

Master thesis

Design of a 3D Interface using a Markerless  
Paper in Augmented Reality Environments  
( 拡張現実環境におけるマーカーレスペーパーを  
用いた 3D インタフェースの設計)

Supervisor: Takashi Chikayama  
Masanori Sugimoto

Yusuke Takeuchi

竹内 優介



Department of Electrical Engineering and Information Systems,  
Graduate School of Engineering, the University of Tokyo

Submitted February 6, 2013

© Yusuke Takeuchi 2013

All Rights Reserved

*Without a lot of supports by my family, this thesis would have never completed.*

## ACKNOWLEDGEMENTS

Experience of my master course at the University of Tokyo was a pleasure and precious days in my life. I would like to appreciate a lot of supports by family, supervisors and friends. Without them, I definitely could not have completed this thesis. In the first place, I would like to thank my supervisor, Professor Masanori Sugimoto, who supervised my research and gives me advice. Even though my English skill is not good, he often lets me do presentation and discussion in English. That encourages improvement of my English as well as my research. He did not mind any expenditure of my research or travel costs for international conferences. That is why I could improve my research without anxiety about budget and have a chance to present my research at an international conference. I also thank Elliot Woods, from *Kimchi and Chips*. He kindly visited my laboratory and gave advice. Especially his advice about the calibration and vvvv were really helpful. I got a lot of architectural knowledge from my best friend, Takahito Hosono who is working as a professional designer at *Cochas*. He advised my research from architectural viewpoints. I wish to express my deep gratitude to my official supervisor, Professor Takashi Chikayama, for him supports. After moving to his laboratory, he gave me a room for my system. Without his supports, I would have never continued my research. I'm also grateful to the members of Chikayama laboratory. I especially thank Eiji Seki who graduated from Sugimoto laboratory and belonging to Chikayama laboratory. Although I was beginner of programing, he kindly helped me and I learned a lot involve my research field and programing skill from him.

Experience in *Interaction Technology Laboratory* must be the largest part in my master course. Saphyra Amaro, best friend of mine, kindly taught me a lot of things include English and information technology knowledge. I really appreciate her kindness. I also thank my best friend, Carlos Morales. I was tutor of him but I learned a lot from him. I will never forget the precious time with him. When I enrolled the University, I could not speak English and did not have any knowledge about my research. My japanese seniors, Shigeki Nakamura and Kyohei Mizutani, always advised



me about research, classes and my job-hunting. They are two of the persons who I respect the most in my life. I often leaned how to write thesis or what I should do with my research from Natapon Pantuwong. He was respected by everyone in my laboratory and one of the best researchers I ever met. I am sure he is going to be best teacher all over the world. I would like to say thank you to all ITL members.

I would like to thank my best friends as well. I especially thank my best friend, Jun Miura from Someya laboratory at the University of Tokyo. I spent plenty of time with him in my master life. The encounter with Jun was absolutely essential for my master life. He thankfully invited me to parties and we made a lot of friends outside campus as well as inside. Going for drinks and trips with team '*ONION*' was relief rime in my master life. They kindly invited me to go for drinks every weekend. Thank you.

Finally, I am deeply grateful to my family. Especially my mother, Setsuko Takeuchi, always supported my life. This is why I could concentrate on my research and finish writing my master thesis.

# TABLE OF CONTENTS

DEDICATION . . . . .	ii
ACKNOWLEDGEMENTS . . . . .	iii
LIST OF FIGURES . . . . .	vii
LIST OF ABBREVIATIONS . . . . .	ix
ABSTRACT . . . . .	x
CHAPTER	
<b>I. Introduction . . . . .</b>	<b>1</b>
1.1 Background . . . . .	1
1.1.1 3D Interface . . . . .	1
1.1.2 Architectural tasks . . . . .	2
1.2 Motivation . . . . .	2
1.2.1 Objective . . . . .	3
1.3 Contribution . . . . .	4
1.4 Structure of this thesis . . . . .	4
<b>II. Related Work . . . . .</b>	<b>7</b>
2.1 Interaction above a surface . . . . .	7
2.2 3D visualization . . . . .	9
2.2.1 Off-axis projection . . . . .	10
2.3 Paper interface . . . . .	11
<b>III. Design Requirements . . . . .</b>	<b>17</b>
3.1 Hardware . . . . .	17
3.2 Software . . . . .	18
3.2.1 3D visualization . . . . .	19
3.2.2 Capability of interaction . . . . .	19

<b>IV. Proposed System</b>	21
4.1 System overview	21
4.2 Implementation	21
4.2.1 Configuration	21
4.2.2 Calibration	22
4.3 3D information space	28
4.3.1 Tabletop display	29
4.3.2 Paper projection	30
4.4 Interaction techniques	36
4.4.1 Interaction techniques using a paper	37
4.4.2 Light effect	39
4.4.3 Menu space	40
<b>V. Experiment and Evaluation</b>	45
5.1 Experiment	45
5.1.1 Accuracy of the paper projection	45
5.1.2 Application	46
5.2 Evaluation	47
<b>VI. Discussion</b>	51
6.1 Discussion	51
6.1.1 Limitations of the proposed system	51
6.1.2 Proposed system and applications	52
<b>VII. Conclusion and Future Work</b>	55
7.1 Conclusion	55
7.2 Future work	55
7.2.1 Multiple users	55
7.2.2 Flexible paper	56
7.2.3 Interaction techniques	57
<b>BIBLIOGRAPHY</b>	59
<b>PUBLICATION</b>	63

## LIST OF FIGURES

### Figure

2.1	Example of a correct perspective view image and the observer’s view	9
2.2	Off axis projection of stereoscopic images . . . . .	10
2.3	The basic gesture set of PaperWindows. (from ( <i>Holman et al., 2005</i> ))	12
2.4	General setup of the top-projected PaperLens (from ( <i>Spindler et al., 2009</i> )) . . . . .	13
2.5	Perspective correction in the PaperLens system, height indicator projected on lens, lens designs (from ( <i>Spindler et al., 2009</i> )) . . . . .	13
2.6	Various prototypes of the PaperLens demonstrating several exploration spaces. (from ( <i>Spindler et al., 2009</i> )) . . . . .	14
2.7	By moving the PaperLens (blue rectangle) through the interaction space ( $X_i, Y_i, Z_i$ , in green) users can study the four types of exploration spaces ( $X_e, Y_e, Z_e$ , in red) (from ( <i>Spindler et al., 2009</i> )) . . .	14
3.1	Two projectors and one tabletop projection screen. . . . .	18
4.1	system overview . . . . .	22
4.2	System configuration: the Kinect acquires the user’s current head and hand position and two projectors are used, for the paper projection and the tabletop projection . . . . .	23
4.3	The geometry and mathematics of the pinhole camera model . . . . .	24
4.4	Two pinhole camera models and a checkerboard . . . . .	25
4.5	(a): Depth image (b): RGB image (c) Point cloud in 3D scene . . .	26
4.6	Kinect device . . . . .	27
4.7	Model of a calibration between a projector and a kinect . . . . .	27
4.8	(a): A checkerboard projection in the RGB image (b): Detecting all corner points on the checkerboard projection in the real world coordinate . . . . .	28
4.9	Setting off-axis projection plane onto the tabletop surface in the 3D scene . . . . .	30
4.10	Left: cross-sectional image, Right: wire-frame rendering . . . . .	31
4.11	Iteration of the plane detection process . . . . .	32
4.12	3D reconstruction of a paper . . . . .	32
4.13	Left: a whole cross-sectional image, Right: the user’s view . . . . .	33

4.14	(a):a paper projection before computing (b):a binary image of the objects above the surface (c):a result image by subtraction (b) from (a) . . . . .	34
4.15	(a) points of the object’s surface in the 3D scene (b) white 3D models are positioned to all points in (a) (c) the points are projected on the paper (d) the white models are projected on the paper . . . . .	35
4.16	The filter, the texture image and the camera are set in the 3D scen.	35
4.17	(a) the 3D objecct and the cutting plane (b) correct perspective cross-sectional image from the user’s view point (c) setting image (b) on the cutting plane (off-axis projection plane) . . . . .	36
4.18	A basic interaction process: Pick up and Release . . . . .	38
4.19	Manipulation of the 3D model using a paper . . . . .	38
4.20	(a) Pinhole window (b) Showcase . . . . .	39
4.21	By moving the user’s hand, the user can move the light source in the 3D scene as if the user was holding the light source. . . . .	40
4.22	Menu options in this system . . . . .	41
4.23	Idea of menu and interaction space . . . . .	42
4.24	General menu display . . . . .	42
4.25	Two bounding box for the switch . . . . .	42
4.26	Feedback icons . . . . .	43
5.1	Average of the pixel error in cases of; (a) The Kinect and projector are facing same direction. (b) The Kinect is facing a user and the projector are facing the tabletop surface (same position as the proposed system) . . . . .	46
5.2	Example of a paper positions and orientations which the paper cannot be detected by the Kinect . . . . .	47
5.3	The reconstructed paper in the 3D scene: (a) Rotation about x-axis (b)Translations on y axis (c), (d) Rotations about z-axis . . . . .	48
5.4	The pictures taken from the user’s view point: (a) Rotation about x-axis (b)Translations on y axis (c), (d) Rotations about z-axis . . . . .	49

## LIST OF ABBREVIATIONS

**CAD** Computer Aided Design

**CT** Computer Tomography

**DOF** Degree of Freedom

**fps** frame per second

**FTVR** Fish Tank Virtual Reality

**GPGPU** General Purpose computing on Graphics Processing Unit

**IR** Infrared

# ABSTRACT

Design of a 3D Interface using a Markerless Paper in AR Environments

by

Yusuke Takeuchi

Supervisor: Masanori Sugimoto

In this thesis, I present an interactive interface that enables a user to work with a virtual 3D information space on a tabletop interface by combining head-coupled perspective interaction above the surface with a mobile paper projection. The mobile paper projection acts as a physical pinhole into the virtual 3D scene. By moving the paper, the user can see cross-sectional image or alternative rendering of the projected 3D model in the 3D scene. An important feature of this work is that the system does not require the attachment of any device, such as tracking marker, to the user or the mobile paper. Several interaction techniques using the paper are also implemented and explained in this thesis. These interaction techniques are focused on architectural tasks. The user can naturally manipulate or inspect the 3D models above the surface using the paper. Besides, the user can move the light source by moving the hand in the 3D scene. The user's head and hands are tracked by a depth camera and all projections in this system are correct perspective images from the user's viewpoint. That is why the user can obtain the immersive 3D illusion. I focus on the application of our proposed system to architectural tasks. The system is evaluated with user experience and experiments of accuracy. Limitations of this system is also discussed.



# Introduction

The purpose of this thesis is to present novel interface which combines a tabletop surface and a handheld paper. This proposed system allows a user to interact with 3D information, such as 3D models. This system is especially focused on a work environment of architecture, such as editing 3D models using 3D Computer Aided Design (CAD), building architectural models, gluing textures on the models and presentation for the clients. I proposed a system would makes their work environment better, comfortable, faster and easy. This chapter mentions background of this research and the tasks of architecture, motivations of this research, contribution and the structure of this thesis.

## 1.1 Background

### 1.1.1 3D Interface

Interaction with virtual 3D models via tabletop surfaces has been developed and improved by a number of researchers. A major achievement has been the extension of the interaction space from a 2D surface (touch-based interaction) to the 3D space above the surface via intuitive and natural interactions (e.g., *Hilliges et al. (2009); Benko and Wilson (2009); Hilliges et al. (2012)* (cf., *Grossman and Wigdor (2007)*). The interaction above the surface (in 3D space) enable the users to directly interact with 3D information though, the two dimensional screen cannot provide 3D illusion to the users. The user should be able to see the projected 3D information as if they existed there to directly interact with the 3D information. There are several methods to make it possible to provide 3D illusion to the user, using shutter glass (such as *Benko et al. (2012); Ugur et al. (2009)*), for example, as mentioned in Chapter II. This proposed system provides immersive 3D illusion to the user using a technique



based on Fish Tank Virtual Reality (FTVR) (*Ware et al.*, 1993) and head tracking.

With increasing works to be related to the interaction in 3D space, some interfaces combining a tabletop screen and a mobile screen has been developed (e.g., *Spindler et al.* (2012, 2009)). These works use another mobile projection screen which acts a pinhole window. This mobile screen helps the user to inspect the 3D models and manipulate them. Recently, tablet gadget (e.g., *iPad* (2013)) has been becoming popular though, I focus on that this work does not require the mobile screen any particular devices. This is why I use projectors and I looked at the fact the user's body surface can be projections. In fact, some interfaces which support body projections have been presented these days (e.g., *Harrison et al.* (2011); *Wilson and Benko* (2010)).

Although extending the tabletop to the third dimension should be applicable to many areas, including medical, education and computer vision, this paper focuses on the field of architecture.

### 1.1.2 Architectural tasks

An architect develops an idea for a 3D building and presents it to the client. Before presenting the idea, several steps are usually required<sup>1</sup>. First, a blueprint, such as the floor plan, is designed on paper. (Nowadays, software is used for designing blueprints, usually as 2D images.) Next, with reference to the blueprint, a 3D model is created and edited using a 3D CAD software. At this stage, most factors are considered, including the size of rooms, wall textures, the arrangement of furniture and lighting. A real model, called the architectural model, is then created, often with textured material glued to some of the walls. Finally, the architectural model is presented and the details of the building are discussed. Usually the presentation should be a competition, hence the presentation is the most important part and they prepare for this so hard.

## 1.2 Motivation

Our system should be applied various areas such as medical, education and computer vision though, this thesis especially focus on architecture field. As mentioned in Section 1.1.2, an architectural work process is composed of several steps; designing blueprint, 3D CAD, building architectural models and presentation. This section presents several problems that many architects complain about and this system fo-

---

<sup>1</sup>Personal communications with friends working at architectural laboratory *Cochas*.

cused in architecture work environment. Actually to solve the problems should be motivation and contribution of this system as well.

### 1.2.1 Objective

There are several problems and difficulties that should be solved and helped in architecture tasks. Main problem is the fact that they need to spend a lot of money, time and efforts building architectural models. This work especially focuses this problem. Sometimes they need to modify the models or remake them from zero due to demands from their clients even though they spent a lot of efforts . Architectural models are used for a range of purpose.

- to compare its size with objects such as trees or people;
- to get an idea of how it looks from different angles;
- to exhibit the design to people such as developers and house buyers, who cannot visualize a 3D design from 2D drawings alone;
- to confirm light and shadow effects;
- to experiment with textures.

Our proposed system helps to achieve these various purposes, which might obviate the need to make architectural models, or at least, the architect would not have to repeatedly rebuild the models. Our system helps not only making architectural models in the process but also 3D CAD. In 3D CAD process, the architects sometimes edit 3D models in the CAD software, such as translation and rotation. However, usually the 3D models have 6 Degree of Freedom (DOF), hence the user cannot manipulate the 3D models using a mouse and a keyboard intuitively. This proposed system supports interactions in 3D space which is above the screen surface, which is why the user can manipulate the 3D models much easier and intuitively. Besides, this work enables the user to get cross-sectional images on any arbitrary planes intuitively and easily. Cross-sectional image is one of the most important in architectural modeling. They often need to see inside of the building by cutting walls or ceilings. However, it is often hard to define a cutting plane using the mouse and the keyboard in the 3D CAD scene. This proposed system supports paper projection. A user hands a paper which is not specific at all and the cross-sectional image cut by the paper is projected on the paper. By moving the paper, the user can see the cross-sectional image on

arbitrary planes. This technique would very help the user to confirm cross-sectional images from various view. This work especially focused the thing the user does not need to wear and hand any specific device, such as tracking markers, head mount display or mobile display simply because of a idea that simple is the easiest.

### 1.3 Contribution

The main contribution of this thesis is that the system approaches the problems which architects often face without any user instruments. In this section, some ideas and techniques which enable the contributions are presented.

As mentioned in Section 1.2.1, this work assists the things that architectural models need for. Regarding size, the users often get paralytic about scaling because they can easily scale using a mouse and a keyboard on 2D display. This is one of the purposes of making architectural models to make sure the volume of the model. This work approached this problem using tabletop surface and correct perspective. The user can fix the scale of models and see the projected 3D models as if they were on the surface, which means the projection enables the user to obtain an environment similar to architectural real models. To enable the user to acquire the image how different look architectural models from various view, this work supports correct perspective projection by head-tracking. The projection of the tabletop screen and paper are seen correct perspective from the user's viewpoint.

Besides, this thesis presents several interaction techniques using a markerless paper with any shape, as mentioned in Section 4.2. These techniques enable the user to interact with the 3D information intuitively and naturally. Changing light effect is also supported and the user can arbitrarily change the position of the light source in this system. The mobile paper screen which is one of important features of this work gives the user another information, such as a cross-sectional image or alternative rendering. These techniques help the user to do tasks in the field of architecture.

### 1.4 Structure of this thesis

Related work are introduced in the following chapter. Almost all related work are described in each section separately, 'Interaction above a surface', '3D visualization' and 'Paper interface'. Off-axis projection is also explained in Section 2.2.

Chapter III mentions design requirements in this system by considering vision of this system and tasks in a 3D scene. There are two principal parts, which are

hardware side and program side. In hardware side, what kind devices are required are mentioned. Besides, what kind of interactions and applications should be supported are described in software part.

Chapter IV explains details of this proposed system. First, configuration of the system and the calibration process based on Zhang's method (*Zhang, 1998*) are explained. After that, the 3D scene, paper projection and tabletop projection are described. How the system avoids overlapping of the two projections, and how 3D reconstruct a paper plane in 3D scene are explained. Besides, the application scenarios and interaction techniques using a paper are also proposed.

Results of some experiments and user feedback are described in Chapter V. Accuracy of the paper projection is mainly mentioned. In addition, I obtained feedback from a architect. They are also mentioned.

Chapter VII conclude this thesis and describes some future vision of this work.



This work can be separated several parts and there are a lot of related work for each part. This chapter presents such related and previous work. This work is composed of three parts: interaction above a surface, 3D visualization and paper interface. Besides, the technique of the 3D visualization is also mentioned this chapter.

## 2.1 Interaction above a surface

Multitouch interaction has been becoming a normal interaction method in our life. Since multitouch enables users to interact with the device more intuitively by feeling to be touching 2D image, such as icons and pictures. However, multitouch is not so intuitive for manipulations of 3D objects, such as computer graphics. A 3D object usually has DOF, whereas a multitouch interface supports only three-DOF manipulations, namely 2D translation on the surface and rotation about the normal vector on the surface. Therefore, multitouch interaction is not well suited to manipulating 3D objects, which is why the input space for interfaces has been extended recently from the 2D surface to the space above the surface (e.g., *Hachet et al.* (2011)). A interaction above a surface is usually enabled by 3D reconstruction of real objects above the surface using depth cameras and applying physics engine to the 3D reconstruction.

*Hilliges et al.* (2009) proposed interaction above a tabletop surface that enables a user to pick up 3D objects generated by a physics engine in a 3D scene. The system recognizes the user's hand gesture of a finger-and-thumb circle by processing the Infrared (IR) image. When the user makes a circle above the surface, the system recognizes this as the 'pickup' gesture, enabling the user to pick up a 3D object.

*Wilson* (2007) presented an interactive tabletop system that uses a depth camera to build a height map on the table surface. The height map is used in a driving simulation game that enables players to drive a virtual car over real objects placed on the table. Objects on the surface are captured and reconstructed by the depth camera. The cars and reconstructed 3D objects are applied to a physics engine, enabling players to build a driving course by folding paper on the surface.

*Benko and Wilson* (2009) applied interaction in the 3D space in front of a screen to a multitouch interface. This work supports head tracking, which enables the user to obtain an immersive experience and to interact more intuitively. Simply due to make sure the user can feel touching directly and correctly, some interfaces which support interactions in a 3D space such as above the surface also support head tracking to show correct perspective from the user's view. Which is why this proposed system also provides correct perspective images to a user by head tracing. There are several proposals that combine 3D interaction techniques with a 3D immersive environment, such as CAVE (*Cruz-Neira et al.*, 1993).

*Wilson and Benko* (2010) presented an interface which enable users to interact above and between surfaces by combining multiple depth cameras and projectors. This work emphasized three concept themes which are 'Surface everywhere', 'The room is the computer' and 'Body as display'. All physical surfaces which are captured by several depth cameras should be interactive displays and not only these physical surfaces but also the space between them is available for interaction such as connecting one to another surface by touching both simultaneously. Graphics may be projected onto the user's body to enable interactions in mid-air such as holding a virtual object as if it were real or making a selection by a menu projected on the user's hand. Three depth cameras and three projectors are hung the ceiling to ensure good coverage of the interactive space. The cameras and projectors need not to be precisely positioned since calibration make them into one 3D coordinate. Usually the system requires a camera and a projector calibrated between them. Details of calibration process of this system is mentioned in Section 4.2.2.

*Hilliges et al.* (2012) presented an interactive system, Holodesk, which is combining an optical see through display and a Kinect. A user can interact with 3D models projected on the optical see through display in space behind the display. The shape of any physical objects in the interactive volume is known by the Kinect depth buffer and the approximates shape is reconstructed in a 3D physics simulation by applying a sphere to each point of the depth buffer. The tracking process is implemented General Purpose computing on Graphics Processing Unit (GPGPU) and based on the

research by *Brox et al.* (2004). This process allows the user to feel direct interaction with 3D objects.

Nowadays, thanks to Kinect device, we can get approximate shapes of physical objects easily and cheaply. Besides, physics simulation allow to interact between the real physical objects and virtual objects in a 3D scene. Finally users can feel direct interaction with 3D objects (such as *Hilliges et al.* (2012); *Wilson and Benko* (2010); *Benko et al.* (2012); *Izadi et al.* (2011); *Dippon and Klinker* (2011)).

## 2.2 3D visualization

Interactions in the 3D space indeed provide intuitive interactions for manipulations of the 3D model. However, if the user does not know exact position the user should interact, the user cannot interact with the 3D objects correctly (cf., *Wayne Arthur* (1993)). The projection should be correct perspective rendering from the user's viewpoint. Which is why this work supports head tracking and provides correct perspective projection. Methods of rendering to tabletop screen have been researched every year (*McKenna*, 1992; *Hancock et al.*, 2009). This Section introduces several works which are about 3D visualization using head tracking.

*Ware et al.* (1993) established foundation of virtual reality for 3D visualization. FTVR provides a stereo image of a 3D scene viewed on a monitor using a perspective projection coupled to the head position of the user. The 3D visualization method is called 'head coupled'. Head coupled display provides correct perspective view images and the resulting scene can be either viewed monocularly, coupled to a single eye position, or binocularly, if suitable stereo equipment is available. Therefore, if the image is projected on the screen, the observer can see the 3D objects as if they were there (see Figure 2.1).

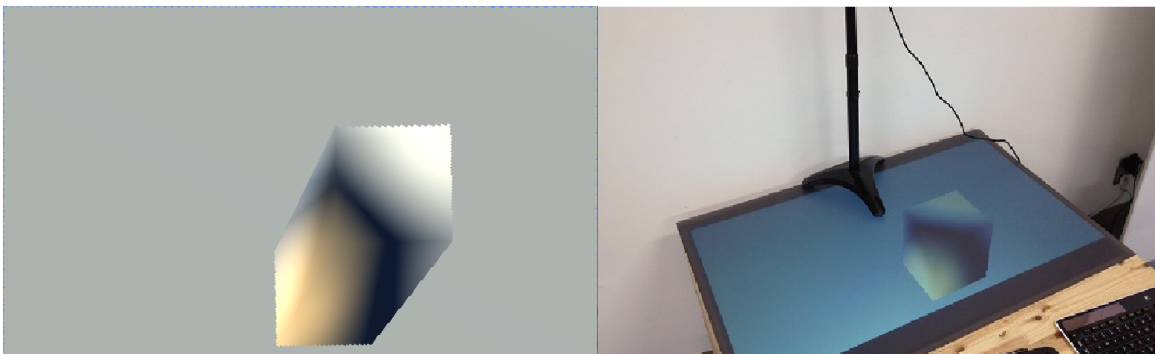


Figure 2.1: Example of a correct perspective view image and the observer's view



### 2.2.1 Off-axis projection

The correct perspective view image is technically acquired from off-axis projection. Usually rendering projection use a pinhole camera model which has one projected view frustum and the frustum usually should be symmetric. In off-axis projection case, the frustum is not symmetric. To acquire stereoscopic images, two cameras are set at some location (Figure2.2).

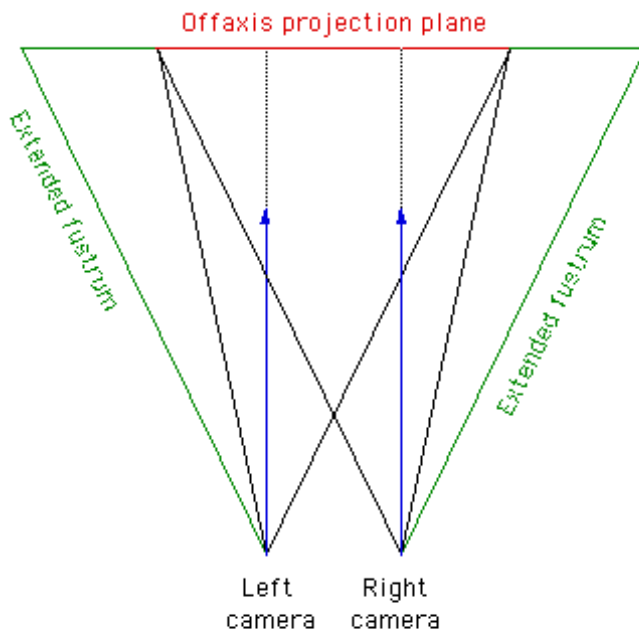


Figure 2.2: Off axis projection of stereoscopic images

The left camera and right camera have a distance called 'stereo base' which corresponds to the eye distance. The cameras with parallel direction vectors are used which means that the right and left cameras see a different part of the image. The overlapped image is a off-axis projection and other parts which are seen by only one camera should be cut off after rendering. Finally, the frustum of each camera is asymmetric. Each image, captured by the right and left camera, should be a correct perspective view from each eye, hence it is enough for monocular image to use only one camera. Some rendering package support off-axis projection such as *OpenGL* (2013).

*Ugur et al.* (2009) presented a stereoscopic head-tracked horizontal display system. The system tracks a user's head direction using helmet with three LEDs by a stereo camera and projects stereoscopic off-axis projection on the horizontal display. The

user can see the projection through a shutter stereo glasses, and then the system creates a realistic visualization system for 3D models.

*Agrawala et al.* (1997) presented the two-user Responsive Workbench: a virtual reality system that allows two people to simultaneously view individual stereoscopic image pairs from their own viewpoints. Usually a correct perspective view image by head tracking is available for only one person. This project computes four images which are two stereoscopic images from each user's viewpoint and uses two stereo shutter glasses to show correct image for each user. There are several work which support multiuser (e.g. *Kitamura et al.* (2001); *Ye et al.* (2010)). Idea of multiple users is mentioned in Chapter VII.

*Francone and Nigay* (2011) implemented head-coupled on mobile device such as smartphones and tablets using only built-in camera. A user's face is tracked by processing RGB camera image in real time and the distance between the user's head and the device is estimated by the size of the face. This resulting image is not stereoscopic image but monocular 3D image. They provide a natural and interactive 3D view: the user has the illusion to look at a small window instead of a flat screen. This idea is one of the good points of mobile devices. This proposed system also uses a mobile projection screen acting a pinhole window.

## 2.3 Paper interface

A handheld paper projection screen is a useful interactive interface that can improve these systems. The second projections makes cross-sectional images of projected 3D objects available to the user. There are several proposals that use handheld mobile projections via paper interfaces (*Hirota and Saeki*, 2007). Besides, interaction techniques using a paper are also introduced in this section.

*Holman et al.* (2005) presented a prototype windowing environment that simulates the use of digital paper displays. By projecting windows on a physical paper with tracking marker, the user can see the information, such as web pages or pictures. This work is motivated by the natural manipulation of papers and supports a number of interaction methods, which include hold, collocate, collate, flip, rub, staple, point and two-hand pointing (see Figure 2.3). Furthermore, the user can interact with the paper using pens, fingers or other objects by tracking them.

*Spindler et al.* (2009) presented interactive interface, PaperLens, using a handheld paper called magic lens. Figure 2.4 shows the configuration of PaperLens. An IR camera and a projector are hung over a user's head and the IR camera tracks a

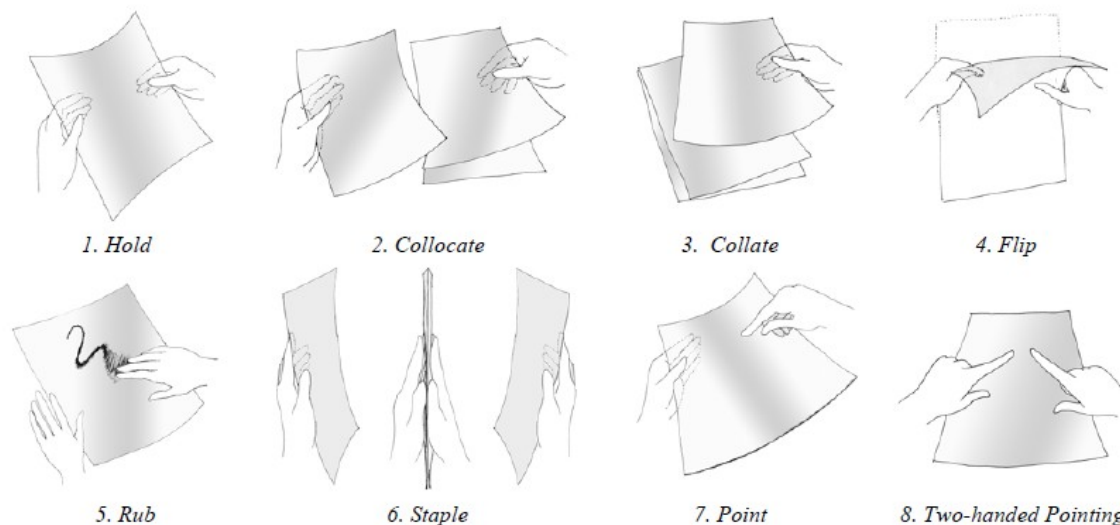


Figure 2.3: The basic gesture set of PaperWindows. (from (*Holman et al.*, 2005))

paper's 6DOF (see Figure 2.5, (d)) IR-reflecting markers are glued onto the corners of a rectangular PaperLens to detect the paper's exact position and orientation.

They implemented four prototypes of PaperLens, which are Volumetric information space, Layered information space, Zoomable information space and Temporal information space (see Figure 2.7). Volumetric information space can be described as a set of voxels (see Figure 2.7 (a)). Depth- and XY-translation can be mapped to simple shifts of height ( $Z_i$ ) and movements in  $X_i, Y_i$ , while tilting provides a natural method for 'slicing'. Typical volumetric datasets are from medical or geological domains, such as Computer Tomography (CT) or seismic data that can be explored with PaperLens (see Figure 2.6(a),(b)).

This technique allows users to scan through 3D volumetric datasets of human bodies by moving the PaperLens above silhouette of a human. Layered information spaces is defined as sets of 2D image data. The projection (layer information) is decided by the paper height and continuous within a 2D plane ( $X_i Y_i$  space). They implemented four layers: the skeletal, muscular, blood, and nervous systems of the human body as application. With zoomable information spaces, large continuous 2D information worlds can be represented that extend far beyond the tabletop in their original size (see Figure 2.7). By lifting the PaperLens on Z axis, the user can zoom into the information world such as maps. This metaphor is similar to the experience of many users who bring interest objects closer to their eyes in order to examine them in more detail.

Temporal information spaces are a concept for describing 2D time-dependent data,

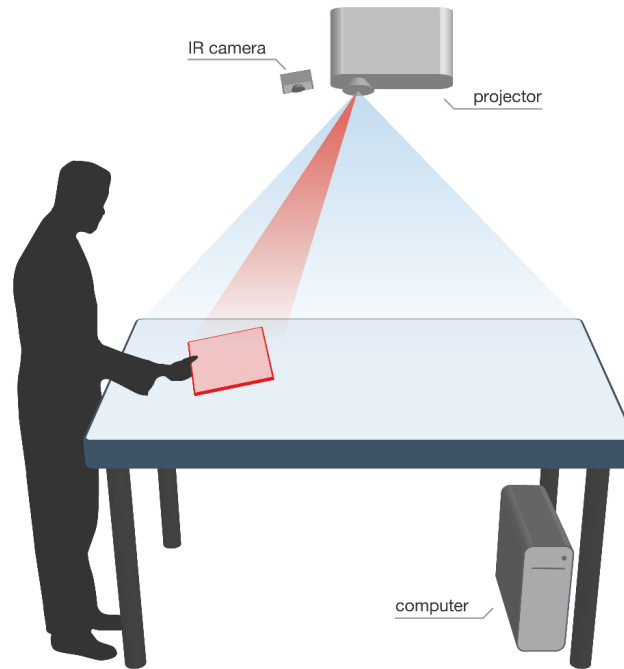


Figure 2.4: General setup of the top-projected PaperLens (from (*Spindler et al.*, 2009))

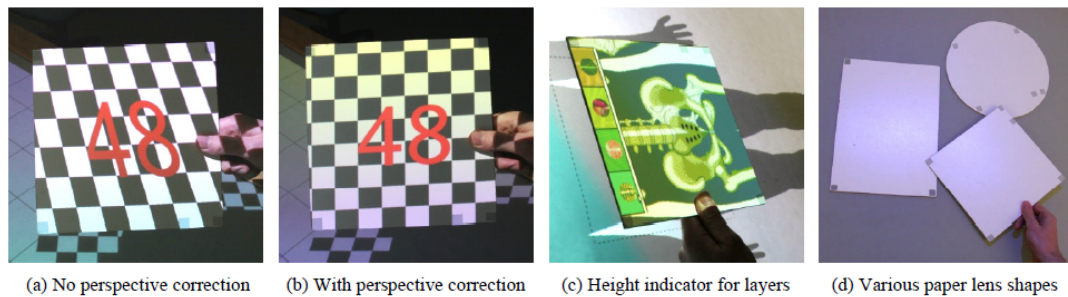


Figure 2.5: Perspective correction in the PaperLens system, height indicator projected on lens, lens designs (from (*Spindler et al.*, 2009))

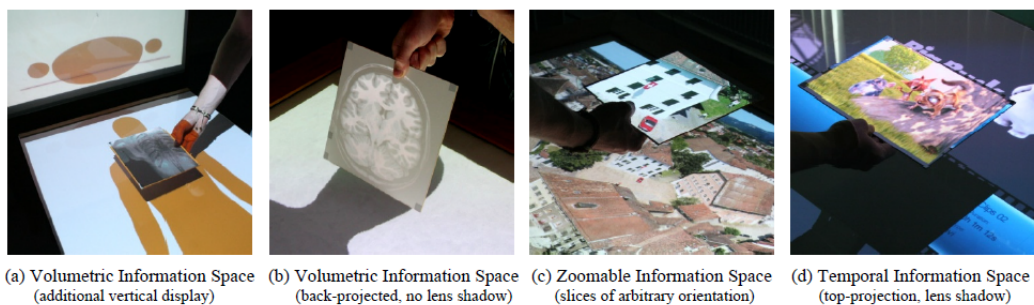


Figure 2.6: Various prototypes of the PaperLens demonstrating several exploration spaces. (from (*Spindler et al.*, 2009))

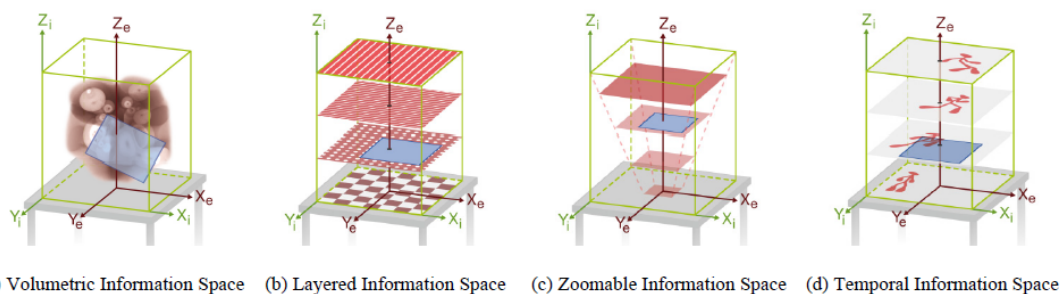


Figure 2.7: By moving the PaperLens (blue rectangle) through the interaction space ( $X_i, Y_i, Z_i$ , in green) users can study the four types of exploration spaces ( $X_e, Y_e, Z_e$ , in red) (from (*Spindler et al.*, 2009))

such as video sequence. As illustrated in Figure 2.7, each temporal image (time frame) is virtually stacked each layer. By lifting/lowering paperlens, the user can see image of one frame (state). Additionally the tabletop screen shows collection of several temporal data sets (e.g. video snippets). PaperLens are also considered correct perspective view from user’s viewpoint to compensate for projection errors. *Spindler et al.* (2012) improved the PaperLens project and presented Tangible Windows whose concept is interacting for virtual 3D information spaces on a workbench such as multi-display environment. Tangible Windows also use a paper looks like PaperLens which is their previous work. They either act as physical peeholes into a virtual 3D world or as physical containers for parts of that world. This work is a multi-display system that consists of two basic types of displays: a single stationary tabletop screen (global display) and one or multiple mobile screen (local displays) that provide independent views into a 3D information space. Global display is shared by all of users and supports multi-touch interaction. Local displays are interactive mobile displays such as PaperLens. The main objective of local displays is to provide several

users with personal view into the virtual 3D world. They provide local modification of objects on general display, alternative representation (e.g.. wire-frame rendering) and perspective images. Local displays are used as not only projection screen but also interactive tangible device with pressure-sensitive buttons. The sensors allows users to input by touching and pen on mobile displays. To track a user's head position and look direction, the users need to wear a crown glued small and IR reflective markers. Twelve IR cameras allow for precise determination of position and orientation (6DOF) of mobile displays (local displays) crowns at 100Hz with tracking error of less than 3mm, they mentioned.



## Design Requirements

In this chapter, design requirements of this work are separately mentioned in hardware and software part. The idea and concept is to interact with 3D information on a workbench-like environment (*Kruger et al.*, 1995). I envision a system that enables a user to directly interact with virtual 3D models and to obtain immersive 3D illusion. This system avoids requiring the user to wear any devices, such as head-mounted display, tracking equipments or heavy displays in order to make this system more useful and novel (*Papagiannakis et al.*, 2008; *Kim et al.*, 2011; *Azuma et al.*, 2001; *Lee et al.*, 2008). Besides, I strongly believe that this system should be able to apply to various fields. Therefore, this system should have abilities of various interaction techniques.

### 3.1 Hardware

Combining two displays makes this system more powerful and productive interface. One big display is not mobile and this is a tabletop projection screen. The other one is a mobile paper projection. However, the paper does not need any additional device, such as tracking markers. That's why in this system, only one big tabletop projection screen is prepared. To avoid overlapping of two projections which are the tabletop and a paper projection, a projector for the tabletop projection is positioned behind the tabletop projection screen. Besides, to be able to track user's bone, height of the tabletop is must be shorter than human's hip. That's why the projector has to be a short throw projector. The other projector needs to be positioned in order to cover an interaction space which the user can interact with the 3D information (see Figure 3.1).

To provide correct perspective projections on each projection, the user's head



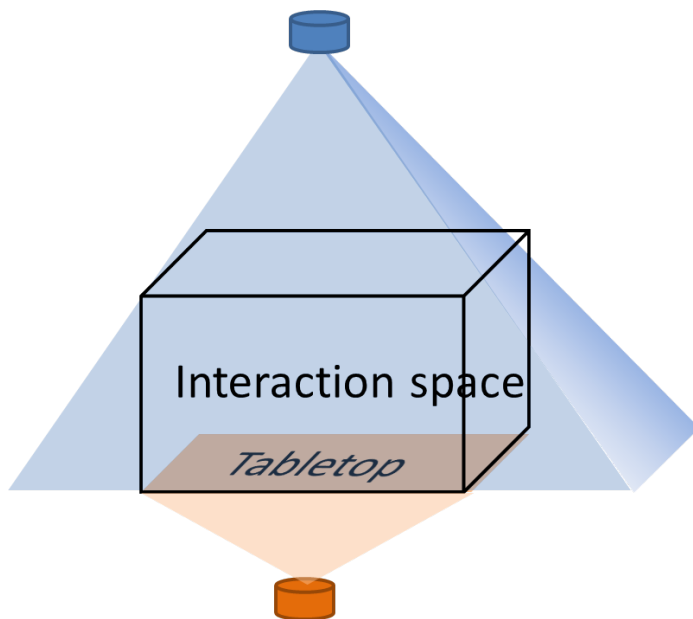


Figure 3.1: Two projectors and one tabletop projection screen.

position needs to be detected. Besides, this system needs to know an equation of a paper in the interaction space without any tracking markers. Because my intent is that the system does not require any wearable devices or tracking markers, the system has to be able to track the paper and the user without them. Even though this work focuses on the aspect that the system does not require a particular paper, there is one restriction that the paper has to be rigid. To unify every coordinate into one coordinate, each device needs to be calibrated. To calibrate them correctly, each device has to be positioned to share their field of views. Considering using a paper for interactions and projection, the height of the tabletop screen should be about human's hip.

### 3.2 Software

Usually interaction tasks that users intend to accomplish are *navigation*, *selection* and *manipulation* including changing viewpoint and applying translation and rotation to 3D models or the whole 3D scene. To accomplish their purposes intuitively, this system should allow the user to feel in the 3D world and directly interact with the 3D information.

### 3.2.1 3D visualization

To enable the user to obtain the immersive 3D illusion, each projection should be correct perspective image. Even though there are a lot of works to use stereoscopic projection using shutter glasses, this system does not focus on this aspect because of my intent that this system does not require a user to wear device. Instead, this system supports monoscopic rendering. Monocular cues involves relative size, texture gradients, linear perspective, occlusion and motion parallax.

### 3.2.2 Capability of interaction

As mentioned in Chapter II, 3D interaction enables users to interact with the 3D information intuitively and naturally. In this system, the user should be able to manipulate the 3D model. Besides, considering the fact that the user is focusing the particular the 3D model during inspecting using the paper, the user should be able to interact using the paper. In architectural tasks, following tasks are required. First, as mentioned in Section 1.1.2, architects often need to define a cross-sectional plane in the 3D scene. Besides, they examine how the 3D model is looked from various viewpoints. That is why the system should enable the user to define the cross-sectional plane and the view point intuitively. General tasks involve rotation, translation, copy and paste are also required in architectural tasks. Especially rotation about the vertical axis on the tabletop surface is required in order to arrange 3D models on the floor. Examining texture and light are also important issue especially in the architectural model phase. They often glue textures on the architectural models repeatedly. Two of the purposes of building the architectural model are examining the texture and light. This system's vision is that the system helps the user examine architectural issue without the architectural models. That is why the ability of the system should cover what the architects require for the architectural model.



I approached several problems that architects often face in their work environments (mentioned in Chapter I). Details of this proposed system, how the system approaches and actually works, are described in this chapter.

## **4.1 System overview**

This section mentions a whole idea of this system. Figure 4.1 shows this system overview. The tabletop screen shows 3D models, such as buildings or houses. The user's head is tracked by Kinect and the image projected on the tabletop surface is correct perspective view image from the user's viewpoint. The user can acquire cross-sectional images on arbitrary plane with a markerless paper. The system enables a user to interact with the projected objects above the surface using the paper or the user's hand. Besides, by moving the user's hand position, the user can move the light source in the 3D scene. This system supports several models, hence the mode can be changed by moving the user's hand naturally.

## **4.2 Implementation**

This section mentions details of the system configuration and implementation. What this section mentions are based on the design requirements (Chapter III).

### **4.2.1 Configuration**

Figure 4.2 shows the configuration of this whole system. I prepared two projectors, for the tabletop projection (CP-AW100N *HITACHI* (2013)) and the paper projection

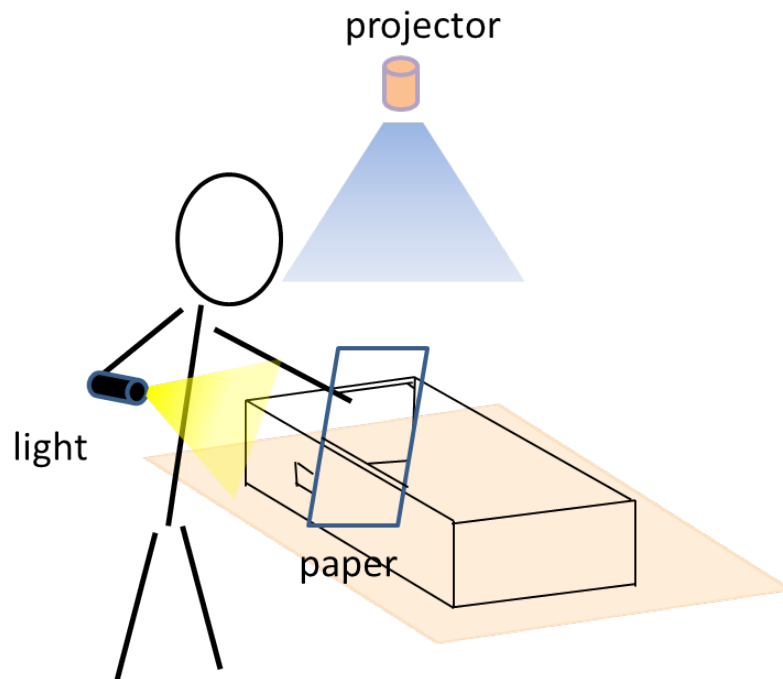


Figure 4.1: system overview

(EB-435W *Epson* (2013)), and a *Kinect* (2013) to detect the user's head position and to reconstruct real objects above the surface in three dimension. A big projection screen, 600mm x 900mm, is prepared as the tabletop surface. The height of the tabletop is 600mm by considering to be able to track the use's body. Both projector's resolution are 1280x800. To avoid generating shadows, the projector for the tabletop projection is mounted under the surface. The Kinect and the hung projector are set by considering that the two devises can share image and the Kinect can aquire the real objects above the surface. The Kinect can acquire the depth data within the distance from 40cm to 350cm so the Kinect is positioned at a distance of 40cmm from the edge of the tabletop. The Kinect and the projectors are calibrated, with each coordinate system being unified into one real-world coordinate system. I use a calibration process proposed by *Zhang* (1998) for this system calibration (mentioned in the following section).

#### 4.2.2 Calibration

This system uses one Kinect and two projectors. A projector for a paper needs to be able to project a image on the the paper accurately. To make the coordinate systems of the kinect and projector unify into one coordinate system, these devices



Figure 4.2: System configuration: the Kinect acquires the user's current head and hand position and two projectors are used, for the paper projection and the tabletop projection

must be calibrated first. In this section, details of the calibration are described. Contrary to cameras, due to the fact a projector cannot capture its view image by itself, the calibration process of a projector is different from a calibration process of a camera. The purpose of the calibration is to find an intrinsic parameter of each device and an extrinsic parameter, rotation and translation, between two devices. I utilized a checker calibration method presented by *Zhang* (1998).

#### 4.2.2.1 Projector and camera model

In this calibration process, pinhole camera models are used for the projector and the camera. Figure 4.3 describes basic pinhole camera model and the mathematical relationship between the coordinates of a 3D point  $(X, Y, Z)$  and the image plane  $(x, y)$  of the model.

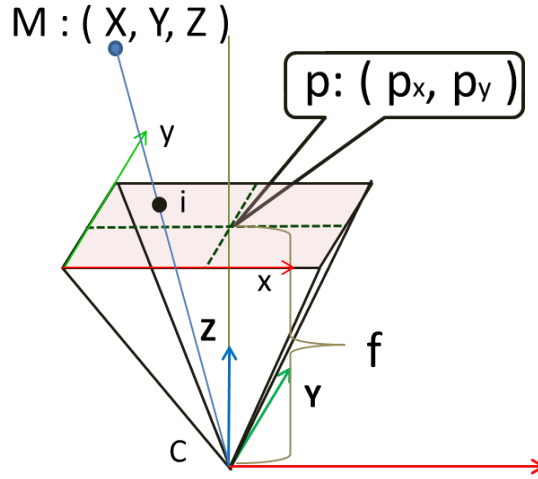


Figure 4.3: The geometry and mathematics of the pinhole camera model

The 3D world is mapped on the image plane of the plane image which is located at distance  $f$  called *focal length* from the origin  $c$  in the camera world coordinate  $(X, Y, Z)$  and parallel to the  $X$  and  $Y$  axis. The point  $p = (p_x, p_y)$  at intersection of the  $Z$  axis and the image plane is referred to as the principal point or *image center*. The point  $i$  is a projection point from a point  $M = (X, Y, Z)$  somewhere in the world and the point  $i = (x, y)$  is at intersection of a line, called *bipolar line* between  $c$  and  $M$ , and the image plane.

Mathematical relation between  $M = (X, Y, Z)$  and  $i = (x, y)$  is expressed by Equation 4.1.

$$[i, 1] = \frac{1}{d} \cdot A \cdot M \quad (4.1)$$

$A$  is usually called an intrinsic camera parameter, where

$$A = \begin{pmatrix} f_x & s_\theta & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{pmatrix} \quad (4.2)$$

$s_\theta$  represents the skew of the lens.

Actually, the pinhole model is able to be applied to the projector calibration by regarding a projector as reverse of a camera. Finally this pinhole model is used for all cameras in this system which are projectors and Kinect IR and RGB camera.

#### 4.2.2.2 Zhang's calibration

A calibration of this system is based on a calibration process presented by *Zhang* (1998). Zhang's calibration method uses a checker board pattern for a calibration for two cameras. Figure 4.4 shows geometrical relation between two pinhole camera models and a checkerboard.

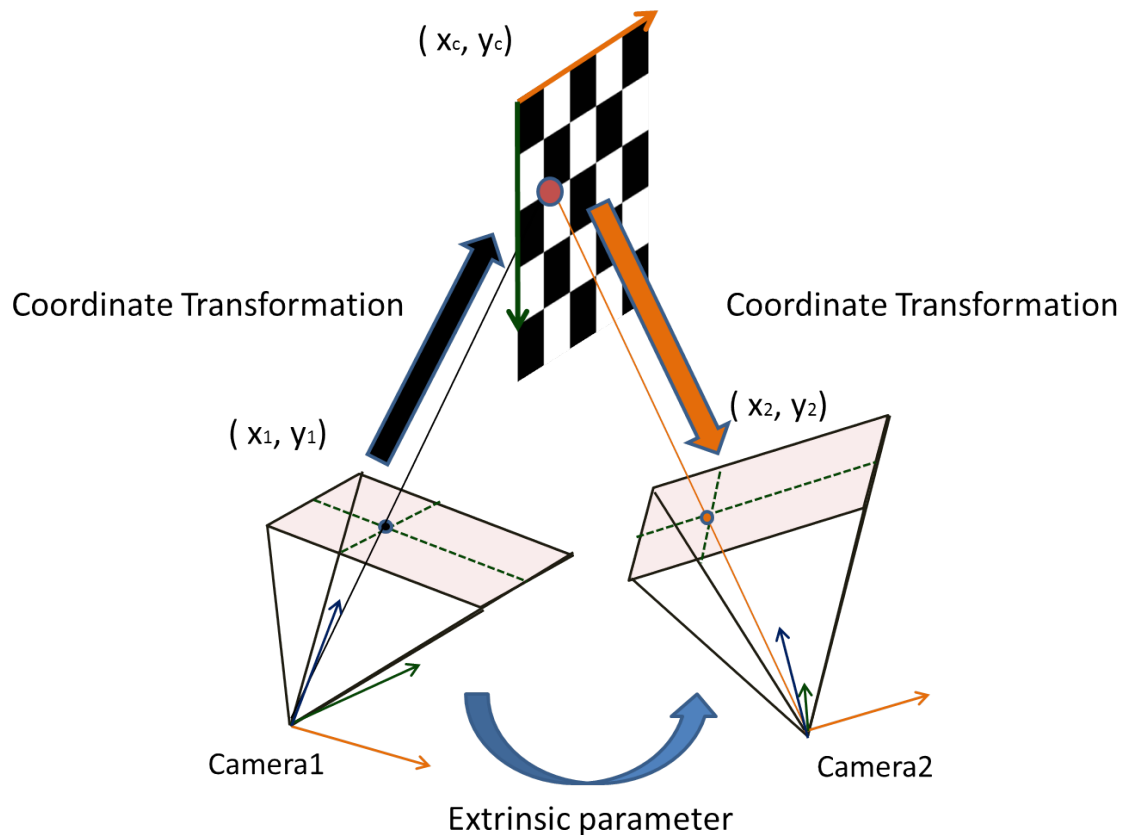


Figure 4.4: Two pinhole camera models and a checkerboard

In Zhang's calibration method, a checkerboard is used to find each intrinsic and an extrinsic parameters, rotation and translation, between two cameras. Because two cameras can easily detect all corner points as feature points on checkerboard. I used *OpenCV* (2013) library to find each corner point.



## 4.2.2.3 3D reconstruction

First, Kinect IR camera and RGB camera must be calibrated. Kinect has two cameras, IR and RGB. IR camera gets depth image whose pixels have depth data that the distance from the IR camera. Figure 4.5,(a),(b) shows the depth image and the RGB image. As you can see, there is a gap between them due to the distance between the IR and RGB camera (see Figure 4.6). Besides, to acquire the corner points of the checkerboard in the Kinect coordinate system (the real-world coordinate in meter), I need to find the intrinsic parameter of the IR camera. Each pixel  $u = (x, y)$  in the RGB map  $D_i(u)$  is converted to point  $M = (X, Y, Z)$  in Kinect coordinate system (the real world coordinate). Given the intrinsic matrix  $K$  (of the Kinect infrared camera), the conversion equation is expressed by Equation 4.3.  $R$  and  $t$  are the rotation and translation between the Kinect IR and RGB camera.

$$M = K \cdot [u, 1] \cdot [R|t] \cdot K \quad (4.3)$$

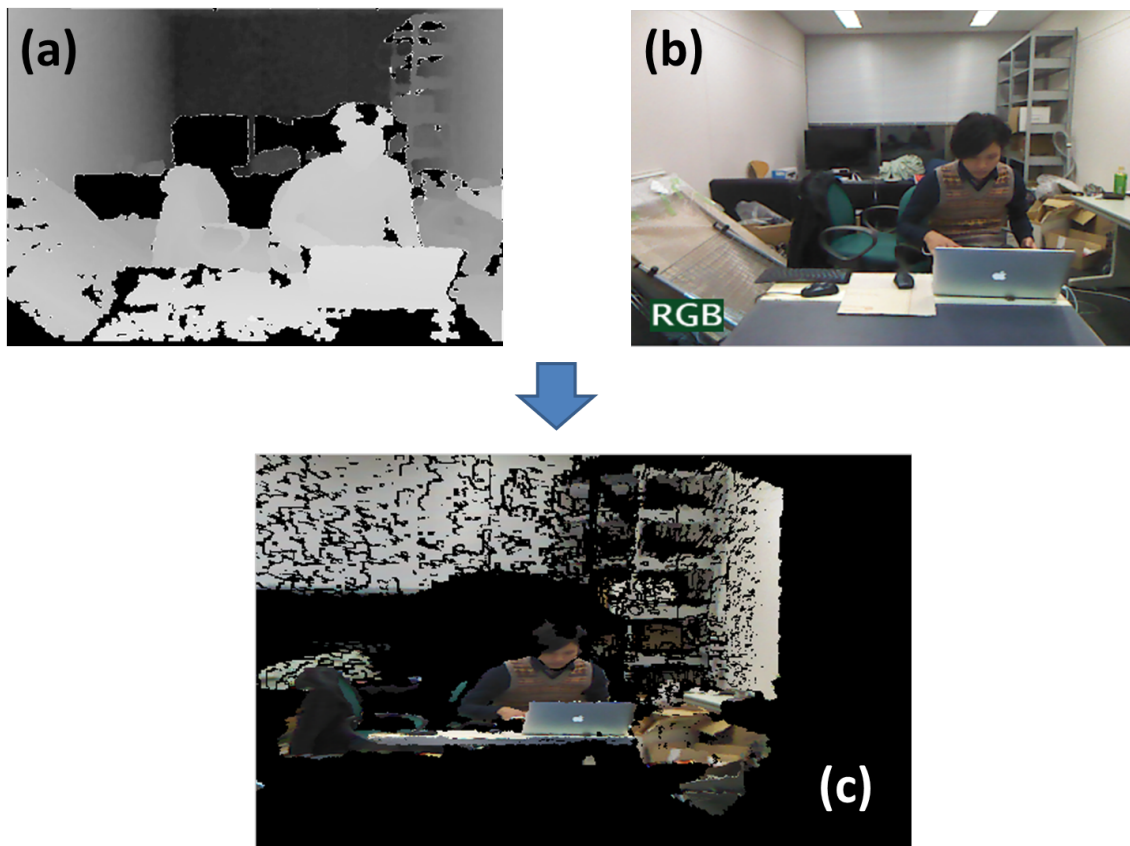


Figure 4.5: (a): Depth image (b): RGB image (c) Point cloud in 3D scene

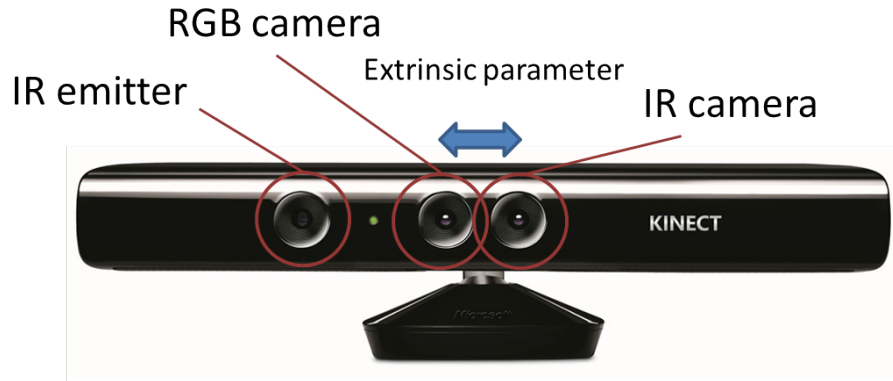


Figure 4.6: Kinect device

The points which are converted from the depth image to the real world coordinate are called *point cloud*. The point cloud is used to find a paper equation and the corners of the checkerboard in this calibration process. Figure 4.5,(c) shows the point cloud mapped RGB data in real world coordinate. The origin in the real world coordinate is the location of the Kinect IR camera. The coordinate system is a general coordinate system in the 3D scene. I used *vvvv* (2013) tool kit for the 3D rendering because *vvvv* show the rendering in real time, while the user is editing though. To acquire the point cloud robustly, I use (*OpenNI*, 2013) as the Kinect's driver.

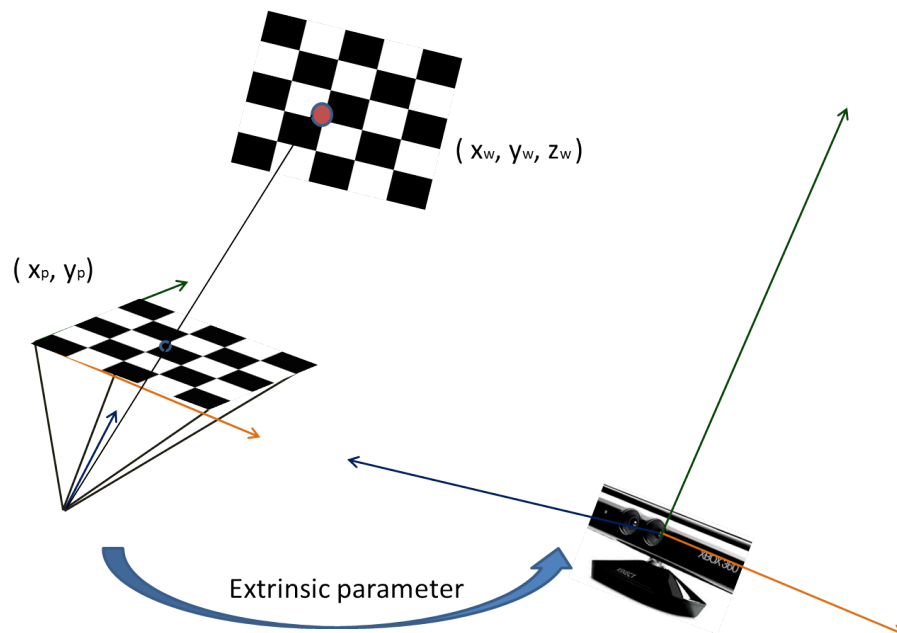


Figure 4.7: Model of a calibration between a projector and a kinect

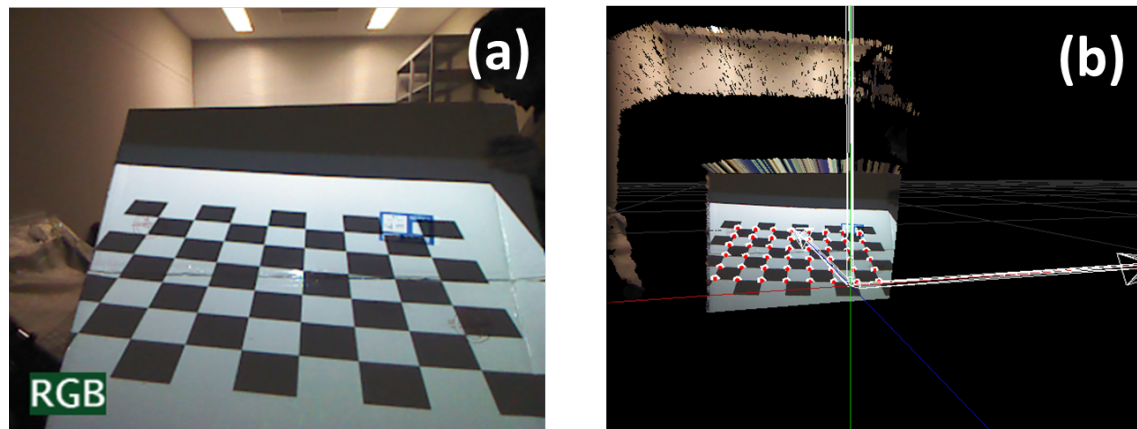


Figure 4.8: (a): A checkerboard projection in the RGB image (b): Detecting all corner points on the checkerboard projection in the real world coordinate

#### 4.2.2.4 Projector-Kinect calibration

I applied two cameras calibration model (Figure 4.4) to the calibration between the projector and the kinect by regarding the projector model as a reverse of a camera model. Figure 4.7 shows this projector-Kinect calibration model. Note that the projector projects a checkerboard on a plane. With it, although a projector cannot capture its view by itself, I can know each corner point in the projector plane coordinate. I prepared a mobile plane or a wide cloth as a projection screen and capture the image various scene by moving the screen many times (see Figure 4.8), then kinect acquires each corner point of the checker board projection in real world. Finally I can know the extrinsic parameter between the projector and Kinect.

### 4.3 3D information space

I create several parallel 3D scenes including for a paper projection and a tabletop projection in vvvv. All coordinate systems are the Kinect coordinate system. my vision is to interact with the 3D information above the tabletop surface. Main interaction tasks that users intend to accomplish within virtual 3D information space are *navigation*, *selection* and *manipulation*. They are based on several fundamental tasks which are changing the view point and applying transformations onto 3D objects. Even though the goal is that the proposed system is useful for any other fields which involve computer vision, education or medical as well as architecture, I focus on interaction techniques for tasks in the field of architecture in this thesis.

### 4.3.1 Tabletop display

As mentioned in Chapter I, an architectural model is required in their tasks to present or discuss the building ideas. This *tabletop display* shows 3D building models for users, such as an architectural model. However, just projecting 3D models on 2D screen is not sufficient for inspecting 3D models. The projection should be correct perspective image from the user's view point. In this proposed system, the user is supposed to walk around the tabletop and inspect the 3D models. To make it possible to enable the user to inspect and manipulate the 3D objects intuitively and naturally, the user should be able to see the 3D objects projected on the tabletop screen as if they existed on the surface (see Figure 2.1). As mentioned in Chapter II, technically, this 3D visualization based on off-axis projection techniques. Whereas some VR displays support stereoscopic rendering of 3D scenes, I do not focus on this aspect here. Instead of a stereoscopic image, I adopted this technique as monocular view whose cues involve relative size, texture gradients, linear perspective, occlusion and motion parallax.

The Kinect tracks a user's head position in the real world coordinate in real time, then the position of a camera in the 3D scene is synchronized with the user's head position and off-axis projections are applied. In Figure 2.2, the off-axis projection plane is set to the projection screen in the real world. In tabletop display case, this is set to the tabletop surface in the 3D scene (see Figure 4.9). Due to moving parallax, the objects in front of the plane shift to opposite of user's view movement. Opposed to that, the objects behind the plane shifts same direction of the user's view movement. For example, when a user move their head to right, objects above the tabletop screen will appear to move left while those behind it will seem to move right. Note there is no visual shift within the plane. This technique enables the user to obtain immersive 3D illusion and applied to a paper projection as well.

In this proposed system, space above the tabletop should be an interaction space which enables the user actually to interact with the 3D information. I define bounding boxes in this system for such spaces. Only the objects inside the bounding box can be detected by the Kinect.

Currently, this tabletop display provides the correct perspective image for one user using the head tracking technique. However, if the system supports multiple users, the projection cannot be the correct perspective image. Multiple users application is mentioned in Chapter VII. Besides, this tabletop surface does not support multitouch interactions. As mentioned in Chapter I and II, multitouch interactions are developed

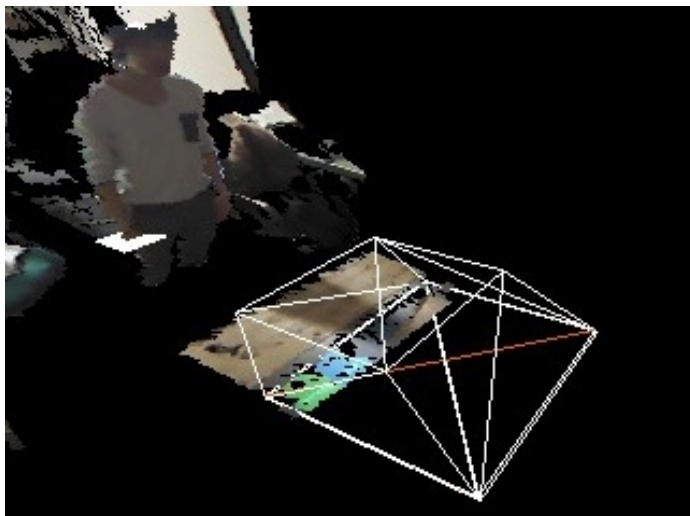


Figure 4.9: Setting off-axis projection plane onto the tabletop surface in the 3D scene

by many researchers and not suit for the manipulation of 3D model so this interaction methods are focused on the interaction above (not onto) the surface.

### 4.3.2 Paper projection

To give users additional information in this system, we prepare a second projection screen as a mobile screen. The mobile screen acts as a pinhole window into the 3D scene. Just as if the user were actually in the 3D scene, the user can move the paper to an arbitrary (the limitation is mentioned in Chapter VI) position and orientation, thereby obtaining additional information about the projected 3D models on the tabletop screen, such as a cross-sectional image or an alternative rendering (see Figure 4.10). These techniques are very useful for architectural tasks. Architects often need to investigate cross-sectional images on arbitrary planes from a variety of viewpoints. However, it is difficult to define plane equations and viewpoints in 3D CAD software.

#### 4.3.2.1 Plane detection

With this system, the user can specify a plane equation and a viewpoint very easily and naturally. I emphasize the fact that the paper has no specific devices attached, such as tracking IR markers, which means that a user can use paper of any size and shape for the mobile pinhole window, which is a departure from other related work such as PaperLens and TnagibleWindows (*Spindler et al.*, 2009, 2012).

I specify the plane equation of the paper using depth data alone, namely the

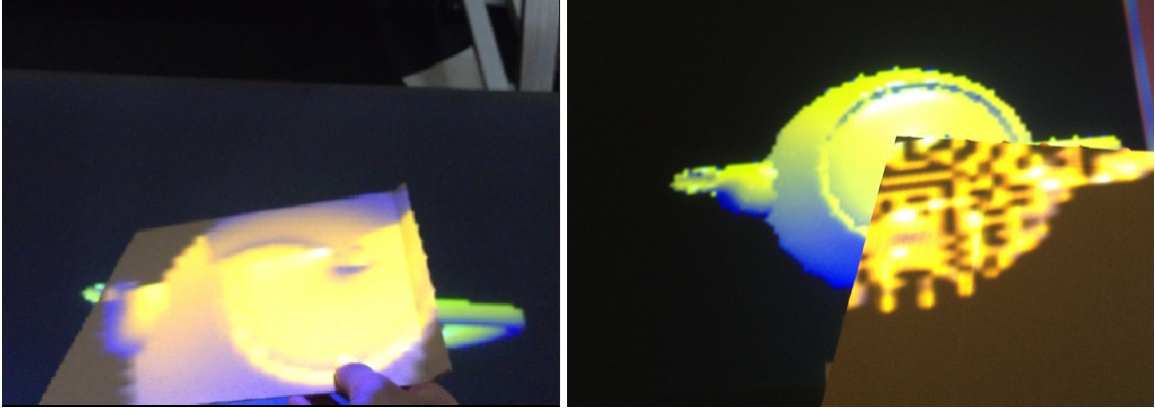


Figure 4.10: Left: cross-sectional image, Right: wire-frame rendering

distances between the paper surface and the Kinect camera. I prepare a bounding box in advance, enabling the Kinect to ignore those parts that are not needed, and the calibrated Kinect obtains depth information about the real objects inside the bounding box. Each pixel has associated depth data and is calculated in terms of a Kinect coordinate system such as a point cloud. To find the paper's plane equation, I apply the plane least-squares method to all points inside the bounding box. To omit the user's hand and arm and specify the plane equation more precisely, points whose distance from the result of the least-squares method exceeds a threshold are eliminated, with the remaining points being processed by the least-squares method again (see Figure 4.11).

After iterating this process several times (twice or three times is usually sufficient), I obtain an accurate paper equation  $z = ax + by + c$ . Note the plane equation is infinity. To make the paper screen act as a pinhole window, I need to set an off-axis projection to the plane in the 3D scene (see Figure 4.12). To set this, I find the position of the plane. However, it is hard to track the paper without tracking markers correctly. Therefore, we compute the translation on only vertical axis of the paper. The parameter is found by calculating an average of all points of the plane of the result from the iteration. The off-axis projection screen set on the plane is always larger than the paper to acquire a whole paper projection, such as cross-sectional image (see Figure 4.13, (a)). Therefore, the translations on  $x - y$  plane of the paper are not required. However, this method allows the projector to project a whole image on the paper and the tabletop surface. A overlap of two projections occurs on the tabletop projection (see Figure 4.13).



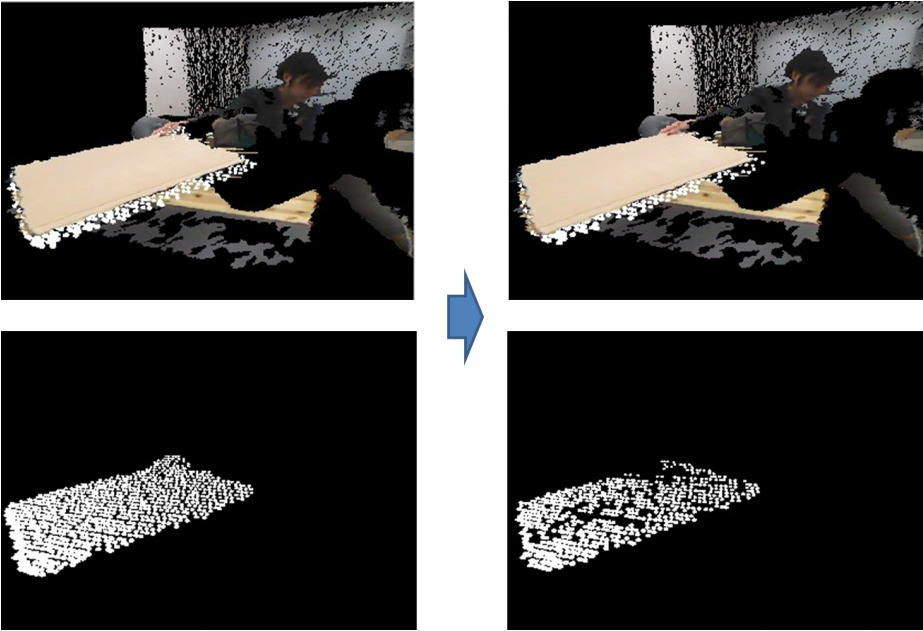


Figure 4.11: Iteration of the plane detection process

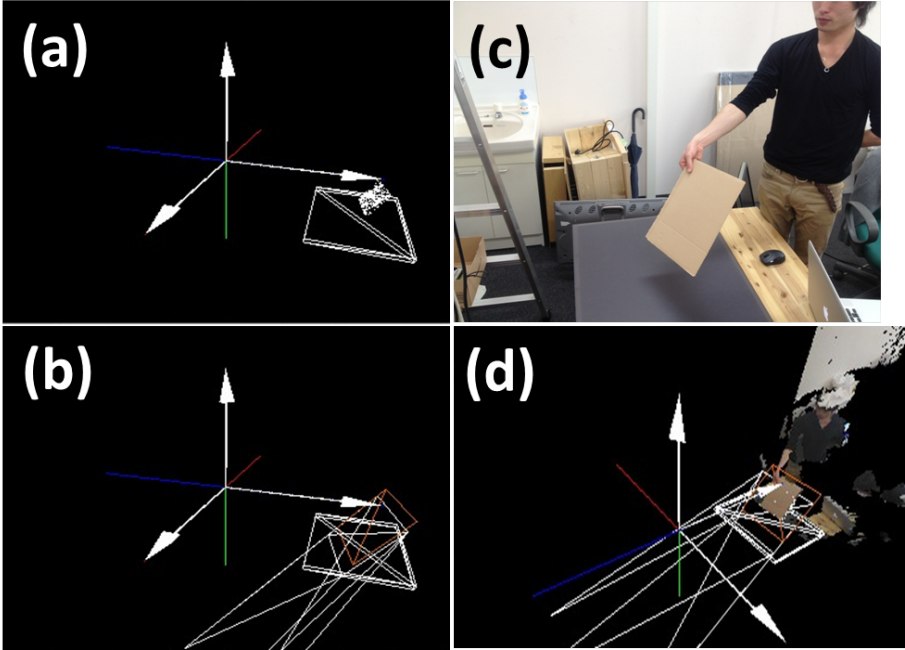


Figure 4.12: 3D reconstruction of a paper

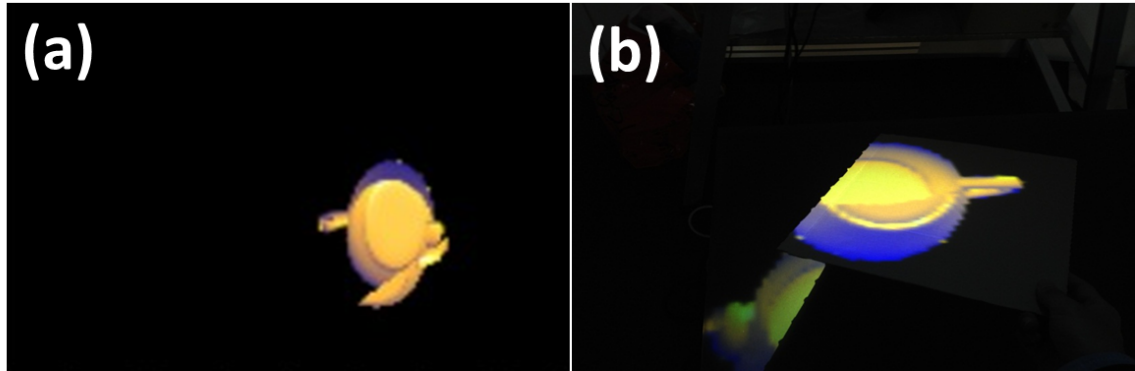


Figure 4.13: Left: a whole cross-sectional image, Right: the user's view

#### 4.3.2.2 Avoid overlapping

To avoid overlapping of the two projections on the tabletop surface, the region outside the projection of the paper should be black. The paper projection is usually projected on the objects above the tabletop, such as the paper and the user's arm, and the tabletop surface. However, the user is supposed to see the tabletop projection and the paper projection simultaneously. Therefore, the part projected on the tabletop screen of the paper projection should be omitted. One of the options how to omit them is to make the color data black. I prepare a binary image to project exclusively on the object's surface inside the bounding box (see Figure 4.14, (b)) and then, I subtract the black part in the binary image from the paper projection (see Figure 4.14).

First, all points of the objects's surface inside the bounding box are detected by the Kinect. If a 3D models (e.g., sphere model) is positioned to each point, the 3D model should be projected on the exact point by the projector for the paper projection. (see Figure 4.15 (a) (c)). To get such a binary image, white 3D models are set to all points in the 3D scene (see Figure 4.15, (b)). Then the projection is projected on only the object's surface inside the bounding box, such as the paper screen and the user's arm (see Figure 4.15, (d)). (Figure 4.15 (d)). The goal is to change all pixels  $(x_i, y_i)$  whose colors are black in the binary image black in the original paper projection as well (see Figure 4.14). However, it is quite heavy process to get and set each color data in these images. Therefore, I compute this process using 3D models in the 3D scene.

I get each pixel's color data in the binary image and if it is black ( $=((x_i, y_i))$ ), I set a small black 3D model (e.g., rectangle) at  $(x_i, y_i, 0)$ . Finally, I can acquire a black plane has holes at approximate the white part in the binary image in the 3D scene.



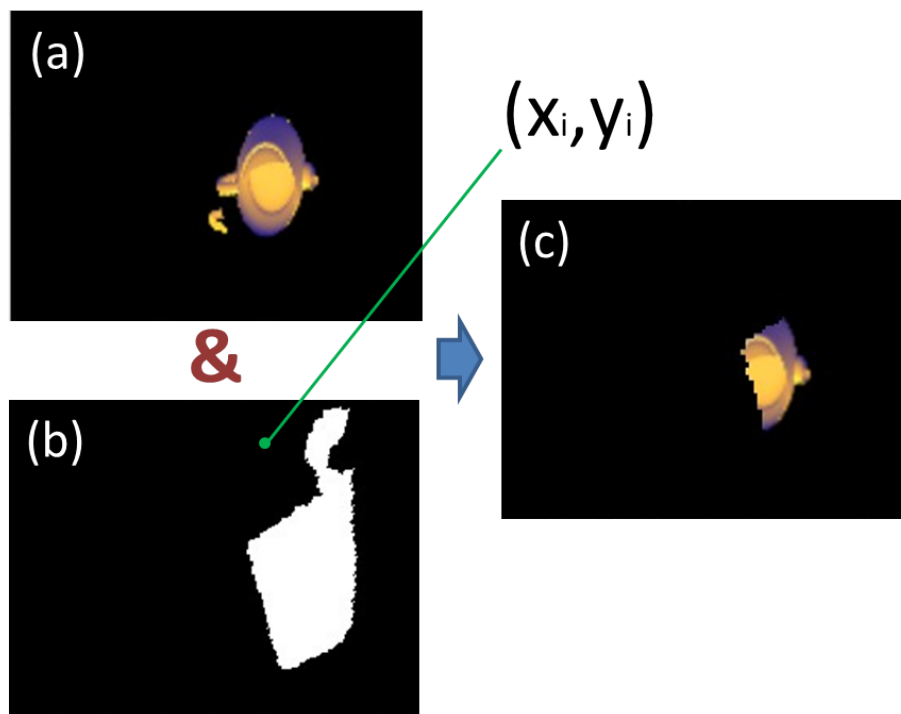


Figure 4.14: (a):a paper projection before computing (b):a binary image of the objects above the surface (c):a result image by subtraction (b) from (a)

This black plane acts a filter in the 3D scene. Besides, the original paper projection image is set parallel to the filter. Then, a camera is set in front of the filter and captures the original paper projection image through the filter in the 3D scene (see Figure 4.16). With it, I can project only on the paper. Furthermore, this technique allows the user to use papers with any shape.

Note that 3D models are position twice in this process, first to the point cloud and second to the point  $(x_i, y_i, o)$  in the 3D scene. The quality of the filter (especially the hole) depends on the size and the number of the 3D models. I need to adjust them to make the contour more accurate.

### 4.3.2.3 Correct perspective

My vision is that the system enables the user inspect the projected 3D models using a paper. The paper projection and the tabletop projection should be regarded as one image from the user's view point (see Figure 4.10, Left). The paper projection should be correct perspective image from the user's view point likewise the tabletop projection. Figure 4.17 (a) shows the 3D model and the cutting plane reconstructed in the 3D scene. Figure 4.17 (b) shows the correct perspective image of the paper

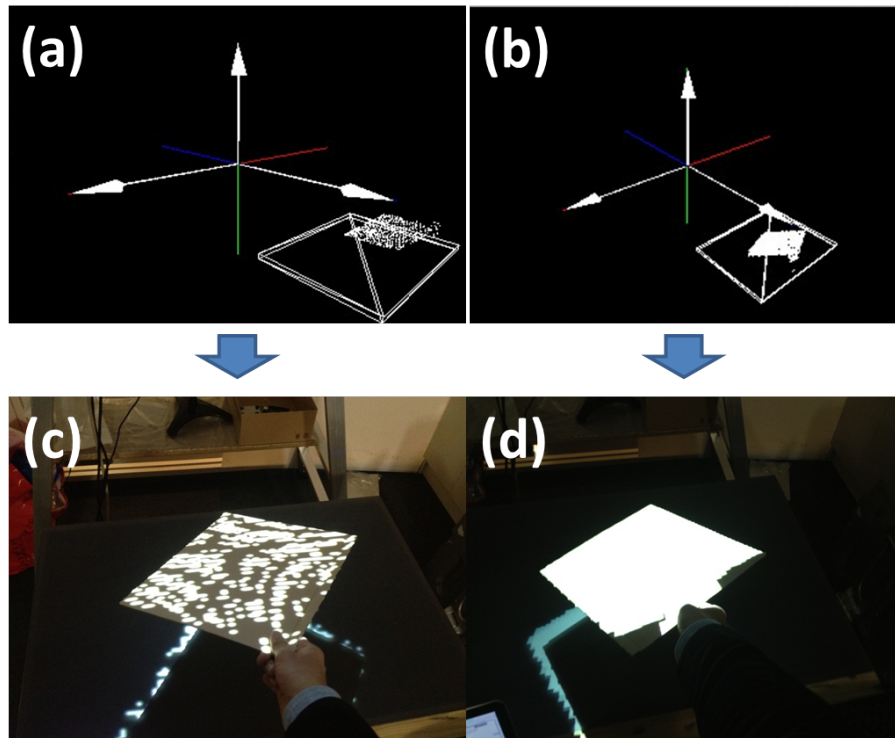


Figure 4.15: (a) points of the object's surface in the 3D scene (b) white 3D models are positioned to all points in (a) (c) the points are projected on the paper (d) the white models are projected on the paper

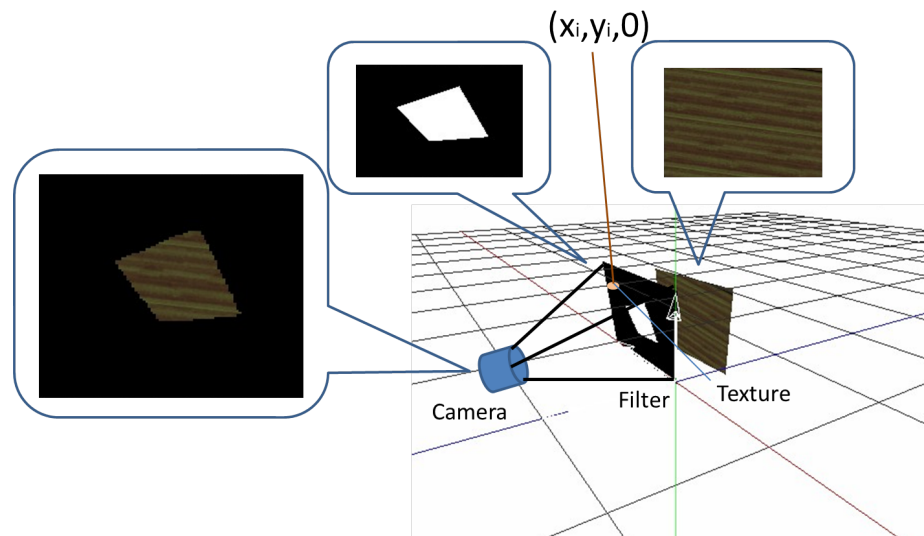


Figure 4.16: The filter, the texture image and the camera are set in the 3D scenen.

projection from the user's view point, which means if the image (b) is projected on the paper, the user can see the projection correctly. In tabletop projection case, the image (such as (b)) is projected on the screen directly because the projector and the screen set in parallel and fixed. However, in the mobile screen case, the paper and the projection is not always parallel. This is achieved by the calibration and setting the image (b) on the cutting plane (off-axis projection plane) (see Figure 4.17 (c)).

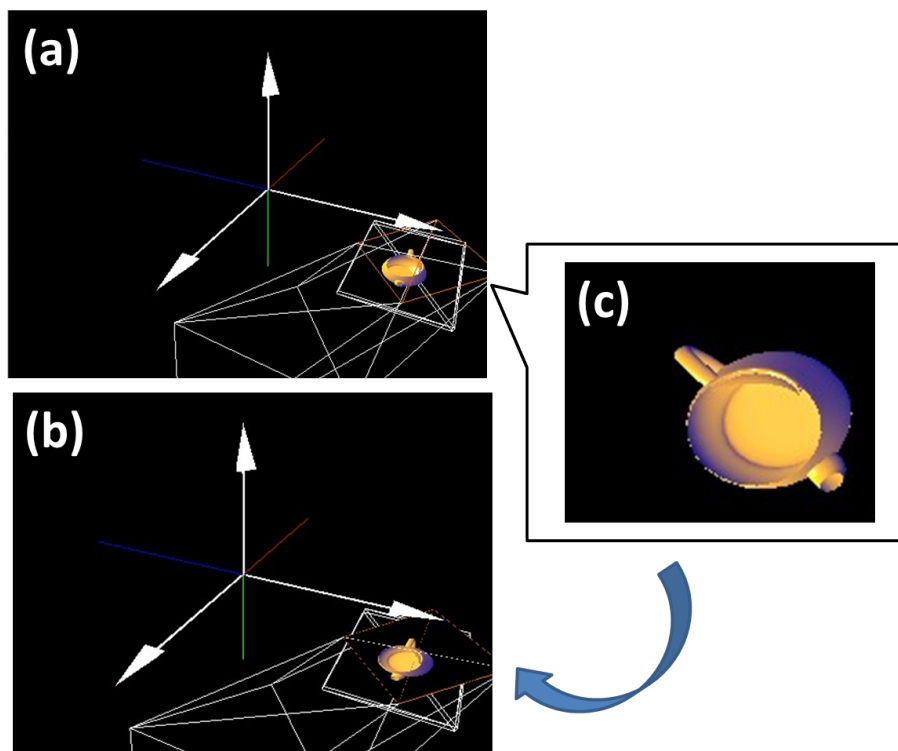


Figure 4.17: (a) the 3D object and the cutting plane (b) correct perspective cross-sectional image from the user's view point (c) setting image (b) on the cutting plane (off-axis projection plane)

### 4.4 Interaction techniques

I envision intuitive interactions with the 3D information in this system. As mentioned in Chapter I and II, I believe that the interaction in 3D space should be efficient for the manipulation of 3D objects. Therefore, I focus on the interaction techniques above the surface and architectural tasks. As mentioned in Chapter I, architects need to build an architectural model to present, inspect or discuss the building idea. Even though the user can intuitively inspect the 3D models projected on the tabletop screen and the paper by moving the paper or her/himself, I think the system should enable

the user to interact with the 3D information because the user sometimes needs manipulate, add or delete 3D models. I surveyed what kind of interactions are required in the tasks.

- Copy and paste
- Delete
- Drag
- Manipulate
- Light effect
- Texture
- Cross-sectional image
- Alternative rendering

*Copy and paste, delete, drag and manipulation* are general tasks in any other fields as well as the field of architecture. In architectural tasks, imagine that users deal with 3D models such as furniture in the 3D building models such as rooms. When they inspect the architectural models, they also adjust alignments of the furniture. Which means these interaction techniques are absolutely required for the architectural tasks. Besides, as mentioned in Chapter I, light effect and textures are also important points in architectural tasks. In this section, details of the interaction techniques and methods in the proposed system are explained.

#### 4.4.1 Interaction techniques using a paper

I implemented some of the interaction techniques that utilize a handheld paper. I note that when the user uses the paper to obtain additional information, he/she will be focusing on the 3D model and may wish to perform manipulations and the paper projection shows details about the 3D model. Given that I already know the equation of the paper (Section 4.3.2), I consider that interaction using the paper should be very helpful for users.

Figure 4.18 shows a basic process of the interaction techniques in this system, which are 'pick up' and 'release'. These interaction methods enable the user to naturally manipulate the 3D model by rotation and translation, which the user can perform by tilting and moving the paper (see Figure 4.19).

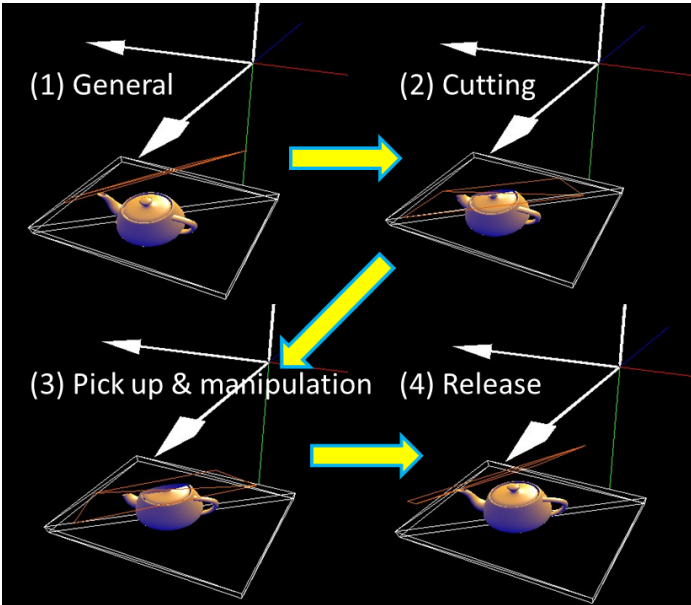


Figure 4.18: A basic interaction process: Pick up and Release

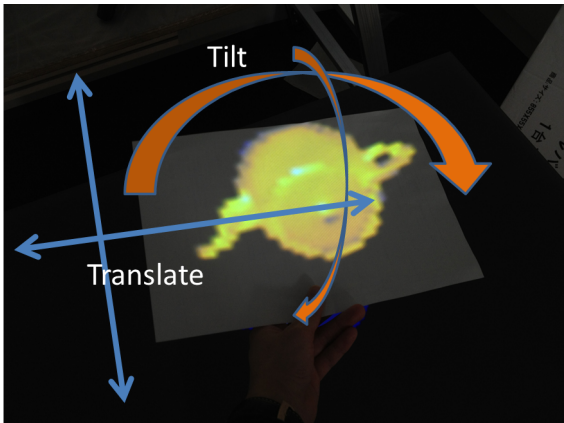


Figure 4.19: Manipulation of the 3D model using a paper

This process is based on the *drag* and *manipulation* tasks.

When the user selected (details of switching are described in Section 4.4.3) a particular object in the 3D scene by holding the paper over the projected object, the paper acted as a pinhole window. Once the user picked up the model, the paper starts acting as a showcase. The user can see the paper projection as if there was the object inside the showcase (see Figure 4.20).

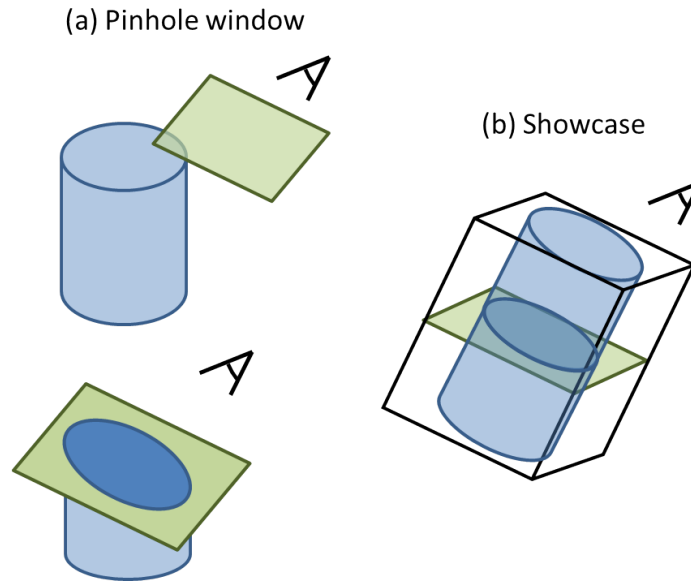


Figure 4.20: (a) Pinhole window (b) Showcase

Even though the user can only rotate the 3D model while the paper's equation remains detectable (if the paper is vertical to the Kinect IR camera, for example, its plane equation is not computable), I consider that users will usually need to manipulate the models only slightly, particularly in architecture. This limitation is mentioned in Chapter VI. The user can also copy and delete the particular models.

Currently, the user can select a particular 3D model to manipulate with this technique by holding the paper over the projected 3D model on the tabletop surface. However, sometimes the paper will be held over several 3D models. In such cases, I will need to consider other selection methods in the future work. One idea is a finger-pointing system, whereby the 3D positions of the user's fingers are tracked and the user can point to a particular 3D object. This idea is described in Chapter VII.

#### 4.4.2 Light effect

As mentioned in Chapter I and III, the examining the light effect is required in architecture tasks. Besides, they often simulate the light effect to the architectural

models.

I envision that the system enable the user to simulate the light effect naturally and intuitively. The parameter of the light source is only 3D position in the 3D scene. I noticed the Kinect tracks the user's hand position in this system. That is why I implemented the system that the user can change the position of the light source by moving her/him hand as if s/he was holding the light source in real. However, the Kinect detects any objects inside the interaction space so the hand should not be inside the space if the user wants to use the cross-section plane system. The idea how to change the mode is mentioned in the following section.

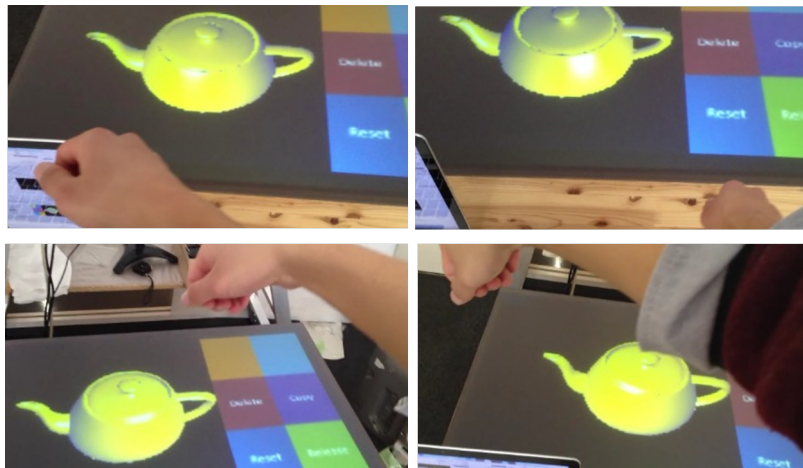


Figure 4.21: By moving the user's hand, the user can move the light source in the 3D scene as if the user was holding the light source.

### 4.4.3 Menu space

#### 4.4.3.1 Idea

This system needs to be able to enable the user to change the mode with increase of the number of abilities. For now, this system supports two main application, such as alternative rendering and manipulation. Each main mode also has several options. Figure 4.22 shows a whole modes in this proposed system.

There are several points I need to consider for creating the menu system. First of all, Since the user needs to be able to change the rendering or turn on the 'pick up' mode, for instance, while holding the paper over the 3D model by considering the interaction processes, the user should be able to change the mode while manipulating the 3D model, especially, holding the paper (see Figure 4.18). Second of all, once the user is satisfied with the position of the 3D model while manipulating, the user

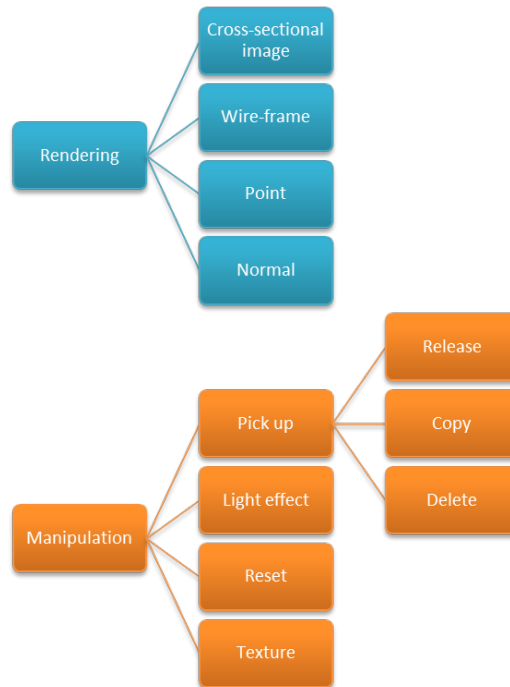


Figure 4.22: Menu options in this system

may want to turn on 'release' switch, for instance, without putting off the balance. Which is why I prefer a 3D interaction to a touch interaction for the selection in menu. Lastly, this menu display should be always visible for the user.

#### 4.4.3.2 Implementation

I implement a menu space on the tabletop surface to allows the user to change the application mode while using this system.

I decide to make the menu space on right or left side on the tabletop surface (see Figure 4.23).

The position of the menu space should depend on which arm the user holds a paper, if the user hold a paper on left arm, for example, the menu space is positioned on right side. This menu works in three dimension above the surface to enable the user to change the mode without bending down. Figure 4.24 shows the menu design. The cursor is the user's hand which does not hold the paper. The position of the user's hand is tracked by the Kinect in the 3D scene. I prepare two bounding boxes for each menu and stuck them such Figure 4.25.

The user can turn on the switch by holding the hand up higher. However, when the user is holding the hand over the menu screen, the user cannot see the icon well due to the occlusion. To solve this problem, another icon is projected on the user's



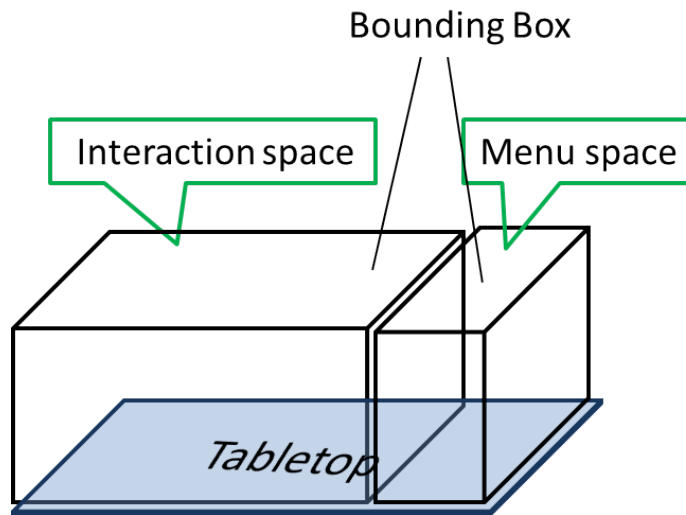


Figure 4.23: Idea of menu and interaction space

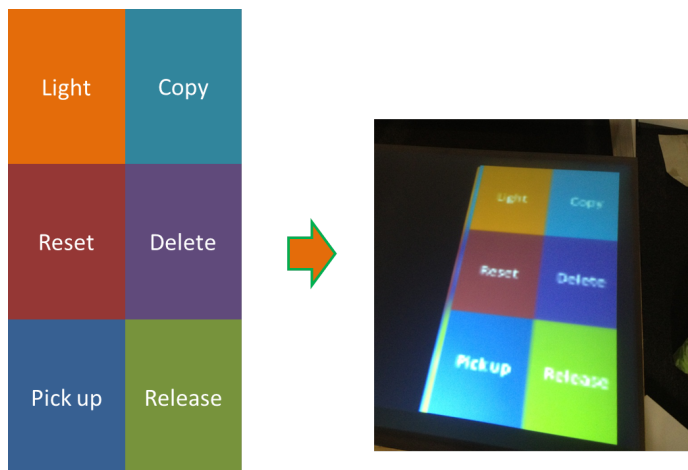


Figure 4.24: General menu display

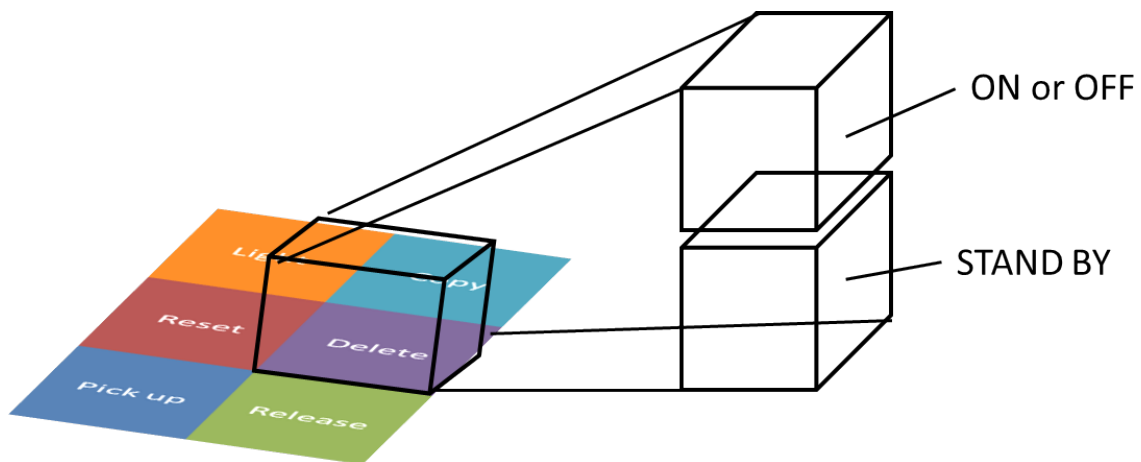


Figure 4.25: Two bounding box for the switch

hand (see Figure 4.26). A primitive is drawn in the 3D scene and the icon texture glued onto the primitive. The position of the primitive is synchronized with the position of the user's hand and set parallel on the tabletop surface in the 3D scene. When the user's hand is inside a bounding box, the particular icon is projected on the hand, for instance, when the hand is inside the bounding box of 'pick up', the icon for 'pick up' appears on the hand (Figure 4.26).

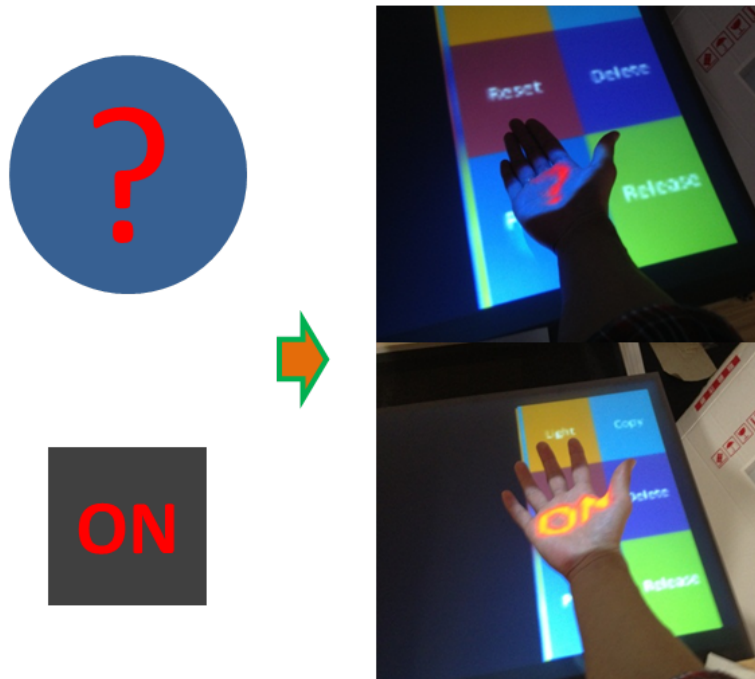


Figure 4.26: Feedback icons





## Experiment and Evaluation

### 5.1 Experiment

#### 5.1.1 Accuracy of the paper projection

The accuracy of the projection mainly depends on the calibration between the Kinect and the projector. There is usually a slight pixel error in the checkerboard calibration process (cf., *Zhang* (1998)). The error tends to occur due to the positioning the devices, which means the projector and the Kinect should be facing same direction and sharing similar image. In this system, however, the projector and the Kinect are positioned as shown in Figure 4.2, the Kinect is facing the user to track his/her body and the projector is facing the tabletop surface. Figure 5.1 shows how much different error between pattern (a) and (b). The vertical axis represents the average pixel error and the calibration process was conducted ten times.

Besides, as mentioned in Section 4.3.2, to avoid overlapping of the two projections, the paper projection and tabletop projection, the out side of the objects in the paper projection is computed to black. By this process, indeed the overlap is reduced. However, the solution cannot solve the overlap problem completely. The accuracy depends on a process that some 3D objects (primitives) are set in the 3D scene.

This system used a short throw projector for the hung projector to be able to project an image on the paper from short distance. The focal length is fixed in this system so the paper projection becomes blur, when the paper screen is positioned close to the projector.

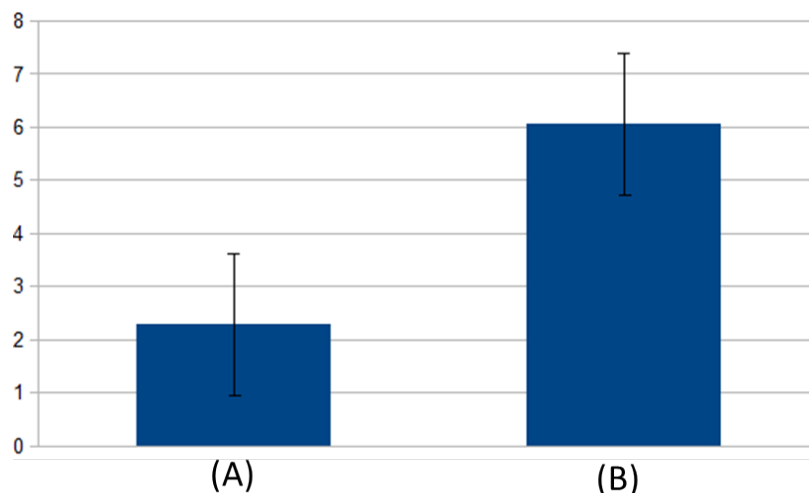


Figure 5.1: Average of the pixel error in cases of; (a) The Kinect and projector are facing same direction. (b) The Kinect is facing a user and the projector are facing the tabletop surface (same position as the proposed system)

### 5.1.2 Application

In proposed interaction techniques using a paper, the paper equation needs to be founded. As mentioned in Section 4.4, the Kinect cannot sometimes detect the paper's equation correctly, for instance when the paper is set vertically to the Kinect image plane ( $x - y$  plane)(see Figure5.2). A process of detecting the paper equation is an iterate of applying the least square method and elimination points exceeded the threshold (shown in Figure 4.11). I experimented how range the paper's equation is correctly detectable. In this proposed system, however, the paper is not supposed to be set in vertical on the tabletop surface by considering the paper projection. That is why I surveyed the detectable and projectable (means the paper is able to work as the projection for the user) range and the usable range is found by combining the two result. The detectable range is experimented by trying to translate and rotate on each axis. Figure 5.3 shows result of each axis. Note that read, green and blue axis represents  $x$ ,  $y$  and  $z$  axis. (a) shows the rotation about  $x$  axis. The rotation about  $x$  axis is robustly detected. (b) shows a translation of  $y$  axis in the 3D scene. Even though the paper plane is in parallel with the tabletop surface, the paper can be detected. However, when the height ( $y$  axis) of the paper is the same as the Kinect height, the paper is not detectable (see Figure 5.2, Left). (c) and (d) shows the rotation about  $z$  axis. Rotations more than about 50 degree cannot be detected (see Figure 5.2, Right). Besides, because the process of the detection plane utilize equation  $z = ax + by + c$  which is infinity plane, the rotation about the vertical axis

of the plane cannot be detected.

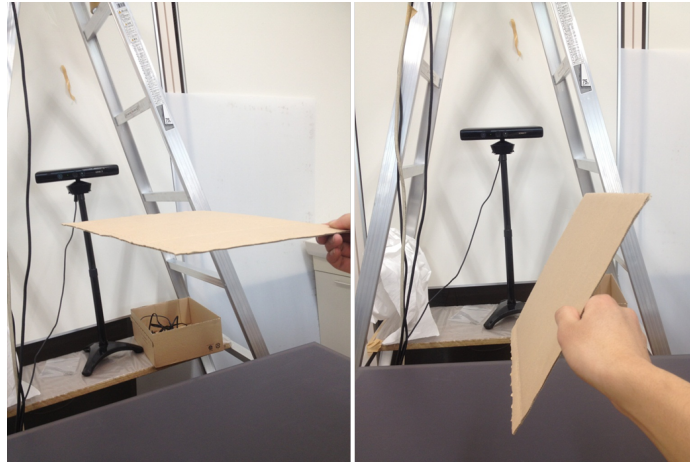


Figure 5.2: Example of a paper positions and orientations which the paper cannot be detected by the Kinect

I conducted the experiment of projectable range in the situation that the user's head is fixed because of the fact that the result depends on a relation among the user's head position, the paper equation and the projector orientation. By assuming that the user uses the paper close to him/her in this system, in this experiment, the user's head is positioned above and in front of the paper. Figure 5.4 shows pictures taken from the user's view point. (a) shows a rotation about x axis. As the paper tilts, the projection that the user can see decrease, even though the paper is robustly detected (see Figure 5.3,(a)). (b) shows a translation of y axis. In this case, the projection is always facing up so the user can see the projection clearly. however, as the paper is moved higher, the projection has holes (see Figure 5.4,(a), (b)). these holes are attributed to the process of avoiding the overlap. (c) and (d) show a rotation about z axis. As the paper is tilted, the projection is distorted.

## 5.2 Evaluation

Although still being in an early stage of implementation, experiences with this system were encouraging. Especially feedback by a professional architect are required and the system cannot be evaluated without this. One professional architect (24 years old, male) and one student (26 years old, male) are invited to the user experience. After brief demonstration, they are immediately able to successfully interact with the system. they liked the concept of the viewing and the paper projection. Especially the idea that the cross-sectional plane and image is defined easily and naturally is

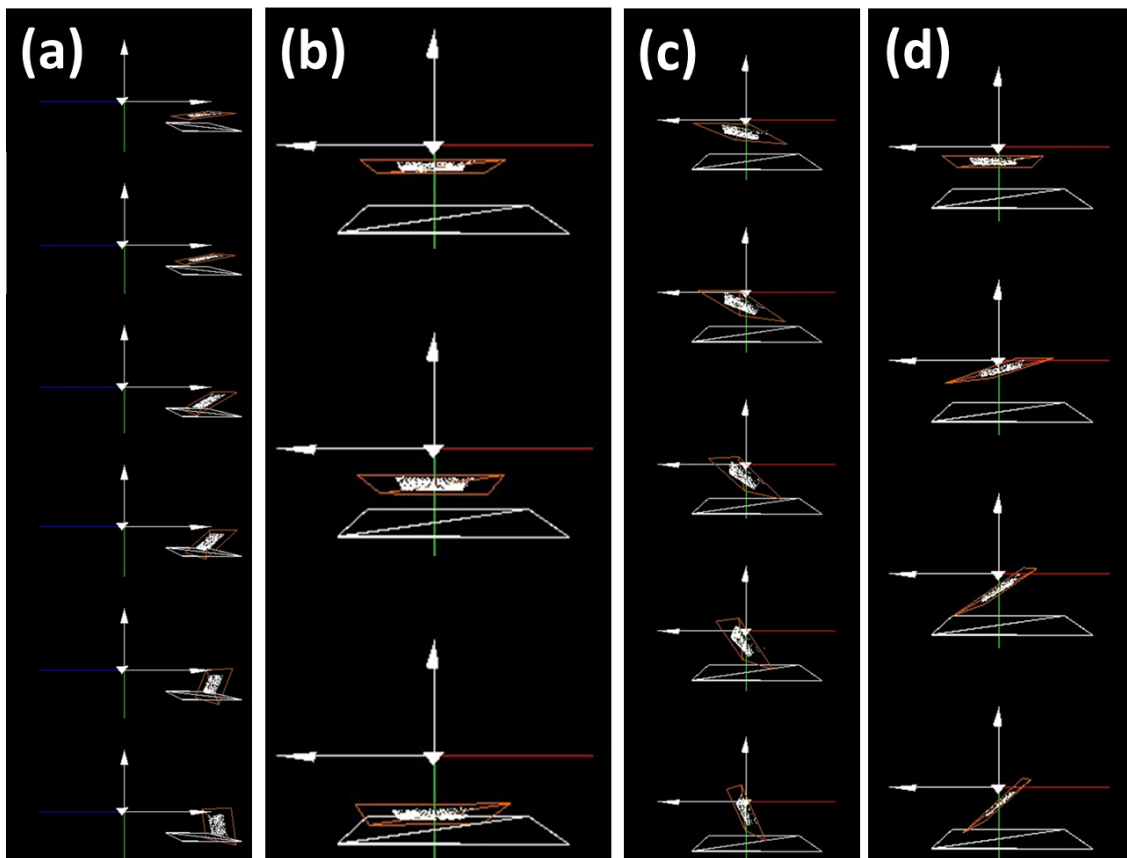


Figure 5.3: The reconstructed paper in the 3D scene: (a) Rotation about x-axis (b)Translations on y axis (c), (d) Rotations about z-axis

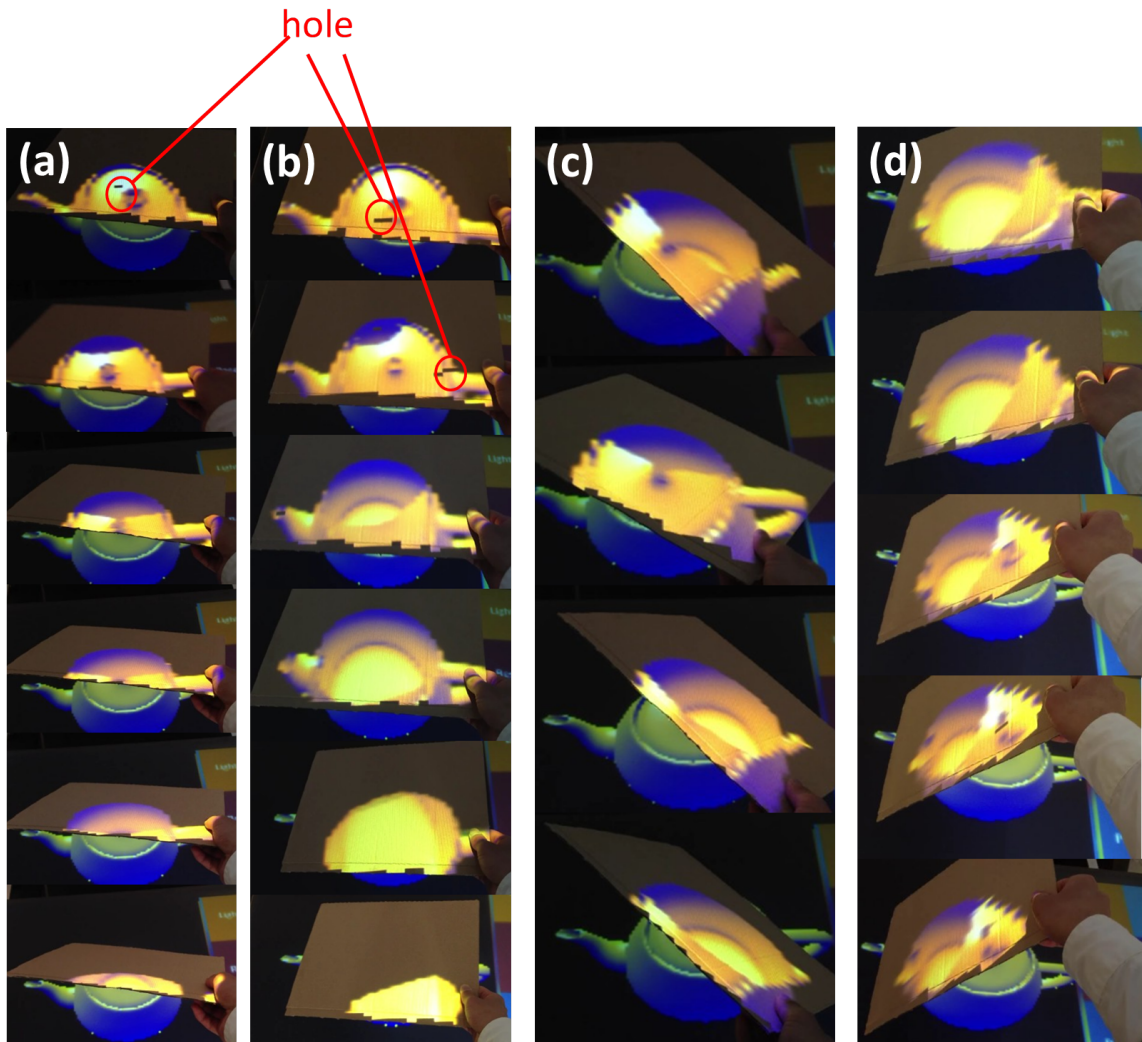


Figure 5.4: The pictures taken from the user's view point: (a) Rotation about x-axis (b)Translations on y axis (c), (d) Rotations about z-axis



evaluated highly. The cross-sectional image is one of the most important things of the 3D modeling in the field of architecture. However, he mainly complained about the latency. I noticed that sometimes the tracking and the projection cannot catch up the movement. Besides, the user cannot rotate the model on the vertical axis of the tabletop surface during looking the paper projection. In architectural tasks, this is a crucial issue because they often need the rotation when they arrange the furniture models, for instance. However, the user can rotate about the vertical axis if the user set the paper on vertical on the tabletop surface. In this case, although the user cannot see the paper projection, the user can see the tabletop projection screen. Besides, they often need to inspect particular models as well. That is why the rotations about x and y axis are also demanded. The users said that the paper projection is not clear and sometimes blur. This is attributed to the fixed focus length of the projector and the wide projection. In this system, although the focus length is fixed, the paper screen is moved. Contrary to the tabletop projection, the size of the paper projection is wide. The users often noticed the difference between the quality of these projections. They mentioned that the idea of the menu system is new and easy. They are surprised the fact that even though they lift the paper or their hand, the size of the projection which is the paper projection or the feedback icon is not change, which means the size is correctly suited to the tabletop projection and looked correctly from the user's viewpoint.

## 6.1 Discussion

### 6.1.1 Limitations of the proposed system

In this section, I describe some limitations of this work. The limitations are mainly about an accuracy and applications. Some limitations occur due to the fact that the system does not require the user to wear any tracking device and a particular paper. I would like to discuss the significances of the limitations. The main limitations are listed below.

- The range that the paper is usable as a projection
- Only one user can use this system at a time
- Simulating a monocular eye movement

As experimented in Section 5.1, this system has a limitation about the paper detection. Especially about the interaction techniques using the paper (presented in Section 4.4), a limitation of the manipulation depends on the limitation of the detection. The user can manipulate the 3D model as far as the plan is detected correctly, which means the user cannot rotate the 3D model more than 180 degree, for example. However, in architectural tasks, I assume that the user only need to slightly manipulate the model in this system, such as a furniture model. However, in this interaction technique, the user needs to position the paper projection to vertical on the tabletop surface if the user want to rotate the model about the vertical axis on the tabletop surface. Which means the user cannot rotate them during looking at the paper projection. Figure 5.2 shows exmample positions that the paper is not

detectable. However, in Figure 5.2 (left) case, the paper is positioned too high to be looked from the user. The user is supposed to use the paper above the tabletop up to 30 cm. That is why this is rare situation to use the paper such as Figure 5.2 (left). In addition, I noticed that the user does not often tilt the paper more than 45 degree (such as Figure 5.2, Right) because the user cannot see the projection. In architecture tasks, the parallel cross-section is more often required than the vertical one. Even though the detectable range is wide in this system (refer Figure 5.3), the projectable range is quite limited (refer Figure 5.4). Which is why the usable range is quite limited. The user cannot tilt the paper more than about 45 degree because of the distortion of the projection. When the paper is held at the same height as the Kinect (see Figure 5.2, Left), the paper could not be detected. However, the problem is easily solved by positioning the Kinect above the interaction space or at the user's view height. The user can see the cross-sectional image of the model in parallel with the tabletop screen (see Figure 5.4 (b)). This is very useful when architects want to work with individual level floor at a time. The user can see each floor plan by just shifting the paper intuitively.

Since the tabletop projection provides a correct perspective image for the particular user, the system is used by only one user at a time although the system uses one big screen. One of advantages utilizing tabletop surface is to enable multiple users to share the screen and interact with the information at a time. That is why I think that the system should be able to be used by multiple users. This system has a capability of the implementation. The idea is explained in Section 7.2.

Although the system provides the correct perspective image and to enable the user to see the projected 3D model as if the model existed on the surface, this is simulated by monocular eye so the user cannot obtain the immersive 3D illusion more than using stereoscopic images with shutter glasses. However, I focused on the aspect that the system does not require the user to wear any device. However, this program can be applied to stereoscopic application as well.

### 6.1.2 Proposed system and applications

AS mentioned in Section 5.2, I obtained feedback from the user experience. First, they liked the concept of the paper projection, tabletop projection and menu system idea. Especially the fact the user can define the cross-section plane is evaluated highly. Although they are not used to interacting in 3D space, the user learned soon and actually interacted with the menu system. Besides, the user were inspired

that the user can move the light source by just moving the hand as if the user were holding the light source. Since the projections are correct perspective images, the user can naturally confirm how the model is looked with the light source from the user's viewpoint.

However, they also mentioned some drawback of the system, First, the complained about the latency of the projection. The architect user said '*The system can actually never be used at an architectural office unless the system worked faster.*'. Besides, the system sometimes catch up the user's movement. Currently, the frame per second (fps) is approximately 3. This is absolutely slow to enable the intuitive interactions. The fact the user cannot walk around the tabletop because of positions of the devices. In architecture, it is better to be able to look the model from any angle. Finally, he suggested additional interaction technique. He said that the user should be able to manipulate a particular model during the user cutting the building model. The proposed techniques is using the paper, which means interaction combing the cross-sectional image and manipulation is not supported in this system so far.



## Conclusion and Future Work

This chapter summarizes the thesis and mentions the future work.

### 7.1 Conclusion

We have presented an interactive interface that has combined a tabletop projection with a mobile paper projection. The paper projection acts as a pinhole window to show additional 3D information and to enable the user to manipulate particular 3D models intuitively and naturally. Each projection shows correct perspective image from the user's viewpoint to enable the user obtain 3D illusion. This system achieved this capability without requiring the paper to have any type of tracking marker. In addition, this thesis has outlined some application scenarios and the limitations. They are especially focused on the task in the field of architecture.

### 7.2 Future work

We have presented the basic aspects of our proposed system and consider that it can be improved further, particularly the interaction and application parts. Our first goal is that this system should be useful in an architectural work environment. To achieve this purpose, we still need to consider several issues.

#### 7.2.1 Multiple users

In architecture, particularly in a presentation, there are situations involves discussion details of the architectural models and actually modifying them by several users. Currently, our system is available for only one user at a time. To make the system available for multiple users, we need to consider only the positions of the Kinect and

the hung projector because the projection should cover the whole of the tabletop surface and the Kinect camera should cover all users. The Kinect can detect up to six users' head positions. However, to make this possible, the tabletop projection would have to be a normal projection rather than a correct-perspective projection. The vision is that the users can share the tabletop projection and interact with the 3D information using each paper that the user hands. The process idea is explained below. First, the Kinect recognizes the number of the users and tracks each user's body. As explained in Chapter IV, the real objects inside the interaction space are reconstructed in the 3D scene by the Kinect. In the proposed system, only one paper equation is found and reconstructed. For the environment for multiple users, the system has to recognize each user's paper. The detection of the papers can be achieved by a clustering algorithm. By assuming the fact that each paper does not interfere each other in this system and each paper is mapped on the depth camera separately, the Kinect acquires a binary image of the depth image. The blobs (white parts in the binary image) are labeled (clustering and put number) using OpenCV library, then the detecting process of the plane equation (described in Section 4.3.2) is applied each cluster. Since the users are supposed to use the paper in front of him/her, each pair (a user and a paper) are found easily.

### 7.2.2 Flexible paper

A capability that would make the system more useful would be to enable projections on flexible paper. Currently, the system supports only projection on a plane, which means that the paper has to be rigid. Using flexible paper to show cross-sectional images on an arbitrary curve would make our application scenarios more productive. In this proposed system, the cross-sectional image is acquired by setting a off-axis projection plain as a window in the 3D scene (see Chapter II). This off-axis projection plane acts the cutting plain as well. To acquire a curve cross-section, the shape and the curve of the paper should be detected first. This can be achieved by the Kinect. The paper can be reconstructed by applying small primitives to the point cloud of the paper in the 3D scene. By regarding each small primitive as a small cutting plane and an off-axis projection plane, the process mentioned in Section 4.3.2 can be applied to the primitives. This is one idea of flexible paper projection.

### 7.2.3 Interaction techniques

Currently, the user can select a particular model by holding the paper over the projected model. However, if there are several models on the paper projection, the user cannot choose the particular model. Even though we presented menu space in Section 4.4.3 but a selection of the model using the menu system is not sufficient. We envision that the user can select the model naturally. One of the ideas is an implementation of pointing program by detecting the user's finger.





## Bibliography

- Agrawala, M., A. C. Beers, I. McDowall, B. Fröhlich, M. Bolas, and P. Hanrahan (1997), The two-user responsive workbench: support for collaboration through individual views of a shared space, in *Proc. SIGGRAPH '97*, pp. 327–332.
- Azuma, R., Y. Baillet, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre (2001), Recent advances in augmented reality, in *IEEE Comput. Graph. Appl.*, pp. 34–47.
- Benko, H., and A. D. Wilson (2009), Depthtouch: Using depth-sensing camera to enable freehand interactions on and above the interactive surface, in *Proc. ITS '09*, pp. 11–14.
- Benko, H., R. Jota, and A. Wilson (2012), Miragetable: freehand interaction on a projected augmented reality tabletop, in *Proc. CHI '12*, pp. 199–208.
- Brox, T., A. Bruhn, N. Papenberg, and J. Weickert (2004), High Accuracy Optical Flow Estimation Based on a Theory for Warping, pp. 25–36.
- Cruz-Neira, C., D. J. Sandin, and T. A. DeFanti (1993), Surround-screen projection-based virtual reality: the design and implementation of the cave, in *Proc. SIGGRAPH '93*, pp. 135–142.
- Dippon, A., and G. Klinker (2011), Kinecttouch: accuracy test for a very low-cost 2.5d multitouch tracking system, in *Proc. ITS '11*, pp. 49–52.
- Epson (2013), Epson eb-435w, <http://www.epson.co.uk/gb/en/viewcon/corporatesite/products/mainunits/overview/11106>.
- Francone, J., and L. Nigay (2011), Using the user's point of view for interaction on mobile devices, in *23rd French Speaking Conference on Human-Computer Interaction*, pp. 4:1–4:8.
- Grossman, T., and D. Wigdor (2007), Going deeper: a taxonomy of 3d on the tabletop., in *Proc. IEEE '07*, pp. 137–144.

## Bibliography

---

- Hachet, M., B. Bossavit, A. Cohé, and J.-B. de la Rivière (2011), Toucheo: multitouch and stereo combined in a seamless workspace, in *Proc. UIST '11*, pp. 587–592.
- Hancock, M., M. Nacenta, C. Gutwin, and S. Carpendale (2009), The effects of changing projection geometry on the interpretation of 3d orientation on tabletops, in *Proc. ITS '09*, pp. 157–164.
- Harrison, C., H. Benko, and A. D. Wilson (2011), Omnitouch: wearable multitouch interaction everywhere, in *Proc. UIST '11*, pp. 441–450.
- Hilliges, O., S. Izadi, A. D. Wilson, S. Hodges, A. Garcia-Mendoza, and A. Butz (2009), Interactions in the air: adding further depth to interactive tabletops, in *Proc. UIST '09*.
- Hilliges, O., D. Kim, S. Izadi, M. Weiss, and A. Wilson (2012), Holodesk: direct 3d interactions with a situated see-through display, in *Proc. CHI '12*, pp. 2421–2430.
- Hirota, K., and Y. Saeki (2007), Cross-section projector: Interactive and intuitive presentation of 3d volume data using a handheld screen, in *Proc. 3DUI '07. IEEE Symposium on*.
- HITACHI (2013), Hitachi cp-aw100n, <http://www.hitachidigitalmedia.com/product.do?actionName=showProductAction&pt=6&pg=72&proid=683&language=en&country=GB>.
- Holman, D., R. Vertegaal, M. Altosaar, N. Troje, and D. Johns (2005), Paper windows: interaction techniques for digital paper, in *Proc. CHI '05*, pp. 591–599.
- iPad (2013), Apple-ipad, <http://www.apple.com/ipad/>.
- Izadi, S., et al. (2011), Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera, in *Proc. UIST '11*, pp. 559–568.
- Kim, S. W., A. Treskunov, and S. Marti (2011), Drive: Directly reaching into virtual environment with bare hand manipulation behind mobile display, in *Proc. 3DUI'11 IEEE Symposium on*, pp. 107–108.
- Kinect (2013), Kinect for windows, <http://www.microsoft.com/en-us/kinectforwindows/>.
- Kitamura, Y., T. Konishi, S. Yamamoto, and F. Kishino (2001), Interactive stereoscopic display for three or more users, in *Proc. SIGGRAPH '01*, pp. 231–240.
- Kruger, W., C.-A. Bohn, B. Frohlich, H. Schuth, W. Strauss, and G. Wesche (1995), The responsive workbench: a virtual work environment, pp. 42–48.
- Lee, M., R. Green, and M. Billinghurst (2008), 3d natural hand interaction for ar applications, in *Proc. IVCNZ'08*, pp. 1–6.

- McKenna, M. (1992), Interactive viewpoint control and three-dimensional operations, in *Proc. I3D '92*, pp. 53–56.
- OpenCV (2013), Opencv library, <http://opencv.org/>.
- OpenGL (2013), Opengl library, <http://www.opengl.org/>.
- OpenNI (2013), Openni library, <http://www.openni.org/>.
- Papagiannakis, G., G. Singh, and N. Magnenat-Thalmann (2008), A survey of mobile and wireless technologies for augmented reality systems., in *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, pp. 221–222.
- Spindler, M., S. Stellmach, and R. Dachsel (2009), Paperlens: advanced magic lens interaction above the tabletop, in *Proc. ITS '09*, pp. 69–76.
- Spindler, M., W. Büschel, and R. Dachsel (2012), Use your head: tangible windows for 3d information spaces in a tabletop environment, in *Proc. ITS '12*, pp. 245–254.
- Ugur, B., A. Vahit sahin, and I. Baris Fidaner (2009), Off-axis stereo projection and head tracking for a horizontal display, in *Proc. ACE '09*.
- vvvv (2013), vvvv - a multipurpose toolkit, <http://vvvv.org/>.
- Ware, C., K. Arthur, and K. S. Booth (1993), Fish tank virtual reality, in *Proc. CHI '93*, pp. 37–42.
- Wayne Arthur, K. (1993), 3d task performance using head-coupled stereo displays.
- Wilson, A. D. (2007), Depth-sensing video cameras for 3d, in *Proc. ITS '07*, pp. 157–166.
- Wilson, A. D., and H. Benko (2010), Combining multiple depth cameras and projectors for interactions on, above and between surfaces, in *Proc. UIST '10*, pp. 273–282.
- Ye, G., A. State, and H. Fuchs (2010), A practical multi-viewer tabletop autostereoscopic display, in *Proc. ISMAR '10*, pp. 147–156.
- Zhang, Z. (1998), A flexible new technique for camera, pp. 1330–1334, Technical Report MSR-TR-98-71.



## **International Conferences**

1. Yusuke Takeuchi, and Masanori Sugimoto. 2012 An Immersive Surface for 3D Interactions *Proceedings of the 2012 ACM International Conference on Interactive tabletops and surfaces (ITS'12) Posters*, 4 pages.
2. Yusuke Takeuchi, and Masanori Sugimoto. 2013. Immersive Magic Mobile Window *In Proceedings of the 2013 ACM International Workshop on CHI (CHI'13)*, 6 pages, to be submitted