a new field to the messages and also modifying the existing clients in order to support that change. Finally, a CAD application employing our tangible prototype, the proposed interaction techniques and tracking system should be prepared for allowing user evaluations of the interface.

As suggestions or improvements for this proposal, the accelerometer employed in the pilot study could be combined with the tracking system. This way, besides estimating the height of the tangible, it would be possible also to estimate the tilt angle of the tangible in respect to the tabletop surface. Also, regarding the marker, two things could be done: first, its geometry could be recreated in an asymmetric way, allowing determining the orientation of the tangible. A second possibility is to build a marker employing brighter infrared LEDs, such as the TLN233[1], which could not be utilized in this work due to its unavailability, as its end of line was announced.

---

[1] Datasheet: `http://www.semicon.toshiba.co.jp/eng/product_detail/opto/sensor/1263812_15032.html`

Bibliography

Aguerreche, L., T. Duval, and A. Lécuyer (2010), Reconfigurable tangible devices for 3d virtual object manipulation by single or multiple users, in *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*, VRST '10, pp. 227–230, ACM, New York, NY, USA, doi:10.1145/1889863.1889913.

Arduino (2013), Arduino board mega, `http://arduino.cc/en/Main/ArduinoBoardMega/`. Accessed in January, 2013.

Au, O., C. Tai, and H. Fu (2012), Multitouch gestures for constrained transformation of 3d objects, *Computer Graphics Forum (CGF): special issue of Eurographics 2012*.

Blender 2013 (2013), Blender, `http://www.blender.org/`. Accessed in January, 2013.

Bush, V. (1945), As we may think, *The Atlantic Monthly 176(1)*, pp. 101–108, `http://www.theatlantic.com/doc/194507/bush`.

Buxton, B. (2008), The long nose of innovation, *Business Week*, `http://www.businessweek.com/stories/2008-01-02/the-long-nose-of-innovationbusinessweek-business-news-stock-market-and-financial-advice`.

Chang, F., C. Chen, and C. Lu (2004), A linear-time component-labeling algorithm using contour tracing technique, *Computer Vision and Image Understanding*, *93*(2), 206–220.

Cohé, A., F. Dècle, and M. Hachet (2011), tbox: a 3d transformation widget designed for touch-screens, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 3005–3008, ACM, New York, NY, USA, doi:10.1145/1978942.1979387.

cvBlobsLib 2013 (2013), cvblobslib – blob detection library, `http://opencv.willowgarage.com/wiki/cvBlobsLib`. Accessed in January, 2013.

Dietz, P., and D. Leigh (2001), Diamondtouch: a multi-user touch technology, in *Proceedings of the 14th annual ACM symposium on User interface software and technology*, UIST '01, pp. 219–226, ACM, New York, NY, USA, doi:10.1145/502348.502389.

Duda, R., and P. Hart (1972), Use of the hough transformation to detect lines and curves in pictures, *Communications of the ACM*, *15*(1), 11–15.

Ganser, C., A. Steinemann, and A. Kunz (2006), Infractables: Multi-user tracking system for interactive surfaces, in *Proceedings of the IEEE conference on Virtual Reality*, VR '06, pp. 253–256, IEEE Computer Society, Washington, DC, USA, doi:10.1109/VR.2006.86.

Gupta, G. (2010), Pulse: tangible touch, in *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, MobileHCI '10, pp. 471–472, ACM, New York, NY, USA, doi:10.1145/1851600.1851710.

Hancock, M., O. Hilliges, C. Collins, D. Baur, and S. Carpendale (2009), Exploring tangible and direct touch interfaces for manipulating 2d and 3d information on a digital table, in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pp. 77–84, ACM, New York, NY, USA, doi:10.1145/1731903.1731921.

Hofer, R., T. Nescher, and A. Kunz (2009), Qualitrack: Highspeed tui tracking for tabletop applications, in *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part II*, INTERACT '09, pp. 332–335, Springer-Verlag, Berlin, Heidelberg, doi:10.1007/978-3-642-03658-3_38.

Kaltenbrunner, M., and R. Bencina (2007), reactivision: a computer-vision framework for table-based tangible interaction, in *Proceedings of the 1st international conference on Tangible and embedded interaction*, TEI '07, pp. 69–74, ACM, New York, NY, USA, doi:10.1145/1226969.1226983.

Kim, J.-S., D. Gračanin, K. Matković, and F. Quek (2008), Finger walking in place (fwip): A traveling technique in virtual environments, in *Proceedings of the 9th international symposium on Smart Graphics*, SG '08, pp. 58–69, Springer-Verlag, Berlin, Heidelberg, doi:10.1007/978-3-540-85412-8_6.

Koike, H., W. Nishikawa, and K. Fukuchi (2009), Transparent 2-d markers on an lcd tabletop system, in *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, pp. 163–172, ACM, New York, NY, USA, doi:10.1145/1518701.1518728.

Microsoft Kinect 2013 (2013), Microsoft kinect, `http://www.microsoft.com/en-us/kinectforwindows/`. Accessed in January, 2013.

Microsoft PixelSense 2013 (2013), Microsoft pixelsense, `http://www.microsoft.com/en-us/pixelsense/default.aspx`. Accessed in January, 2013.

Nintendo DS 2013 (2013), Nintendo ds, `http://www.nintendo.co.jp/ds/`. Accessed in January, 2013.

Nishino, H. (2010), A 6dof fiducial tracking method based on topological region adjacency and angle information for tangible interaction, in *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, TEI '10, pp. 253–256, ACM, New York, NY, USA, doi:10.1145/1709886.1709937.

Olwal, A., and A. D. Wilson (2008), Surfacefusion: unobtrusive tracking of everyday objects in tangible user interfaces, in *Proceedings of graphics interface 2008*, GI '08, pp. 235–242, Canadian Information Processing Society, Toronto, Ont., Canada, Canada.

OpenCV 2013 (2013), Opencv – open computer vision framework, `http://opencv.org/`. Accessed in January, 2013.

Patten, J., H. Ishii, J. Hines, and G. Pangaro (2001), Sensetable: a wireless object tracking platform for tangible user interfaces, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, pp. 253–260, ACM, New York, NY, USA, doi:10.1145/365024.365112.

PGR FlyCapture 2013 (2013), Point grey research flycapture, `http://www.ptgrey.com/products/pgrflycapture/`. Accessed in January, 2013.

Seifert, J., A. Simeone, D. Schmidt, P. Holleis, C. Reinartz, M. Wagner, H. Gellersen, and E. Rukzio (2012), Mobisurf: improving co-located collaboration through integrating mobile devices and interactive surfaces, in *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*, ITS '12, pp. 51–60, ACM, New York, NY, USA, doi:10.1145/2396636.2396644.

SparkFun 2013 (2013), Sparkfun electronics, `http://www.sparkfun.com/`. Accessed in January, 2013.

SplitCamera 2013 (2013), Splitcamera, `http://splitcamera.com/`. Accessed in January, 2013.

Syphon 2013 (2013), Syphon, `http://syphon.v002.info/`. Accessed in January, 2013.

Trimble Sketchup 2013 (2013), Trimble sketchup, `http://www.sketchup.com/`. Accessed in January, 2013.

Underkoffler, J., and H. Ishii (1999), Urp: a luminous-tangible workbench for urban planning and design, in *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, CHI '99, pp. 386–393, ACM.

Wellner, P. (1993), Interacting with paper on the digitaldesk, *Communications of the ACM*, *36*(7), 87–96.

Williams, C., X. D. Yang, G. Partridge, J. Millar-Usiskin, A. Major, and P. Irani (2011), Tzee: exploiting the lighting properties of multi-touch tabletops for tangible 3d interactions, in *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pp. 1363–1372, ACM, New York, NY, USA, doi: 10.1145/1978942.1979143.

Wu, A., D. Reilly, A. Tang, and A. Mazalek (2011), Tangible navigation and object manipulation in virtual environments, in *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, TEI '11, pp. 37–44, ACM, New York, NY, USA, doi:10.1145/1935701.1935710.

Publications

1. Saphyra Amaro and Masanori Sugimoto. 2012. Novel interaction techniques using touch-sensitive tangibles in tabletop environments, *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces (ITS '12) – Posters*, 347–350, doi: 10.1145/2396636.2396697.

2. Saphyra Amaro and Masanori Sugimoto. 2013. Novel tangible tracking method for tabletop interfaces using active markers, *Proceedings of the 2013 ACM International Conference on Human Factors in Computing Systems (CHI '13) – Workshop.* (Submitted)