

Projector-Based Interactive Visual Processing

LI, Zhaorong

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Computer Science and Engineering

The Chinese University of Hong Kong

June 2011

UMI Number: 3497753

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent on the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 3497753

Copyright 2012 by ProQuest LLC.

All rights reserved. This edition of the work is protected against unauthorized copying under Title 17, United States Code.



ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

Thesis/Assessment Committee

Professor WONG Tien Tsin (Chair)

Professor WONG Kin Hong (Thesis Supervisor)

Professor SUN Hanqiu (Committee Member)

Professor WONG Kenneth K. Y. (External Examiner)

Abstract of thesis entitled:

Projector-Based Interactive Visual Processing

Submitted by LI Zhaorong

for the degree of Doctor of Philosophy

at The Chinese University of Hong Kong in April 2011

The recent trend of human-computer interaction technologies has revealed the potential of the projector as an powerful interaction tool. More than a pure display tool, a projector has great strength that can change largely the way a traditional user interface works. Although some possibilities have been investigated in previous work, certain applications and approaches deserve further studies. For example, 1) Projection showing 3D information: viewing 3D models is usually achieved by projecting polarized light of different phases for left and right eyes, and the user is required to wear specially designed spectacles. The cost of building such a system is high. 2) Projection on flexible surface: most existing systems display information on flat rigid projection screens, extending it to non-planar flexible surfaces is an interesting and useful research direction; 3) Direct user-info interaction: existing systems using mouse and screen have limited

freedom of control and low level of user experience. Direct manipulation of the display object by the hands of a user is more natural;

4) Mobile projector display: portable or embedded projectors are becoming more and more popular, but some fundamental problems, e.g. the keystone correction, are not fully studied.

Motivated by these problems, we explore the potential of projectors in interactive information visualization and processing in this thesis. In particular, we make three contributions. First, we propose a computer vision solution for direct 3D object exhibition and manipulation without the user wearing spectacles. In our approach, a new 3D display interface is designed by projecting images on a hand-held foam sphere which can be moved freely by the user. By tracking the motion of the sphere and projecting motion-dependent images onto the sphere, a virtual 3D perception can be created. Using this interface, the user will experience as if he is holding the real object in hands and be able to control the viewing angle freely.

Second, we extend the projection on traditional rigid screen to projection on flexible surfaces. A new flexible display method is proposed, which can project information on a hand-held flexible surface (e.g. an ordinary white paper with a checker pattern at the back) that can be twisted freely. While the user twists the projection surface, the system recovers the deformation of the surface and projects well-tailored information onto the surface corresponding to the de-

formation. As a result, the viewer will see the information as if it was printed on the paper. Two applications, the flexible image projection and curvilinear data slicing are created to demonstrate the usefulness of the method.

After the studies on fixed-position projection, we conduct an investigation on mobile projectors, which is becoming especially necessary with the rapid popularity of mobile projectors. We propose a hand-held movable projection method that can freely project keystone-free content onto a general flat surface without any markings or boundaries on the displaying screen. Compared with traditional static projection systems that keep the projector and screen in fixed positions, our projection scheme can give the user greater freedom of display control while producing undistorted images at the same time.

To verify the correctness of our methods, we built prototype systems using off-the-shelf devices and conducted extensive experiments, including both simulation and real experiments. The results show that the proposed methods are effective and good performance has been achieved. In particular, the real-time speed and low-cost requirement make it quite appealing in many application areas, such as education, digital games, medical applications etc. Capitalizing on the shrinking size, increasing portability, and decreasing cost of projectors, it is predictable that projector-based interactive processing

will become more and more popular in the near future. We believe the research work in this thesis will provide a good foundation for further research and development on computer vision and projector-based applications.

論文摘要

近年來，隨著人機交互技術的不斷發展，投影儀在人機交互方面的強大潛力開始被發掘出來。它不再僅僅被當作一種純粹的顯示工具，而是被研究人員當著一種人機交互工具，用來改變傳統用戶界面的工作方式。儘管，在過去的研究工作中，人們已經對它作了較多的研究，但是它在某些方面的應用和方法仍值得進一步研究，譬如：

- (一) 低成本的三維物體顯示：傳統顯示三維物體的方法是利用偏振光原理，通過對左右眼顯示不同相位的偏振光，並讓用戶佩戴特製的眼鏡，以觀察到三維效果。這種方法被廣泛應用在立體電影上，但其造價高昂。
- (二) 可彎曲的投影屏幕：現有的大部分顯示系統都是以一個剛性的平面作為屏幕，把信息顯示在可彎曲的曲面上是一項有趣且有用的研究課題。
- (三) 直接的用戶信息交互：現有的現實系統大多採用鼠標或者觸摸屏來實現人機交互，其可控性和用戶體驗都受局限。探索更加直接、更自然的人機交互方式具有重要意義。
- (四) 手持式移動投影儀自動形變校正：手持式或者嵌入式投影儀已經變得十分普遍，但是關於它的一些重要問題，譬如移動過程中的自動形變校正，還未被仔細研究。

基於此，在本論文中，我們重點研究投影儀在交互式信息顯示方面的一些問題。

本文的研究貢獻主要有以下三點：

首先，我們利用計算機視覺原理，提出一種嶄新的低成本三位物體顯示方案，無須用戶配戴立體眼鏡。該方案利用一個由用戶手持並可隨意移動的白色球體作為投影屏幕，將物體影像投射到該球上。通過追蹤球體的位置及轉移角度，一個相應的三維影像就會即時產生並投射到球體上。用戶會感覺猶如手持影像中的實物一

樣，更可以隨意控制觀賞角度。

其次，我們將在傳統剛性屏幕上的投影擴展到可彎曲的屏幕上。在我們的方案中，用戶手持一個可自由彎曲的顯示屏幕（這個顯示屏幕可以是一張普通的白紙，在背面印上棋盤格）。當用戶彎曲屏幕時，通過追蹤並計算屏幕的變形，並據此對投影影像進行變形，便可使得投影到屏幕上的影像產生屏幕一樣的變形，讓用戶感覺猶如影像是印刷在屏幕上的一樣。作為應用之一，我們實現了一個可以觀察三維體數據的彎曲橫截面的工具。

最後，我們將研究焦點轉移到移動投影儀上。移動投影儀由於體積和價格的不斷下降，變得越來越普遍，其研究價值也隨著提高。我們重點研究了手持移動投影儀的自動形變校正問題，即對自由移動中的投影儀，實時校正產生的形變。我們提出的算法可以應用在普通的平面屏幕上（譬如牆壁、地板等），不需要借助屏幕邊界信息，也不需要再在屏幕上面作標記，因而應用範圍較廣。與傳統的固定投影系統相比，用戶可以利用裝備該校正算法的手持投影儀，實現自由控制投影角度。

爲了驗證我們的方案，我們構建了相應的原型系統，並做了大量試驗，包括模擬數據和真實數據測驗。結果顯示，我們提出的方案可行並有效。而且，實時性和低成本的特點使得我們的方案可以廣泛應用於教育、電子遊戲以及醫療應用等方面。尤其是隨著移動或潛入式投影設備的不斷發展，可以預見，對投影儀的交互顯示方面的研究將會越來越廣泛深入。我們希望，本文的研究工作會對未來的交互式顯示及投影研究起到一個拋磚引玉的作用。

Acknowledgement

First and foremost, I would like to express my greatest gratitude to my supervisor, Prof. WONG Kin-Hong, for his continuous help, encouragement, trust and kindness throughout my PhD study. His patient guidance and insightful comments have improved my research skills greatly. I am so fortunate as a student to work with him. The experience will be always invaluable to my life.

I am also very grateful to Prof. WONG Tien-Tsin, Prof. SUN Hanqiu, and Prof. JIA Jiaya, for their valuable comments and suggestions, which help me extend my work with different perspectives and methodologies.

Thanks to our research group members, Bob (GONG Yibo), Chuen (LEUNG Man-Chuen), Fung (KO Hoi-Fung), KK (LEE Kai-Ki), Edward (FUNG Hung-Kwan) and Simon (WONG Ming-Yee), for their suggestions and help.

I also would like to thank Prof. HENG Pheng-Ann, Prof. LUI Chi-Shing John and Prof. LEE Ho-Man Jimmy for their help and friendship.

Last but not least, thanks to all my friends.

This work is dedicated to my parents, my wife, and my lovely baby.

Contents

Abstract	i
Acknowledgement	vii
1 Introduction	1
1.1 Motivation	1
1.2 Our Work and Contributions	7
1.2.1 Possible applications	11
1.2.2 New technologies developed	11
1.3 Organization of the Thesis	13
2 Background Study	15
2.1 Projective Model and Calibration	15
2.2 Previous Methods and Systems	20
2.2.1 Static projector and screen systems	20
2.2.2 Movable projector or screen systems	22
2.2.3 Non-planar and flexible screen systems	23
2.3 Other Related Research Areas	25

2.3.1	Camera pose estimation	25
2.3.2	Nonrigid surface reconstruction	26
3	Interaction Using A Hand-held Sphere	28
3.1	Introduction	29
3.2	System Overview	33
3.3	System Calibration	34
3.3.1	Calibration of the projector camera pair	36
3.3.2	Calibration of the Wiimote camera pair	41
3.4	Sphere Detection and Tracking	42
3.4.1	The Wiimote tracking	43
3.4.2	Alternative: tracking with cardboard	48
3.4.3	Extension of the configurations	49
3.5	Movement and View Dependent Projection	51
3.5.1	View spot allocation	51
3.5.2	Projection image warping	52
3.6	Implementation and Results	54
3.7	Discussions	62
4	Projection On A Flexible Screen	64
4.1	Introduction	65
4.2	System Overview	68
4.3	System Calibration	69
4.3.1	The projective model	70

4.3.2	Calibration method	72
4.4	Paper Surface Tracking and Recovery	77
4.5	Applications	83
4.5.1	Flexible image projection	84
4.5.2	3D volume visualization	86
4.6	Experimental Results	87
4.7	Discussions	98
5	Mobile Projection Keystone Correction	101
5.1	Introduction	102
5.2	System Overview	106
5.3	Projector-Camera Pair Calibration	107
5.4	Projection Region Detection and Tracking	109
5.4.1	Detection	109
5.4.2	Particle filter tracking	111
5.5	Automatic Keystone Correction	114
5.5.1	Recovering 3D projection region	114
5.5.2	Looking for inscribed rectangle	116
5.5.3	Pre-warping projection image	117
5.6	Experimental Results	117
5.7	Discussions	126
6	Conclusion and Future Work	129
	Bibliography	133

List of Figures

1.1	Emerging devices with a built-in projector.	2
1.2	The classification of projection systems.	4
1.3	Flow of our research.	8
1.4	The overall diagram of the hand-held spherical display.	9
1.5	The overall diagram of the flexible display.	10
1.6	The overall diagram of the mobile keystone correction.	10
2.1	The projective camera model.	16
3.1	The configuration of our system: (a) The overall diagram of our system; (b) The projector, camera and Wiimote fixed on the rig; The foam sphere with four IR LEDs embedded on the surface; Another foam sphere encompassed with a cardboard.	31
3.2	The overview of our system.	33
3.3	The projective geometry among the projector, camera and Wiimote.	36

3.4	The center-collinear spheres and their common conic section. The sphere at arbitrary position can be viewed as rotated from a sphere on the z-axis.	38
3.5	The definition of the object coordinate. (a) In the Wiimote configuration, the origin of the object coordinate is defined as the center of the sphere and its x-y plane is parallel to the plane formed by the four LEDs. (b) In the cardboard configuration, the origin of the object coordinate is defined as the center of the sphere and its x-y plane is parallel to the cardboard.	43
3.6	The flow diagram of the proposed particle filter algorithm.	46
3.7	Movement and view dependent projection.	53
3.8	(a) The accuracy of the projector-camera pair calibration. (b) The accuracy of the Wiimote-camera pair calibration.	55
3.9	Some frames of the tracking results with (a) the Wiimote tracking configuration and (b) the cardboard tracking configuration.	57

3.10	Some frames of the projection results of a 3D face with the Wiimote configuration. The left column shows the projection images and the right column shows the display results.	60
3.11	Some frames of the projection results of a 3D face with the cardboard configuration.	61
3.12	The processing time of the particle filter tracking algorithm against the number of particles.	63
4.1	The configuration of our system: (a) The overall diagram of our system; (b) The facing down projector and camera, the facing up tracking camera pair, and an ordinary paper with checker pattern printed on the back as the flexible surface.	67
4.2	The overview of our system.	70
4.3	The corresponding images captured by (a) the observation camera and (b) the reference camera. The cross in the reference camera found via the homography is also shown in (b).	73
4.4	An example of the flexible triangulation of a 4×3 checker pattern. The triangulation in (b) is reprojected to the paper in (a).	79

4.5	The distributions of the back projection error for the two calibrations. (a) Estimation of P . (b) Estimation of M	88
4.6	Some frames of the synthetic 4×3 surface sequence.	90
4.7	The performance of our algorithm on a synthetic sequence. The flexible triangulation achieves better accuracy and stability.	90
4.8	Recovering the surface of a paper printed with 4×3 checker pattern in live webcam capture. The first row shows the tracked check patterns. The second row shows the recovered surface in another perspective.	91
4.9	(a) The recovering accuracy of the curvature. The errors between recovered curvature and ground-truth are also plotted. (b) The approximate maximum acceptable deformation of the paper.	92
4.10	The working area of the system.	93
4.11	Tracking and recovering the surface of an A3 paper printed with a 6×4 checker pattern in a live webcam capture. The first row shows the tracked checker patterns. The second row shows the recovered surface in another perspective.	94

4.12	The green grid is projected while the black grid is printed. The coincidence of the green grid with the black grid indicates the projection accuracy of the system.	95
4.13	Some results of flexible image projection.	96
4.14	Some results of slicing a MRI brain.	97
5.1	(a) Keystone correction for mobile projector. (b) The mobile projector attached with a webcam in our prototype system.	103
5.2	The keystone correction flowchart. The dashed boxes highlight the three modules of the method.	106
5.3	(a) A cardboard is held to collect the cross projected onto it. (b) The cardboard and the cross in the camera are detected to calculate the 3D-2D correspondence.	108
5.4	Some typical shapes of the green frame.	110
5.5	(a) The original projection image. (b) The pre-warped projection image with a green frame added along the border.	118
5.6	The distribution of back-projection errors of the estimated projection matrix.	119

5.7	The mean and std image error of the tracked projection region w.r.t different noise levels on simulation data.	120
5.8	The recovered trajectory of the projector on real data.	121
5.9	(a) The projection result without keystone correction. (b) The projection result after keystone correction.	122
5.10	Some correction results in real projection.	123
5.11	The error of keystone correction against different poses of the projector.	125
5.12	Comparison of our keystone correction module with the static projector keystone correction method proposed in [30].	125

List of Tables

3.1	Accuracy of the tracking with two configurations . . .	58
3.2	Running time per frame with two configurations . . .	62
5.1	Overall comparison between the proposed method and others	105

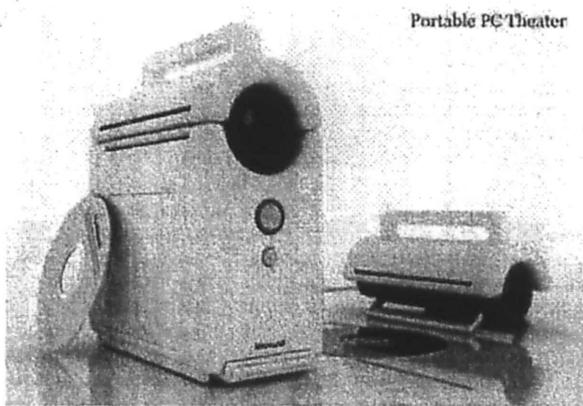
Chapter 1

Introduction

1.1 Motivation

Visual information exhibition and interaction are essential in human daily life. Continuous endeavor has been devoted to seek more natural interfaces and devices, as witnessed by the evolution from traditional 2D to 3D TVs and cinemas nowadays. Another well-known example is that the traditional teaching interface of writing on blackboards in most classrooms is replaced with computers and projectors.

As an important display device, a projector has been widely used to display information for decades, and has become a standard computer output device for a large audience, for example in a cinema. Moreover, the physical size of a projector is much reduced recently, which makes it feasible to be incorporated into a personal computer, or many mobile devices such as cell phones or cameras (see



(a) PC theater



(b) ASUS projector laptop



(c) LG projector phone



(d) Nikon projector camera

Figure 1.1: Emerging devices with a built-in projector.¹

¹All images are downloaded from Internet:

(a) from <http://www.yankodesign.com/2007/12/13/portable-pc-theater/>,

(b) from <http://www.slashgear.com/asus-pico-projector-laptop-prototype-0511907/>,

(c) from <http://www.artuji.com/lg-projector-phone-unveiled/3071>,

(d) from <http://www.dpreview.com/news/0908/09080402nikons1000pj.asp>

Fig. 1.1). It is predictable that the projector will become ubiquitous in the near future.

We all know that traditional over-headed projection systems have certain limitations. For example, a standard projection systems usually displays 2D information on static flat screens and the interaction with a user is limited to indirect pointing using a mouse or direct pointing by a laser pointer. The control of the display such as changing the viewing angle and viewing distance etc, cannot be done conveniently. Consequently, the static display technology is not sufficient for many emerging applications, such as product exhibition, medical data visualization, game and other interactive applications. In addition, a new batch of mobile projector products provide a new opportunity to develop a set of low-cost, more user-friendly, and highly interactive tools and applications. These limitations and opportunities have made the research of display technologies a hot topic in the past few years.

In the literature, innovative applications of projectors have been investigated. Representative systems include the CAVE system [11], which uses three rear projectors to project information onto three walls of a cube-shaped room and one facing down projector to project on the floor, creating a fully immersive virtual reality environment; Also, the Shader Lamp [46] uses the projection light to alter the appearance of a complex 3D object, simulate a real scene, or make one

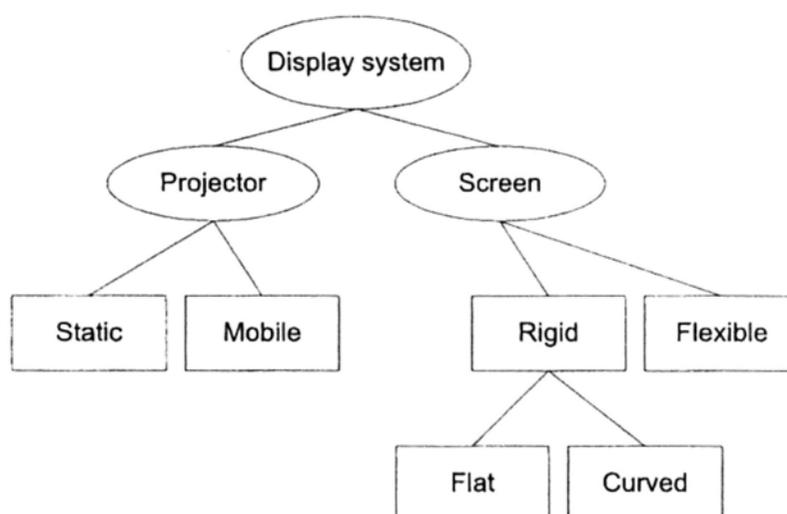


Figure 1.2: The classification of projection systems.

object look like another. Another direction is the movable display, in the Portable Display Screen system [6] and Active Pursuit Tracking system [17], they track a hand-held cardboard to create a movable projection system; Multiple-projector systems are also popular in research, the example is in [47], which creates a panoramic display with multiple casually positioned projectors.

As shown in Fig. 1.2, different display systems can be classified according to the property of the projectors or screens the system used. In general, the projector can be classified into static and mobile, and the screen can be classified into rigid and flexible. So, there are roughly five kinds of systems: static-projector system, mobile-projector system, flat-screen system, curved-screen system, and flexible-screen system.

Although many possibilities have been investigated in the previ-

ous work, certain aspects such as user-oriented display and interaction still require further investigation. According to our survey, the following issues have not been studied or further improvements are needed:

(1) Traditional 2D display systems usually display 3D objects on static screens. The perceived level of 3D sensation is low. Technologies using polarized light are popular to create 3D perception by 2D monitors. However, they require the user to wear specially designed spectacles and the cost of building such a system is high. Another approach is using motion parallax, that is, the user can perceive 3D information when the display is moving. Projectors provide the possibility to create motion parallax. We propose that we can create a low-cost 3D display by motion parallax using projectors and other off-the-shelf devices.

(2) Existing projection system usually project information on flat screens, rarely on flexible surfaces. However, a flexible surface is common in everyday life, and applications that deal with flexible display is quite necessary under some circumstances. For example, in medical visualization, a common way for doctors to analyze medical volumetric data such as MRI and CT is to view the cross-sectional slices of the data. Although some slicing tools that can display the planar slices have been produced, viewing curved slices is necessary in some cases, since the human internal organs are usu-

ally not planar but curved. So a flexible slicing tool that not only displays planar slices but also curved slices may help the doctor to diagnose diseases.

(3) The interaction with the display system or the information is traditionally achieved via indirect input devices such as a mouse or laser pointer. This kind of interaction is relatively unnatural, indirect, and limited. Moreover, the level of reality of user perception is low. These reasons lead to the popularity of the human-centric interactive systems in which the user can directly interact with the displaying content. A common problem of these systems is high cost, since special designed expensive hardware is usually employed. Low-cost and easy-to-build approaches are in need.

(4) Mobile or embedded projectors are becoming more and more popular in the personal electronic consume market. The research about employing mobile projector is also booming in recent years. However, some fundamental problems have not been investigated, such as the automatic keystone correction problem of rectifying the distorted projection region to a rectangular area when the projector is oblique to the screen. Although the problem for desktop projectors has been studied for many years [30] [44], for mobile projector it has not been investigated yet. Some may think that the keystone correction could be adjusted manually by the user. However, it is much more appealing that the keystone distortion can be automati-

cally removed. Moreover, this function will be a necessary step for many useful applications.

The above issues motivate us to propose creative methods and systems. This thesis is devoted to investigating such issues and exploring the uncovered potential of projectors in interactive visual information processing. Specifically, it investigates in depth the research issues involved in the design, implementation and evaluation of interactive projector-based visual systems, especially in terms of how the information is displayed, what types of projectors and surfaces are used and how the user interacts with the information. In this work, technologies in multimedia, computer vision and graphics fields are investigated and a number of novel ideas and technologies are proposed. These will certainly be useful in the further research in similar areas.

1.2 Our Work and Contributions

The work of this thesis can be divided into three related parts, as shown in Fig. 1.3. Firstly, a low-cost solution is proposed for direct 3D object manipulation and exhibition without spectacles. A new 3D display interface is designed by projecting images on a hand-held foam sphere which can be moved freely by the user. By tracking the position and the rotation of the sphere, an image of the 3D object will be generated and projected onto the sphere in real-time. The motion

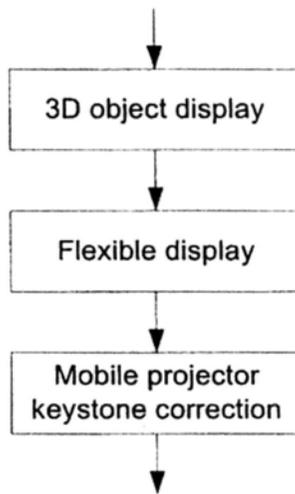


Figure 1.3: Flow of our research.

parallax will give the user a perception as if he was holding the real object in hand and he could control the viewing angles freely. The illustrative diagram is shown in Fig. 1.4.

Secondly, we extend the projection on rigid screens to projection on flexible screens, which is particularly useful when visualizing some medical data. A new flexible display method is proposed which can project information on a hand-held flexible surface (e.g. an ordinary white paper) that can be twisted freely. While the user twists the projection surface, the system recovers the deformation of the surface and projects well-tailored information onto the surface corresponding to the deformation. So that the viewer sees the information as if it was printed on the paper. Two applications, a flexible image projection and the curvilinear slicing are created to demonstrate the method. The illustrative diagram is shown in Fig. 1.5.

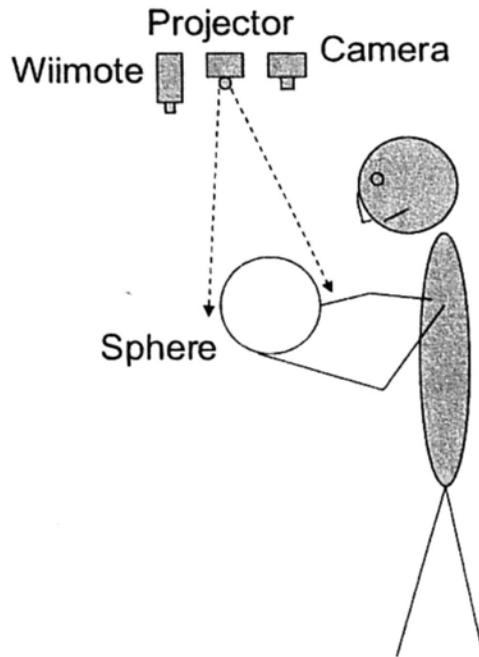


Figure 1.4: The overall diagram of the hand-held spherical display.

After the investigations on over-headed projector, we conduct an investigation on the mobile projection. The research becomes especially necessary with the rapid development of mobile projectors. We propose a hand-held movable projection method that can freely project keystone-free content on a general flat surface without any markings or boundary printed on the screen. Compared with traditional static projection systems that keep the projector and screen in fixed positions, our projection scheme can give the user greater freedom of control while producing undistorted display at the same time. The illustrative diagram is shown in Fig. 1.6.

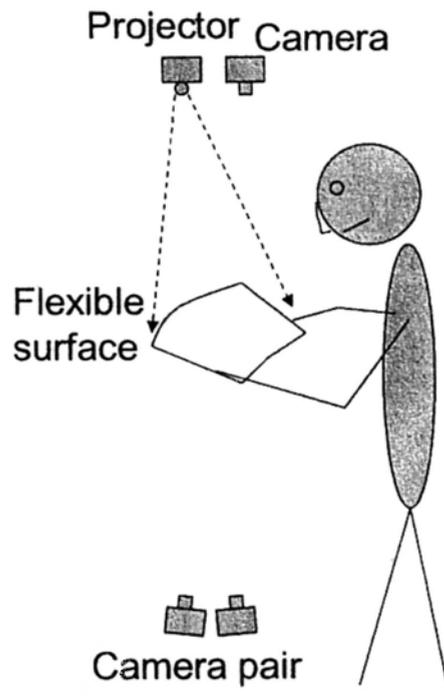


Figure 1.5: The overall diagram of the flexible display.

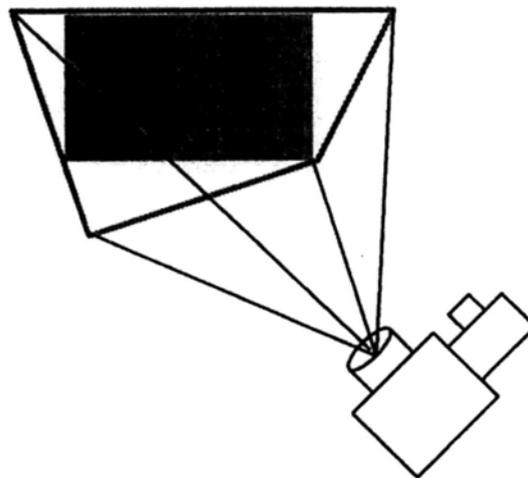


Figure 1.6: The overall diagram of the mobile keystone correction.

1.2.1 Possible applications

The proposed methods and systems have a lot of applications in practice. For example, in a museum mounted with many projectors, the visitor can use the hand-held 3D object manipulation and exhibition tool to explore the computer-generated copy of the historical artifact that is not available to the visitor for touching. This gives the visitor a realistic feeling about the artifact, and offers direct manipulation sensation for users, while the real object is protected from damage. Likewise, the flexible display system can be applied to the medical field for helping doctors to analyze the MRI and CT data. Similarly, in manufacturing industry, it can be a model-preview tool to preview the appearance of flexible models and see how it can be twisted by the users in the design stage. Finally, the movable keystone correction technology will become very useful as portable electronic devices with a built-in projector such as cell phones, cameras, and PDAs are popular in the near future.

1.2.2 New technologies developed

To summarize, the major contributions of this thesis are the proposal and implementation of a few creative ideas and systems using low-cost off-the-shelf projectors, webcams, and IR LEDs etc. Furthermore, a series of new methods and technologies which are useful in virtual reality development are proposed. They are summarized as

follows:

(1) Projector and camera calibration technologies

We propose a simple and robust algorithm for calibrating the relationship between the projector-camera pair, or the camera pair. A 3×4 projection matrix, instead of explicit parameters, is estimated by projecting correspondence points between the projector and camera. Compared with traditional full calibration methods, the proposed calibration method is simple, robust, and easy-to-use.

(2) Vision and IR LEDs based camera tracking method

Robust camera tracking algorithms using visual features and/or IR LEDs are proposed under the framework of Particle filter. The tracking system is built based on low-cost webcam, IR LEDs, Wiimote (a lost-cost game remote controller with a built-in IR camera), and foam sphere or cardboard. Robust real-time performance is obtained.

(3) Real-time 3D surface reconstruction and tracking algorithm

We propose an efficient 3D surface tracking algorithm to recover the surface of a flexible paper. A flexible triangulation model is proposed to improve the deformation ability of traditional fixed triangulation model. A two stage optimization alternating between the reconstruction error and the smoothness of the surface is employed to recover the 3D surface efficiently.

(4) Continuous and markless mobile keystone correction method

An effective solution is proposed to deal with the movable markless projector keystone problem. By projecting and tracking a green boundary of the projection screen, the keystone distortion is automatically corrected in a continuous manner. The major novelty lies in a particle filter based tracking scheme artfully introduced without full calibration of the projector. It also uses a coplanarity enforcement process to improve the accuracy of the recovered projection region. To our knowledge, we are the first to deal with the mobile markless keystone correction problem.

1.3 Organization of the Thesis

The rest of the thesis is organized as follows:

Chapter 2 gives a survey of the previous work and introduces the background knowledge related to the thesis.

Chapter 3 discusses the problem of displaying 3D information using projectors and motion parallax. A hand-held spherical display system for 3D object exhibition and manipulation is presented, including the design, implementation and evaluation of the prototype system.

Chapter 4 discusses the problem of projection on flexible displays, especially the recovery of the shape of the surface, and image warping to adjust the information, and the applications of the flexible display. Also, a prototype system is built, including the design,

implementation and the evaluation.

Chapter 5 discusses the problem of automatic correcting the keystone distortion of a mobile projector. The method is discussed in detail, and its differences and advantages over static projector keystone correction methods are discussed and compared.

Chapter 6 gives the conclusion of the thesis, discusses the limitation of the work as well as directions for future exploration.

End of chapter.

Chapter 2

Background Study

In this chapter, we present the background of projection technologies. The chapter is organized as follows: we start with an introduction of the projective models (camera and projector model) and the calibration methods. Then previous projection methods and systems in different categories are reviewed, such as static projector-and-screen systems, movable projector-or-screen systems, non-planar screen system and flexible screen systems. Other technologies related to this thesis are also introduced in brief.

2.1 Projective Model and Calibration

The projective model of a camera is generally modeled as the pin-hole model that maps the 3D world point to the 2D image pixel as a perspective projection model, as shown in Fig. 2.1. The geometric relationship between a 3D point $\mathbf{X}(x, y, z)$ in the 3D space and its

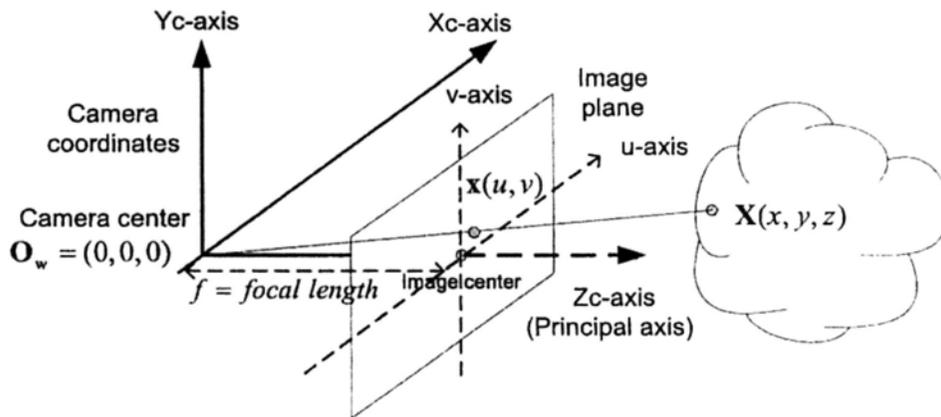


Figure 2.1: The projective camera model.

2D Projection $\mathbf{x}(u, v)$ is established as follows:

$$\begin{aligned} \lambda \tilde{\mathbf{x}} &= \mathbf{G} \tilde{\mathbf{X}} \\ &= \mathbf{K} [\mathbf{R} \ \mathbf{t}] \tilde{\mathbf{X}} \end{aligned} \quad (2.1)$$

with

$$\mathbf{K} = \begin{pmatrix} \alpha & c & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.2)$$

where λ is a scale factor; \mathbf{G} is the projection matrix; \mathbf{R} and \mathbf{t} are the rotation matrix and translation vector which relates the world coor-

dinate system to the camera coordinate system, which are known as the extrinsic parameters; \mathbf{K} is known as the camera intrinsic matrix, in which α and β are the focal lengths in the two axes of the image plane, c is the skew parameter that describes the skewness of the two image axes, u_0 and v_0 are the coordinates of the image center.

The task of camera calibration is to find the intrinsic and extrinsic parameters. It is an essential step in many computer vision applications. Once this projection model is established, 3D metric information can be recovered from 2D images. To perform the calibration, the most important thing is to find a number of correspondences between the 3D world points and the 2D image pixels. In the literature, much work has been done. According to whether a calibration object is needed or not, existing methods can be roughly classified into three categories: in the first category, a specially-designed calibration apparatus with known 3D metric is used and an elaborate setup is needed. By taking pictures of the calibration object at different views, a number of 3D to 2D correspondences can be established conveniently. Representative work include [56] [12]; the second category is known as self-calibration, which does not use any calibration object. Just by moving the camera in a static scene, the rigidity of the scene provides constraints on the camera's intrinsic parameters. By taking more than three images, the correspondences of the scene features between them are enough to recover the cam-

era intrinsic parameters up to a scale factor. Representative methods include [39] [19] [38]. This category of methods are very flexible, since there is no special calibration object required. However, it is difficult to obtain reliable results. One well-known self-calibration method [18] is using the Kruppa's Equations [13]. It first calculate the epipolar transformations by tracking a number of salient points between image frames captured by the moving camera, and then uses the Kruppa's equations to solve the intrinsic parameters of the camera. The third category of methods is between the two categories. It seeks less requirement about the calibration process than the first category while ensures more reliable calibration results than the self-calibration methods. This kind of methods usually employ a planar board as the calibration object. For example, Zhang [61] [62] proposed a method that requires a planar checkerboard to be observed by the camera at different views. The corners points of the checker pattern are automatically detected and used to computer the projective transformation between the image points of different images. The camera intrinsic and extrinsic parameters are then solved with a closed-form solution, and refined by a nonlinear optimization. Another method is developed by Triggs [54], which is similar to [61]. However, it requires at least five views of a planar scene and is difficult to initialize.

The projective model of a projector is the same to that of a camera

except for the projection direction. For a camera, 3D world points are projected onto 2D image pixels, while 2D image points are projected out as light rays for a projector. The projective model of a projector can be also modeled as a perspective projection via intrinsic parameters and extrinsic parameters:

$$\begin{aligned}
 \lambda \tilde{\mathbf{y}} &= \mathbf{M} \tilde{\mathbf{Y}} \\
 &= \mathbf{J} [\mathbf{R} \ \mathbf{t}] \tilde{\mathbf{Y}} \\
 &= \begin{pmatrix} \alpha & c & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} [\mathbf{R} \ \mathbf{t}] \tilde{\mathbf{Y}} \quad (2.3)
 \end{aligned}$$

similarly, λ is a scale factor; \mathbf{M} is the projection matrix; \mathbf{R} and \mathbf{t} are the rotation matrix and translation vector which relates the world coordinate system to the projector coordinate system; \mathbf{J} is the projector intrinsic parameter matrix, α and β are the focal lengths in the two axes of the projector image plane, c is the skew parameter, u_0 and v_0 are the coordinates of the image center.

To calibrate a projector, an additional camera is needed, since the projector cannot observe the scene, a camera is usually included as a reference and to guide the projection. Apart from this, the calibration method of a projector is quite similar to that of a camera.

Normally, the projector projects some pre-defined patterns onto a screen and the camera observes the patterns. In this way, a projector-screen-camera correspondence is established, and the calibration can be carried out based on the correspondence.

2.2 Previous Methods and Systems

2.2.1 Static projector and screen systems

In the literature, numerous projector camera systems have been developed to create many kinds of applications. One popular application is to use multiple projectors to build large display walls for creating immersive environment with both rear projection [21] [31] and front projection [59]. The CAVE system [11] uses three rear projectors to project onto three walls of a cube-shaped room and one facing down projector to project onto the floor, creating a fully immersive virtual reality environment. The Teleport system [15] uses a projected wall to create the illusion of extending the room to another one for a teleconferencing system. Projection technologies are also used to modify the appearance of a real object or an environment. The Shader Lamp [46] explores the use of projection light to alter the appearance of a complex 3D object. The ability of controlling the appearance of an object enables applications such as simulating a real scene [36], or making one object look like another [16]. In most

of these applications, the projector and screen are at fixed positions, and a planer screen is usually used.

In order to create correct projection, projector-based systems require various calibration processes, including geometric calibration, photometric calibration etc. Since the projector cannot observe the projection result, a camera is usually included as a visual feedback in the calibration. For a planar screen, the geometric relationships among the projector, camera and screen can be represented by a couple of 3×3 homographies, which can be estimated via the correspondences among them. For example, Sukthankar *et al.* [52] proposed a smart presentation system in which they estimated the projector-to-camera homography via projecting known patterns to the screen, and estimated the screen-to-camera homography via detecting the boundary of the screen. An auto-calibration algorithm without detecting the screen boundary or using markers on the screen is also proposed [41]. The full calibration algorithm [44] can estimate all parameters including intrinsic parameters of the projector and the relative pose between the projector and camera. However, these two methods are more complicated. In multiple projector systems, the calibration further requires estimating the relationship among the projectors [43]. Moreover, in order to stitch multiple projection images from different projectors, photometric calibration is also needed. Detailed discussion can be found in [9][4].

2.2.2 Movable projector or screen systems

Mobility is being given more and more attention in the development of projector camera systems. With a movable projector or screen, more interaction can be introduced to enrich the user experience. A steerable projector [2] is one that the beam can be moved under computer control to illuminate different objects. The Pan-tilt projector [6] is another kind of popular movable projectors, which is mounted on a pan-tilt base, thus allows rotation about its optical center. Other than controlling the projector directly, other methods are also proposed to give mobility to the projector. For example, by using a pan-tilt mirror and the computer-vision based technologies, the Everywhere Display Projector System [42] can control the projection light to nearly everywhere in a room like a retail store [51].

The movement of the projection surface is more common than the movable projector. By allowing the users to hold and move the screen with their hands, more natural and easier interaction can be achieved. Technically, the success of movable-surface system relies on reliable tracking of the surface. In existing systems, two categories of tracking mechanisms are most frequently used. One is based on magnetic sensors or specially-designed tracking devices. For example, the Dynamic Shader Lamp [3] extends the Shader Lamp [46] to allow users to hold the object in hands by adding a six degree optical tracker and a magnetic tracker. Though good track-

ing accuracy and robustness can be achieved, the tracking sensors are expensive and also limited to certain environments. On the other hand, computer vision technology is usually employed to track regular surfaces. Since a camera is often included to calibrate the projector, it is natural to use it to track the display surface without including extra cameras. Many existing systems use vision-based tracking methods. The Portable Display Screen (PDS) system [6] detects and tracks a cardboard with black borders using Hough transform and Kalman filter. The Active Pursuit Tracking system [17] attaches four color fiducials to a white cardboard and tracks them using the Camshift algorithm. Leung *et al.* [28] proposed a particle filter algorithm to track a white cardboard based on edge features. The major advantage of vision-based tracker over sensor-based tracker is its low cost, though the tracking accuracy and robustness may be not very high. Since low cost is one of the design goals of our system, we track the motion of the movable based on vision algorithms as long as satisfactory robustness and accuracy can be achieved.

2.2.3 Non-planar and flexible screen systems

Non-planar display surfaces are mainly employed in two kinds of applications. One is for arbitrary surface display. For example, Kondo *et al.* [25] proposed a Free Form Projection Display (FFPD) system for displaying images on arbitrarily-shaped surfaces. By

scanning the 3D structure of the surface with a 3D scanner, they can display 3D content onto the surface without distortion. Another kind of application is for large scale immersive display which gives the viewer an immersive experience and more freedom of view. Multiple projectors and cameras are included in such systems in order to cover a larger portion of the surface. For example, Raskar et al. [45] [47] proposed a scalable panoramic display system with multiple casually positioned projectors. The geometric calibration of the projector-camera pair with curved surface is more complicated since their relationship is no longer a planar homography. In [47], Raskar et al. proposed a parametric approach called quadric transfer to represent the relationship for quadric surface. The quadric transfer can be used to perform the image warping efficiently after it is found during the calibration process. However, it is not suitable when the surface is movable.

In most existing systems, the screens are static and non-deformable. Although movable display systems such as [53] [28] allow more freedom in control, they still project on rigid projection surfaces. As one of the few flexible screen systems, Lee et al. [27] proposed to display content on some regularly-foldable surfaces such as scroll, fan and umbrella. They attached IR (Infrared) LED markers on the control points of the surface and tracked them using a Wiimote. The surface is then recovered via the markers. Since the Wiimote can

track at most 4 IR LEDs at the same time, the foldable surfaces are limited to those having fairly high folding symmetry. Konieczny et al. [26] built a flexible projection system to project display content onto a flexible surface. However, this approach uses position sensors to track the surfaces. Moreover, it only allows the surface to bend in one dimension.

2.3 Other Related Research Areas

2.3.1 Camera pose estimation

Pose estimation is a classical problem in the field of computer vision. The goal is to estimate the pose including the rotation and translation between the camera and an object given its structure. It plays an important role in many vision problems and applications, including structure from motion [55], and augmented reality etc. There are various techniques to deal with the pose estimation problem. Early work concentrates on investigating how many feature points are needed at least to recover the pose, which is also known as the Perspective n Points (PnP) problem. Fishler and Bolles [14] investigated using three feature points to estimate the pose. Horaud *et al.* [22] estimated the pose of an object using four non-coplanar points by solving biquadratic polynomial equations. Iterative methods [37] [35] [60] are proposed to solve the pose estimation itera-

tively. It finds a good pose to fit the image data in a steepest descent manner, i.e., to minimize the residual error between the predicted and real image position of the feature points. Zhang [61] proposed an efficient pose estimation algorithm in developing his camera calibration method. It is quite efficient for planar objects like a cardboard.

2.3.2 Nonrigid surface reconstruction

The non-rigid surface recovery problem refers to estimating the 3D shape of the surface based on its 2D image observation. It is a severely ill-posed problem in the case of a monocular camera since the depth information is lost under perspective projection. Many prior models and regularization methods have been proposed to solve the ill-posedness. For example, Bregler et al. [8] proposed a factorization method to build the 3D model from the tracked 2D feature points. They simply represented the 3D model as a linear combination of a set of basis vectors, which may limit the deformation ability of the model. Physics based methods [23] have better approximation to model the behavior of a general surface, but the complexity of the model may be very high. Statistical learning techniques [58] [50] are usually employed to simplify complex nonlinear models. However they need a lot of training data in order to obtain a good model. Recently Salzmann *et al.* [49] proposed a tracking method for 3D

surface recovery based on a simple triangulation model. They proposed to constrain the edge orientations of the triangulation model between consecutive frames and formulated it as a Second Order Cone Programming (SOCP) problem. Though the state-of-art result can be obtained, the computation time is very long, about 10 seconds is needed to process a frame. Zhu *et al.* [63] improved the speed by reformulating it into an unconstrained quadratic optimization problem and solving it efficiently.

□ **End of chapter.**

Chapter 3

Interaction Using A Hand-held Sphere

In this chapter, we investigate the possibility of using low-cost projector to improve the interactivity of traditional display systems and add 3D perception to it. An interactive 3D object manipulation and exhibition tool without the viewer having to wear spectacles is developed by projecting the display object onto a hand-held foam sphere. With our system, we can manipulate the object with our hands as if we were holding the real 3D object. While the user holds the sphere and moves it freely, we project well-tailored images onto the sphere to follow the motion of it, giving the viewer a virtual perception as if the object were sitting inside the sphere and being moved by the viewer. The design goal is to develop a low-cost, real-time, and interactive 3D display tool. An off-the-shelf projector-camera pair is first calibrated via a simple but efficient algorithm. Vision based

methods are proposed to detect the sphere and track its subsequent motion. The projection image is generated based on the projective geometry among the projector, sphere, camera and the viewer. We describe how to allocate the view spot and warp the projection image. We also present the result and the performance evaluation of the system. This work is published in [34].

3.1 Introduction

Traditional 2D display systems usually display 3D objects on static screens and the viewer interacts with it using indirect input devices such as keyboard or mouse. It is unnatural and the perceived level of reality is low. A projector is a good choice to improve the freedom and interactive ability of these systems, but nearly all existing systems can only display 2D information. Technologies using polarized light are popular to create 3D perception by 2D monitors, but they require the viewer to wear specially-designed spectacles. Another approach is motion parallax in which the viewer can perceive 3D information when the display is moving. This can be achieved by projectors and motion sensors. The Free Form Projection Display (FFPD) system [25] is one that uses magnetic sensors to create motion parallax for displaying 3D medical organs on irregularly-shaped surfaces. Though good results can be achieved, the system has some limits that prevent it from being widely used in everyday applica-

tions: first, it has a high cost due to the expensive motion sensors used; second, it needs to scan the 3D structure of the surfaces. In this chapter, we propose a low-cost solution that uses motion parallax to create a direct 3D virtual object manipulation tool based on computer vision and several off-the-shelf devices. A prototype system is presented in which we display the 3D object onto a hand-held sphere and manipulate it with our hands directly as we do in our daily lives.

The proposed system finds many applications in real life. For instance, in a museum mounted with many projectors, the visitor can use this hand-held display tool to explore the computer-generated copy of the historical artifact which is not available to the visitor for touching. This gives the visitor a realistic feeling about the artifact and direct manipulation of it while the real object is protected from damage. Likewise, manufacturers can also use this tool to promote a new product when it is still in the design stage and a real product is not yet available. The 3D perception and direct manipulation interface can definitely give the viewer a better feeling about the product.

The objective of our work is to build a low-cost, easy-to-build and workable 3D object manipulation tool without requiring the viewer to wear spectacles. Instead of using magnetic sensors or specially-designed hardware, we use several low-cost off-the-shelf devices and computer vision technologies to build the system. The main

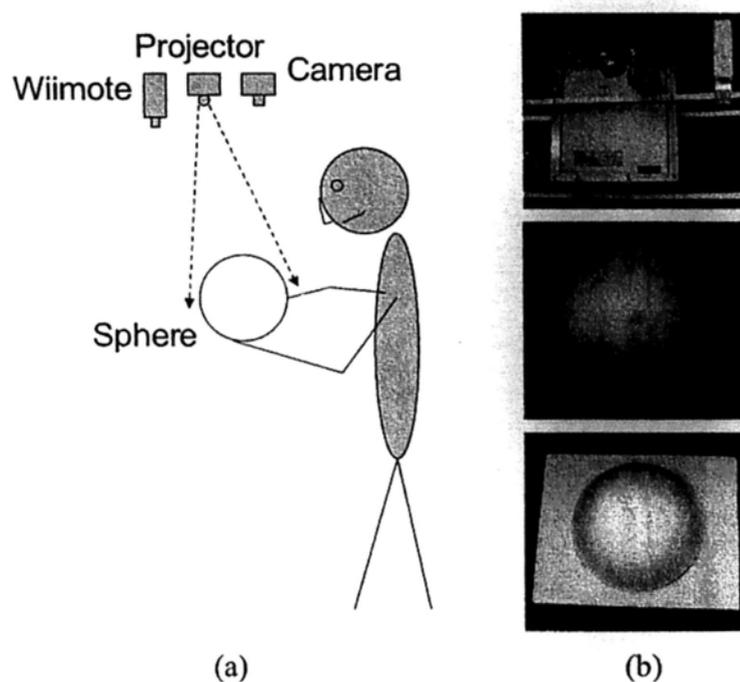


Figure 3.1: The configuration of our system: (a) The overall diagram of our system; (b) The projector, camera and Wiimote fixed on the rig; The foam sphere with four IR LEDs embedded on the surface; Another foam sphere encompassed with a cardboard.

idea of the system is to use a hand-held sphere as the projection surface. When the viewer moves and rotates the sphere, we use object tracking techniques to track the translation and rotation of the sphere. Meanwhile, based on the pre-calibrated projective geometry among the camera, sphere, projector and the viewer, we project well-tailored images of the object onto the sphere depending on its translation and rotation. By continuously adjusting the images projected to the sphere as it moves and rotates, the motion parallax gives the viewer a virtual 3D perception as if the object were sitting inside

the sphere and being manipulated by the user directly. The devices used in our system include a projector, a webcam, a Nintendo Wiimote and a foam sphere. The consideration for choosing a sphere as the projection surface is its symmetry in shape. Moreover, to adapt to different application scenarios, we design two kinds of configurations to track the sphere. In the first configuration, we embed four co-planar IR LEDs on the surface of the sphere, and use the Nintendo Wiimote (which has a PixArt IR camera embedded inside) to track their positions. In the second configuration, the sphere is encompassed with a cardboard. By tracking the cardboard, we can also obtain the motion of the sphere. In Fig. 3.1 (a), we show the whole diagram of our system. The camera, projector, and Wiimote are fixed on a rig, as shown in Fig. 3.1 (b). The two configurations of the sphere are also shown in Fig. 3.1 (b).

The development of the whole system faces many challenges in various computer vision and graphic fields, including projective geometry, projector-camera calibration, object tracking, and spherical display. Our main contribution is the proposal of a new type of display system and the realization of it through the integration of various technologies and devices. The remainder of this chapter presents how we handle these challenges. It is organized as follows: Section 3.2 gives the overview of the system. Section 3.3 describes the calibration of the projector-camera pair. In section 3.4, we introduce

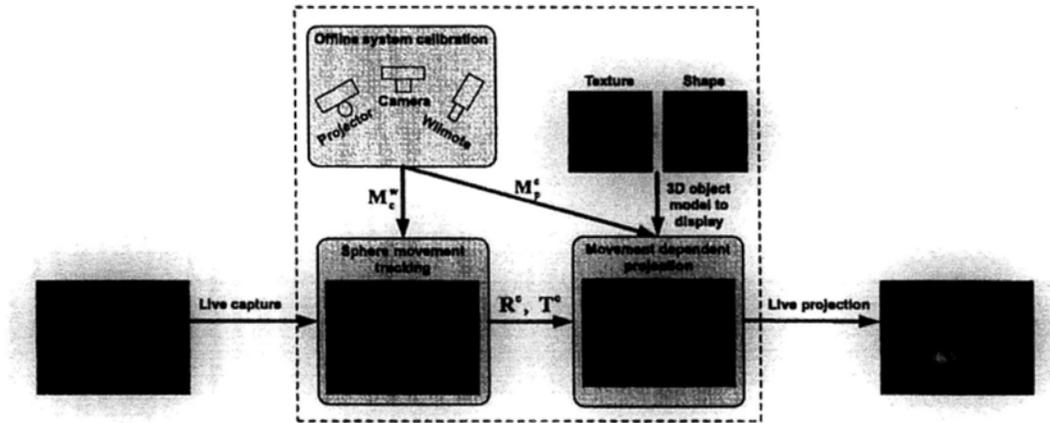


Figure 3.2: The overview of our system.

how to track the translation and rotation of the sphere. Section 3.5 describes how to generate the projection image correctly. The implementation details and results are given in Section 3.6. We discuss this chapter in Section 3.7.

3.2 System Overview

Our system is an integration of three major modules, the calibration module, the tracking module, and the projection module. The calibration module is an off-line module, which finds the relationship among the projector, the camera and the Wiimote. The tracking module takes the live capture from the camera as input, and tracks the translation and rotation of the sphere relative to the camera for each frame. Based on the calibration result and the tracked motion of the sphere, the projection module generates the projection image

of the displaying object and projects it onto the sphere. The workflow of the system and the interaction between different modules are shown in Fig. 3.2. In following sections, we describe each module in detail.

3.3 System Calibration

In the calibration stage, we calibrate two geometric relationships, one between the projector and the camera, and the other between the Wiimote and the camera. The target of calibrating the projector camera pair is to align the projector with the camera, while the calibration of the Wiimote camera pair is for tracking the IR LEDs on the sphere. Previous calibration methods applied to planar surfaces and static systems are no longer applicable due to the movable nature of the spherical surface we used. In our approach, without the need to know explicit geometric parameters, we simply estimate two projection matrices, one from the camera coordinate system to the projector image plane, and another from the Wiimote coordinate system to the camera image plane. Both projection matrices are constant and independent from the movement of the sphere. This makes our calibration fairly easy. A simple calibration algorithm is proposed. The basic idea is to use the sphere as the calibration object. By manually labeling a number of corresponding points on the sphere in the projector and camera (or Wiimote and camera) images,

we can estimate the projection matrices.

The projective model The ideal projection model of the projector is the same as that of the camera except for the projection direction.

The projection from a 3D point to the 2D projector image pixel is also via a 3×4 perspective projection matrix. Then any 3D point in the camera coordinate, for example, a point $\mathbf{X}^c(x, y, z)$ on the sphere, corresponds to its projector pixel $\mathbf{x}^p(u, v)$ via a projection matrix \mathbf{M}_p^c :

$$s\tilde{\mathbf{x}}^p = \mathbf{M}_p^c \tilde{\mathbf{X}}^c \quad (3.1)$$

and

$$\mathbf{M}_p^c = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \quad (3.2)$$

where $\tilde{\mathbf{x}}^p$, $\tilde{\mathbf{X}}^c$ are the homogenous coordinates and s is a scale factor.

The Wiimote has a PixArt IR camera embedded inside which is able to track IR LEDs. So the projective model of the Wiimote can also be represented by a 3×4 projection matrix. Then any 3D point in the Wiimote coordinate, for example, an IR LED $\mathbf{L}^w(x, y, z)$ on the sphere, corresponds to its projection in the camera $\mathbf{l}^c(u, v)$ via a

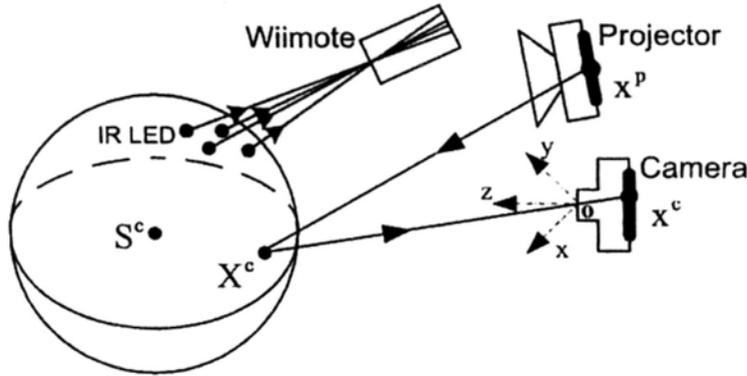


Figure 3.3: The projective geometry among the projector, camera and Wiimote.

projection matrix M_c^w :

$$s\tilde{l}^c = M_c^w \tilde{L}^w \quad (3.3)$$

The task of calibration is then subject to estimating two projection matrices M_p^c and M_c^w .

3.3.1 Calibration of the projector camera pair

The projective geometry between the projector camera pair is shown in Fig. 3.3. The light from some pixel $x^p(u, v)$ in the projector image intersects the sphere at $X^c(x, y, z)$ (in camera coordinate), and then create pixel $x^c(u, v)$ in the camera. These three points (x^p, X^c, x^c) form a correspondence. The basic idea of estimating the projection matrix is to collect a number of such correspondences. A correspondence is collected in the following way: we project a cross at known position onto the sphere surface and observe the cross using the cam-

era. For each correspondence, the 2D coordinates of the point in the camera x^c can be manually labeled while the 3D coordinates of the point on the sphere surface X^c cannot be directly obtained. In order to calculate X^c , we need to locate the 3D position of the sphere's center in the camera coordinate first.

The 3D position of the sphere's center can be located based on its projection in the camera. According to [20], the projection of a sphere is a conic section under the pinhole perspective camera model. Since the depth information is lost in perspective projection, the conic section could be created by a family of center-collinear spheres. Given only the conic section, we cannot uniquely recognize the actual sphere out of the family. However, once the physical radius of the sphere is given, the sphere can be uniquely located. We use the geometric method proposed in [57] to locate the center of the sphere. The basic idea of the method is to investigate the relationship between the general case where the sphere lies at arbitrary position and the special case where the sphere lies on the z-axis of the camera. In the special case, the image of the sphere is a circle and the center of the sphere can be easily located given the circle. The sphere at arbitrary position can be viewed as rotated from a sphere on the z-axis. Accordingly, the image of the sphere changes from a circle to a conic section due to the rotation. So given the conic section C and the radius of the sphere R , we first locate the center

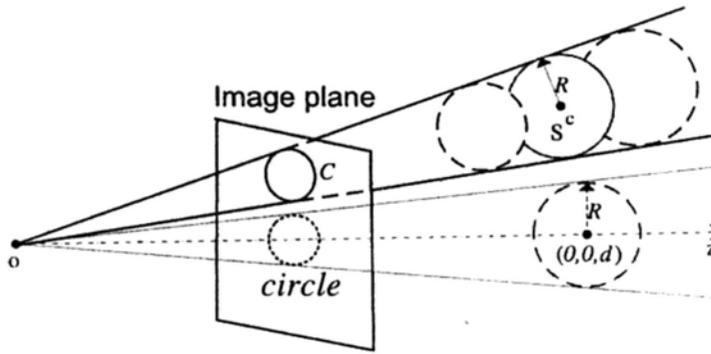


Figure 3.4: The center-collinear spheres and their common conic section. The sphere at arbitrary position can be viewed as rotated from a sphere on the z -axis.

of the sphere $(0, 0, d)$ in the special case and rotate it to the actual position S^c . This is illustrated in Fig. 3.4. In our implementation, we use the Hough transform circle detection algorithm to detect a circle to approximate the conic section. According to our experimental investigation, the relative error of the approximation in locating the center of the sphere is about 4%.

After the center of the sphere is located, we can calculate the corresponding point on the sphere surface for each pixel within the conic section. For each correspondence (x^p, X^c, x^c) , the sphere surface point X^c in camera coordinate should satisfy the following equations:

$$\begin{aligned} s\tilde{x}^c &= K^c X^c \\ \|X^c - S^c\|_2^2 &= R^2 \end{aligned} \quad (3.4)$$

where K^c is the intrinsic parameter matrix of the camera which is

calibrated beforehand using the OpenCV toolbox [24]. The first equation is the projection equation, and the second is to constrain the distance between the surface point and the sphere center. We solve them to obtain \mathbf{X}^c for each correspondence.

Now for each calculated correspondence, we can write a projection equation according to Eq. (3.1) and Eq. (3.2):

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (3.5)$$

We rewrite it in the equivalent form by eliminating the scale factor:

$$\begin{aligned} u &= \frac{m_{11}x + m_{12}y + m_{13}z + m_{14}}{m_{31}x + m_{32}y + m_{33}z + m_{34}} \\ v &= \frac{m_{21}x + m_{22}y + m_{23}z + m_{24}}{m_{31}x + m_{32}y + m_{33}z + m_{34}} \end{aligned} \quad (3.6)$$

It can be further rearranged into the following form:

$$\begin{aligned}
xm_{11} + ym_{12} + zm_{13} + m_{14} - uxm_{31} \\
- ym_{32} - zm_{33} - um_{34} &= 0 \\
xm_{21} + ym_{22} + zm_{23} + m_{24} - vxm_{31} \\
- ym_{32} - zm_{33} - vm_{34} &= 0
\end{aligned} \tag{3.7}$$

Assuming that we have collected totally n correspondences, $(\mathbf{x}_i^p, \mathbf{X}_i^c, \mathbf{x}_i^c), i = 1 \dots n$, we rearrange all the equations to a linear system of the form $\mathbf{G}\mathbf{m} = \mathbf{0}$, where \mathbf{G} is a $2n \times 12$ matrix, \mathbf{m} is a 12×1 vector arrangement of the rows of the projection matrix to estimate. There are totally 12 variables, so $n \geq 6$ correspondences are enough to solve it. We obtain a solution which introduces the least error using Singular Value Decomposition (SVD). Moreover, in order to compensate labeling errors and obtain a stable solution, we take following steps: first, we use a RANSAC scheme in our algorithm. For each run of RANSAC, we randomly select 6 correspondences to estimate the projection. The criterion for admitting an inlier is that its back-projection error is below 10 pixels. Second, a fine adjustment is carried out on the RANSAC result. It minimizes

the following sum of the squared back-projection errors:

$$\sum_{i=1}^n \left(u_i - \frac{m_{11}x_i + m_{12}y_i + m_{13}z_i + m_{14}}{m_{31}x_i + m_{32}y_i + m_{33}z_i + m_{34}} \right)^2 + \left(v_i - \frac{m_{21}x_i + m_{22}y_i + m_{23}z_i + m_{24}}{m_{31}x_i + m_{32}y_i + m_{33}z_i + m_{34}} \right)^2 \quad (3.8)$$

Taking the RANSAC solution as the initialization, we use the Levenberg-Marquardt algorithm [29] to minimize the error. With these strategies, the accuracy of the estimated projection matrix is further improved.

3.3.2 Calibration of the Wiimote camera pair

The calibration of the projection matrix M_c^w can be done in a similar way. First, we calibrate the intrinsic parameters of the Wiimote. Since the four co-planar IR LEDs can be viewed as four corners of a calibration board, we can also use the OpenCV toolbox to calibrate the intrinsic parameters of the Wiimote. Second, according to the projective geometry in Fig. 3.3, each IR LED on the sphere L^c , its position in the Wiimote I^w and its position in the camera I^c , form a correspondence. To collect a number of correspondences, we place the sphere to different positions within the common field of view of the camera and Wiimote. Four correspondences can be obtained at each position. The locations of the IR LEDs in the Wiimote images can be directly detected by the Wiimote and their positions in the

camera images are manually labeled. Their 3D coordinates in the Wiimote coordinate are calculated by a Perspective 4 Points (P4P) algorithm given the configuration of the four IR LEDs and the intrinsic parameters of the Wiimote. In our implementation, the P4P algorithm proposed by Zhang[61] is employed. After all correspondences are obtained, the projection matrix M_c^w can be estimated in the same way as estimating M_p^c .

3.4 Sphere Detection and Tracking

The tracking module detects and tracks the translation of the sphere relative to the camera. The translation of the sphere is defined as the position of the sphere's center in the camera coordinate, which can be located via its projection (conic section) in the camera. However, it is difficult to define the rotation since the sphere is centrisymmetric in shape. So we embed four LEDs on the surface of the sphere and employ a Wiimote to track them. Based on the tracking result from Wiimote and the calibrated relationship between the Wiimote and the camera, we can calculate the rotation of the sphere relative to the camera. The configuration is user-friendly in appearance since the IRLEDs embedded are almost invisible. Experimental results also show that it achieves quite good accuracy and robustness.

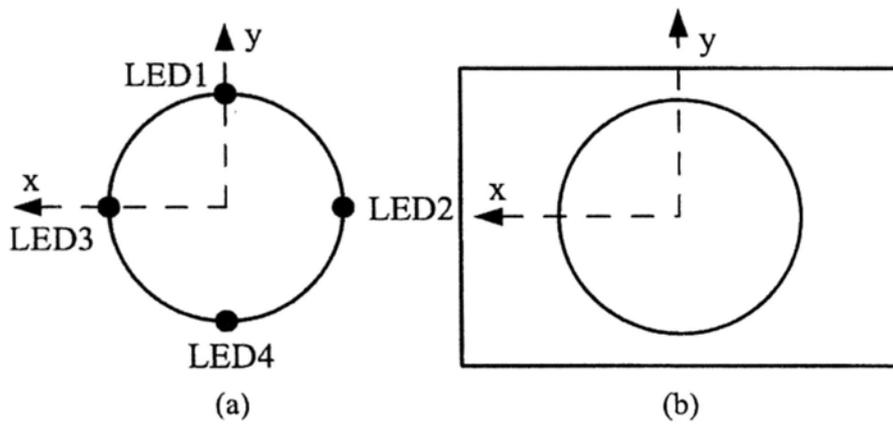


Figure 3.5: The definition of the object coordinate. (a) In the Wiimote configuration, the origin of the object coordinate is defined as the center of the sphere and its x-y plane is parallel to the plane formed by the four LEDs. (b) In the cardboard configuration, the origin of the object coordinate is defined as the center of the sphere and its x-y plane is parallel to the cardboard.

3.4.1 The Wiimote tracking

Detection

We first use the Hough transform circle detection algorithm to detect a circle to approximate the conic section in the initial frame of the video stream, and employ the algorithm in [57] to locate the center of the sphere. The rotation of the sphere is defined as follows: we define an object coordinate in the center of the sphere. The x-y plane of it parallels the plane formed by the four IR LEDs. An illustrative figure is shown in Fig. 3.5 (a). The rotation of the sphere is then defined as the rotation from the object coordinate to the camera coordinate. We calculate it as follows: given the detected positions of the four IR LEDs in the Wiimote, we calculate their 3D coordinates

in the Wiimote coordinate using the P4P algorithm in [61]. The four 3D points are then projected to the camera image plane by the projection matrix M_c^w . Finally we can calculate their 3D coordinates in the camera coordinate using Eq. (3.4) since we have located the center of the sphere. Assuming that they are $L_i^c, i = 1 \dots 4$, we can obtain the base vectors of the object coordinate and the rotation matrix from the object to the camera by:

$$\mathbf{i} = \frac{\mathbf{L}_3^c - \mathbf{L}_2^c}{|\mathbf{L}_3^c - \mathbf{L}_2^c|}, \quad \mathbf{j} = \frac{\mathbf{L}_1^c - \mathbf{L}_4^c}{|\mathbf{L}_1^c - \mathbf{L}_4^c|}, \quad \mathbf{k} = \mathbf{i} \otimes \mathbf{j}$$

$$\mathbf{R}^c = [\mathbf{i}, \mathbf{j}, \mathbf{k}] \quad (3.9)$$

We further refine the translation and rotation by minimizing the following squared errors:

$$\sum_{i=1}^4 \|\mathbf{R}^c \mathbf{L}_i^o + \mathbf{t}^c - \mathbf{L}_i^c\|_2^2 \quad (3.10)$$

where $L_i^o, i = 1 \dots 4$ are the 3D coordinates of the four IR LEDs in the object coordinate, which are manually measured.

Tracking

After detecting the translation and rotation of the sphere in the initial frame, we track them in the subsequent frames. The tracking state is the concatenation of the rotation and translation vector in the

following form:

$$s = \left(r_x \ r_y \ r_z \ t_x \ t_y \ t_z \right) \quad (3.11)$$

where r_x, r_y, r_z are the Euler angles and t_x, t_y, t_z are the translations along the x, y and z axis respectively.

Particle filter[1] is employed to estimate the posterior density of the pose. It represents the pose as a set of discrete particles. Each particle has a weight to indicate how confident it is to represent the pose. The two main components of a particle filter are the state dynamic model and the observation model. The state dynamic model determines how the particles propagate from frame to frame. The observation model determines how much weight is assigned to particles providing the observation at that frame. The workflow of the particle filter used in our system is shown in Fig. 3.6. We describe the state dynamic model and observation model as follows:

state dynamic model Since the sphere is freely moved, a simple random walk model based on a uniform density U about the previous state is used. The variable e represents the uncertainty about the movement of the sphere.

$$p(s_k | s_{k-1}) = U(s_{k-1} - e, s_{k-1} + e) \quad (3.12)$$

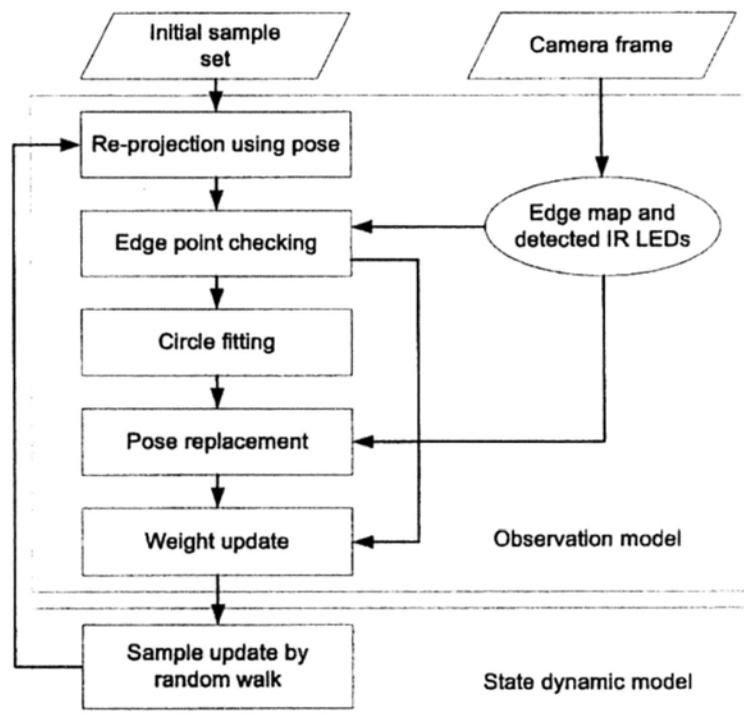


Figure 3.6: The flow diagram of the proposed particle filter algorithm.

observation model The observations in our algorithm are the edge map obtained by Canny edge detector and the detected position of the four IR LEDs. To evaluate the likelihood of each particle, we first re-project the sphere and the four IR LEDs to the image plane according to the pose represented by the particle. The projected sphere is approximately a circle. Then we check how many edge points are on the circle. An edge point is regarded as on the circle if its distance to the circle is within 5 pixel. For each degree of the 360-degree circumference, we check if there is an on-edge point. If the number of the on-edge points is less than 90, i.e, one fourth of the complete circumference, we judge that the on-edge points are not enough to match a valid circle, and a very low likelihood is assigned to this particle. Otherwise, we fit a circle centered close to the projected circle based on these on-edge points. Its fitting rate (the ratio of the inliers to the total on-edge points) is assigned to the particle as its likelihood. To give a more precise tracking result, we introduced a replacement scheme into our observation model. For particle whose fitting rate is above a threshold (0.6 in our implementation), we relocate the center of the sphere based on the fitted circle and recalculate its rotation based on the detected IR LEDs, and replace the translation and rotation of the particle with the calculated result. In this way, each particle surviving from the evaluation procedures will represent a good approximation of the real sphere in the

scene.

3.4.2 Alternative: tracking with cardboard

The Wiimote tracking configuration works quite well in our experiments. However, it requires an extra Wiimote as the tracking device, and also a calibration step before using the Wiimote. Alternatively, we develop another configuration to track the rotation of the sphere which requires no extra tracking device. Our method is to encompass the sphere in the center of a rectangular cardboard. The rotation of the sphere is then defined as the orientation of the cardboard and we track it based on its projection in the camera. The object coordinate is defined in the center of the sphere, and the x-y plane of it parallels the cardboard. An illustrative figure is shown in Fig. 3.5 (b). We then track the sphere and the cardboard as a whole to calculate its translation and rotation.

Detection

We first use the method proposed in [28] to detect the cardboard. The quadrangle detected is then used to calculate both the translation and orientation of the cardboard using the pose estimation algorithm proposed in [61]. After that, the sphere is projected to the image using the calculated pose. We evaluate the likelihood of the projected sphere using the method introduced in last section. If the likelihood

is above a threshold, the cardboard and sphere are correctly detected.

Tracking

The pose is also tracked using particle filter. The work-flow and the dynamic model of the particle filter are almost the same as mentioned above. The main difference lies in the observation model, i.e., how to evaluate the likelihood of the particle. The observation is the edge map obtained by Canny edge detector and the line segments detected by Hough transform. We re-project the cardboard and the sphere to the image according to the pose represented by the particle, and evaluate its likelihood based on the edge map and the line segments. The likelihood is set to the sum of two parts, the matching rate of cardboard to the line segments, and the likelihood of the sphere. The likelihood of the sphere is calculated using the method introduced in last section. The matching rate of the cardboard is discussed in detail in [28]. If both parts are above a threshold, we conclude that the cardboard and sphere are matched correctly with a high confidence. We then calculate the pose based on the matched cardboard and sphere, and replace the pose of the particle with it.

3.4.3 Extension of the configurations

In the current implementation, we only embedded four IR LEDs on the sphere to demonstrate the idea. The rotation of the sphere is lim-

ited to a range that the four IR LEDs are visible to the Wiimote. In principle, we can enlarge the range of the rotation by adding more IR LEDs and choosing four of them to track at each time instant by turning on the four and turning off the others using a switching circuit. Since the physical configuration of all LEDs is fixed and can be measured manually, the 3D coordinates of all IR LEDs with respect to the Wiimote can be obtained based on the tracked four IR LEDs via the P4P algorithm. Therefore, we can judge whether the four working LEDs are going out of the field of view of the Wiimote. If not, we simply keep the working LEDs. Otherwise, we replace them with four LEDs closest to the center of the field of view of the Wiimote and turn on them for the next time instant. Depending on the distribution of the LEDs, usually only one or two LEDs are needed to be turned on/off at one replacement. At the beginning of the tracking, we can arbitrarily turn on four neighboring LEDs as the initial working LEDs. By this mechanism, the rotation range can be extended to omni-view of the sphere in theory. As for the cardboard configuration, the range of rotation is limited since the sphere may occlude the edges of the cardboard under big rotations and thus fail the tracking. A possible solution is to use a semi-sphere, i.e., to cut the half sphere at the back of the cardboard, and place a facing-up camera on the floor to track the cardboard. The cardboard would not be occluded by the sphere and always be tracked in this configura-

tion as long as the cardboard is not reversed. This method can extend the range of rotation to half omni-view of the sphere in theory.

3.5 Movement and View Dependent Projection

From the tracking algorithm, we have known the relative pose of the sphere to the camera at each frame. In order to make correct projection, we also have to know the position of the viewer's head. In our system, we simply allocate a fixed position where the viewer can view the projection correctly. We refer this position as the view spot. We discuss how to allocate the view spot and generate the view and movement dependent projection.

3.5.1 View spot allocation

The allocation of the view spot essentially equals to finding the 3D location of the view position relative to the camera. Our method is to place another camera (referred as the view camera) at the view spot. By calibrating this view camera and the tracking camera, we know the geometric relationship between them. Similarly, without the need to explicitly know the relative pose between them, we simply estimate a projection matrix from the tracking camera to the view camera. The calibration method is similar. We use the sphere as the calibration object, and project a cross to the sphere surface. The crosses observed by the two cameras form a correspondence. The

projection matrix is then estimated via a number of such correspondences.

3.5.2 Projection image warping

Now, the translation \mathbf{t}^c and rotation \mathbf{R}^c of the sphere relative to the tracking camera, the projection matrix from the tracking camera to the eye of the viewer (the view camera), denoted as \mathbf{M}_e^c , are all obtained. We are ready to generate the movement and view dependent projection. The projection model from the display object to the eye of the viewer can be given by:

$$s\tilde{\mathbf{x}}^e = \mathbf{M}_e^c \begin{pmatrix} \mathbf{R}^c \mathbf{V}^o + \mathbf{t}^c \\ 1 \end{pmatrix} \quad (3.13)$$

where \mathbf{V}^o is a vertex of the object, \mathbf{x}^e is the projection of the vertex in the retinal image of the viewer.

The next step is to warp the projection image. The light path among the object, the projector, and the viewer are shown in Fig.3.7. An intuitive way to generate the projection image is, for each vertex of the object in the object coordinate, first to find the intersecting point of the light path $\overrightarrow{\mathbf{V}^o \mathbf{O}^e}$ and the sphere surface, say X^c , and then project it to the projector image pixel \mathbf{x}^p . In principle, these three points should have the same color. However, this procedure may

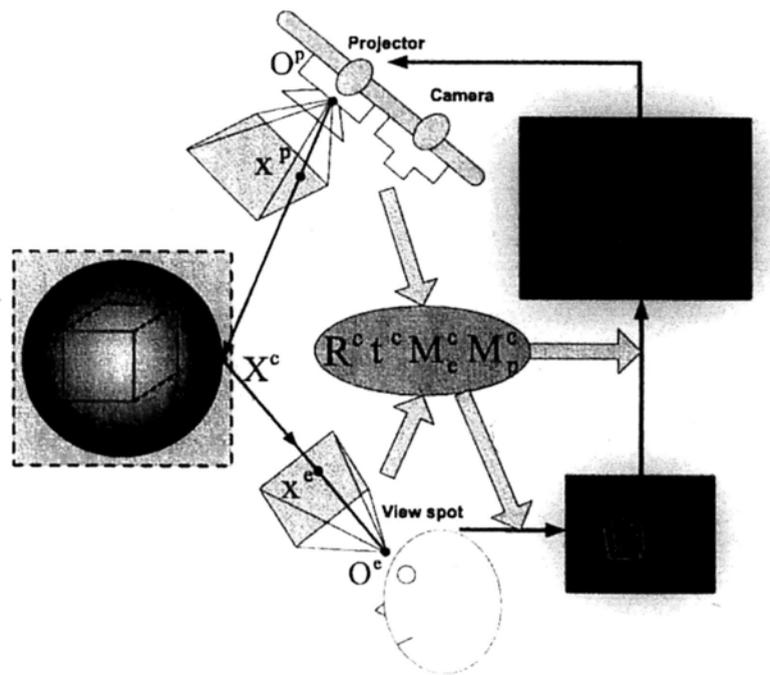


Figure 3.7: Movement and view dependent projection.

cause some pixels of the projection image not covered, i.e, cause some holes in the projection image. To overcome this problem, we invert the procedure. We first project all vertices of the object to the eye using Eq. (3.13). Then, for each pixel in the projection image \mathbf{x}^p , we find its correspondence, \mathbf{X}^c , i.e, the intersection point on the surface of the sphere. The intersection point \mathbf{X}^c can be found by solving the following equations:

$$\begin{aligned} s\tilde{\mathbf{x}}^p &= \mathbf{M}_p^c \tilde{\mathbf{X}}^c \\ \|\mathbf{X}^c - \mathbf{t}^c\|_2^2 &= R^2 \end{aligned} \quad (3.14)$$

If the equations have a solution, we project it to the eye to obtain \mathbf{x}^e , and set the color of \mathbf{x}^p to that of \mathbf{x}^e . Otherwise, it means \mathbf{x}^p has no correspondence point on the sphere surface. We set its pixel color to 0 in this case.

3.6 Implementation and Results

We have built a prototypical system with the following devices: an off-the-shelf projector with resolution of 1280×1024 , two Logitech Quickcam Pro 4000 webcams with resolution of 320×240 (one as the tracking camera, and the other as the view camera), a Nintendo Wiimote, and two foam spheres with radius of 150 mm (one for the Wiimote configuration, and the other for the cardboard configura-

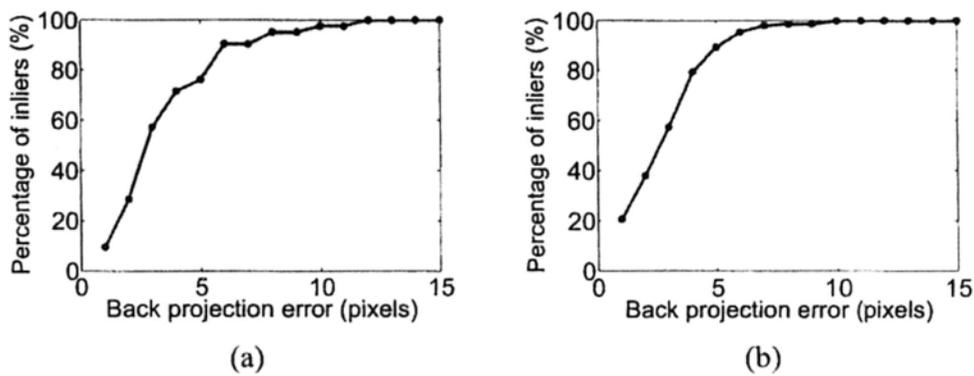


Figure 3.8: (a) The accuracy of the projector-camera pair calibration. (b) The accuracy of the Wiimote-camera pair calibration.

tion). The length of the arc between the diagonal LEDs is 160 mm. The size of the cardboard encompassing the sphere is 455×370 mm. A dual core 2.16GHz PC with 1GB memory is used as the testing platform. Since we are not using any expensive devices, the cost of our system is quite low (about 4000 HK\$ not including the PC).

System calibration To calibrate the projector-camera pair, we place the sphere at several positions to collect enough correspondences. About 10~30 correspondences are collected at each position and totally 186 correspondences are collected. We run RANSAC estimation for 10000 iterations and it takes about 1 minute. The accuracy of the estimated projection matrix is measured by the error between the labeling points and their back-projections. We evaluate the distribution of the back-projection errors, which is the percentage of the points with back-projection error below some pixel levels (in-

liers) among all labeled points. The distribution is shown in Fig. 3.8 (a). The mean back-projection error is 5.3 pixels.

For the Wiimote-camera pair, we similarly change the position of the sphere to collect enough correspondences. Totally 160 correspondences are collected in our experiment. We also evaluate the distribution of its back-projection errors, which is shown in Fig. 3.8 (b). The mean back-projection error is 4.2 pixels.

The calibration error mainly comes from two sources: the error in locating the image correspondences and the linear projection model we assumed. Carefully labeling a large number of correspondences (at least 100) is crucial to obtain good results.

During the experiment, we also notice that the sphere should be placed to many different positions in order to obtain a stable calibration. For the projector-camera pair, normally more than 4 different positions are needed and for the Wiimote-camera pair, more than 10 positions are needed. The possible reason for this may be that correspondences collected at the same position have little difference in depth (all at the sphere surface), so the estimation result overfits these correspondences but may not fit correspondences in other depths.

Sphere detection and tracking We test the accuracy and robustness of the tracker in tracking the translation and rotation of the sphere under different movements. A video sequence of 339 frames contain-

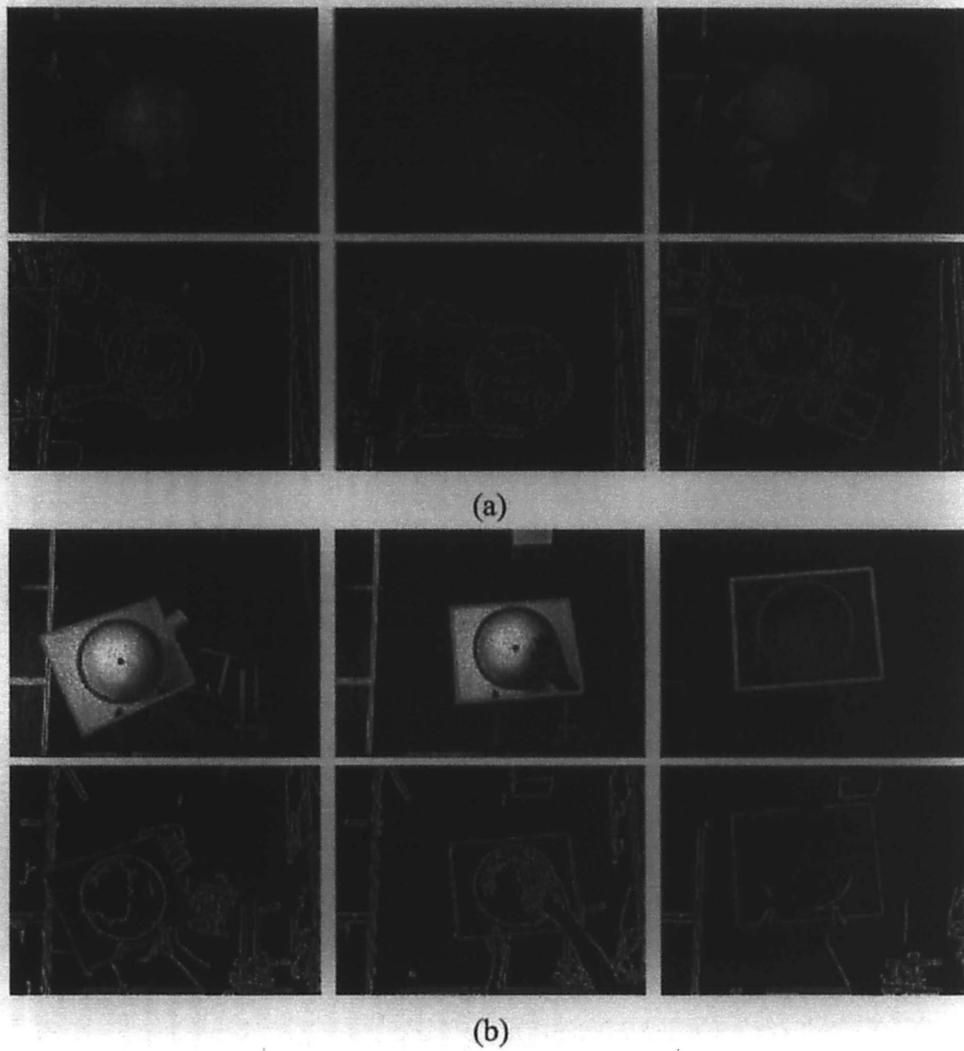


Figure 3.9: Some frames of the tracking results with (a) the Wiimote tracking configuration and (b) the cardboard tracking configuration.

Table 3.1: Accuracy of the tracking with two configurations

Configuration	Center (pixels)	Radius (pixels)	IR LEDs or Corners (pixels)
Wiimote	2.8	2.9	3.6
Cardboard	3.4	4.0	4.8

ing translation, rotation and free movements is recorded to evaluate the tracking accuracy. We manually label the center and the radius of the circle, as well as the positions of the four IR LEDs. The accuracy of the tracked circle and the IR LEDs is defined as the error between the tracked positions and the manually labeled ground-truths. Similarly, for the cardboard configuration, we record a video sequence of 316 frames to evaluate its tracking accuracy, which is defined as the tracking errors of the circle and the cardboard. Table 3.1 lists the mean tracking errors in both configurations. From the table, we can see that the Wiimote configuration achieves a better tracking accuracy. It is reasonable since the Wiimote and IR LED tracking is more accurate and robust than the cardboard tracking, though at the expense of an additional Wiimote and a calibration step.

We also test the performance of the trackers under different backgrounds, including lighting changes, partial hand occlusion and dense clutter. Fig. 3.9 shows some frames extracted from the tracking process. For illustration purpose, in the Wiimote configuration, the tracked sphere is re-projected to the image in red and the four tracked

IR LEDs in the camera are marked in green. In the cardboard configuration, the tracked sphere is also shown in red, and the cardboard is shown in green. The edge maps in both configurations are also shown (the line segment features in the cardboard configuration are shown in red). Experiments show that both trackers can tolerate certain amount of disturbance.

Display results A 3D face model obtained from the USF Human ID 3-D database [5] is used to test the projection performance. We test it under different types of motions of the sphere, including pure rotation, and free movement. Fig. 3.10 shows some result frames of the face in free movement. The left column shows the projection images and the right column shows the corresponding display results. Fig. 3.11 shows some display results with the cardboard configuration. In all of our experiments, our system can track the sphere and generate the projection image with satisfactory accuracy and robustness, and the face can be displayed onto the sphere with desired effect. More results can be found in the supplementary video.

Speed In the 2.1GHz CPU and 1GB memory platform, our system can achieve real-time processing smoothly in both configurations. The cardboard configuration is a little slower (about 18 fps) than the Wiimote configuration (about 20 fps) because it needs to evaluate the likelihood of both the sphere and the cardboard. The whole

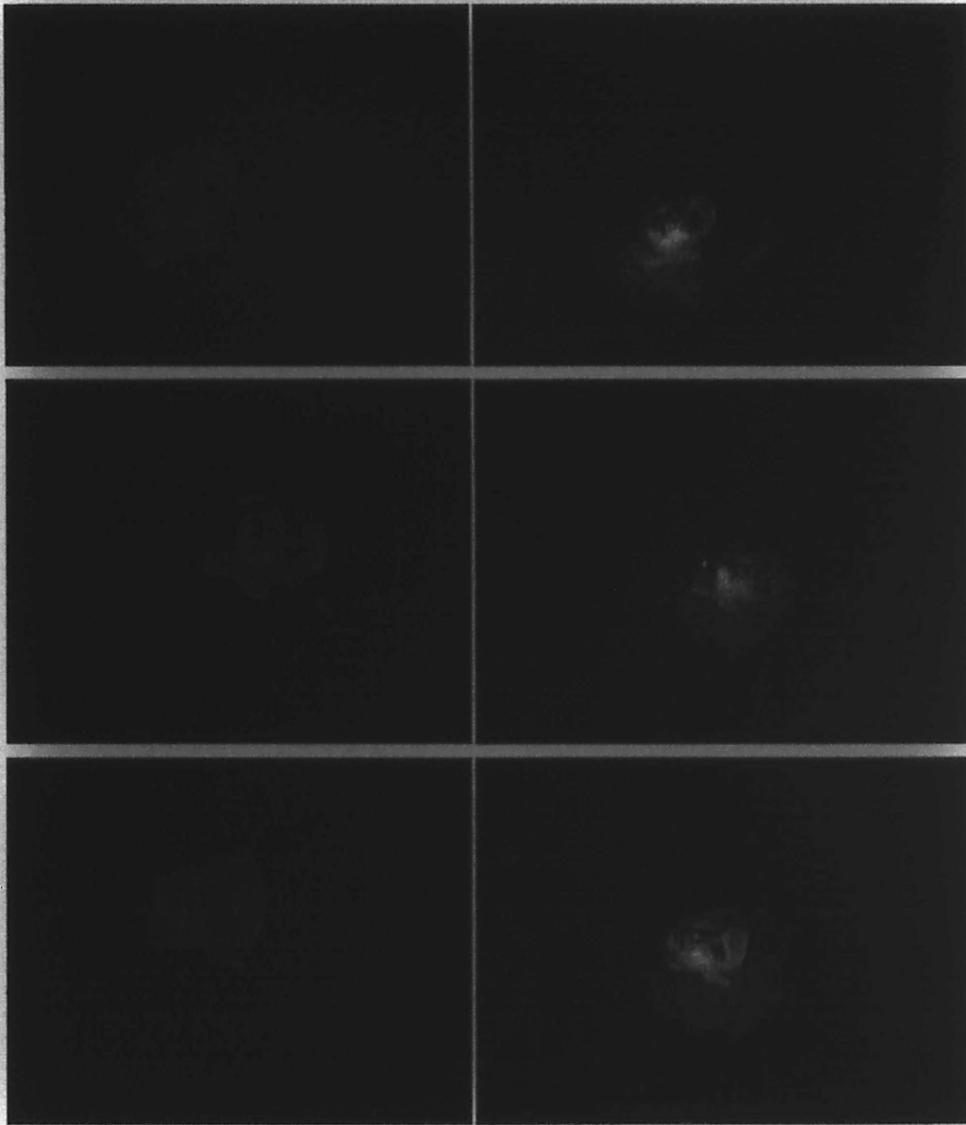


Figure 3.10: Some frames of the projection results of a 3D face with the Wiimote configuration. The left column shows the projection images and the right column shows the display results.

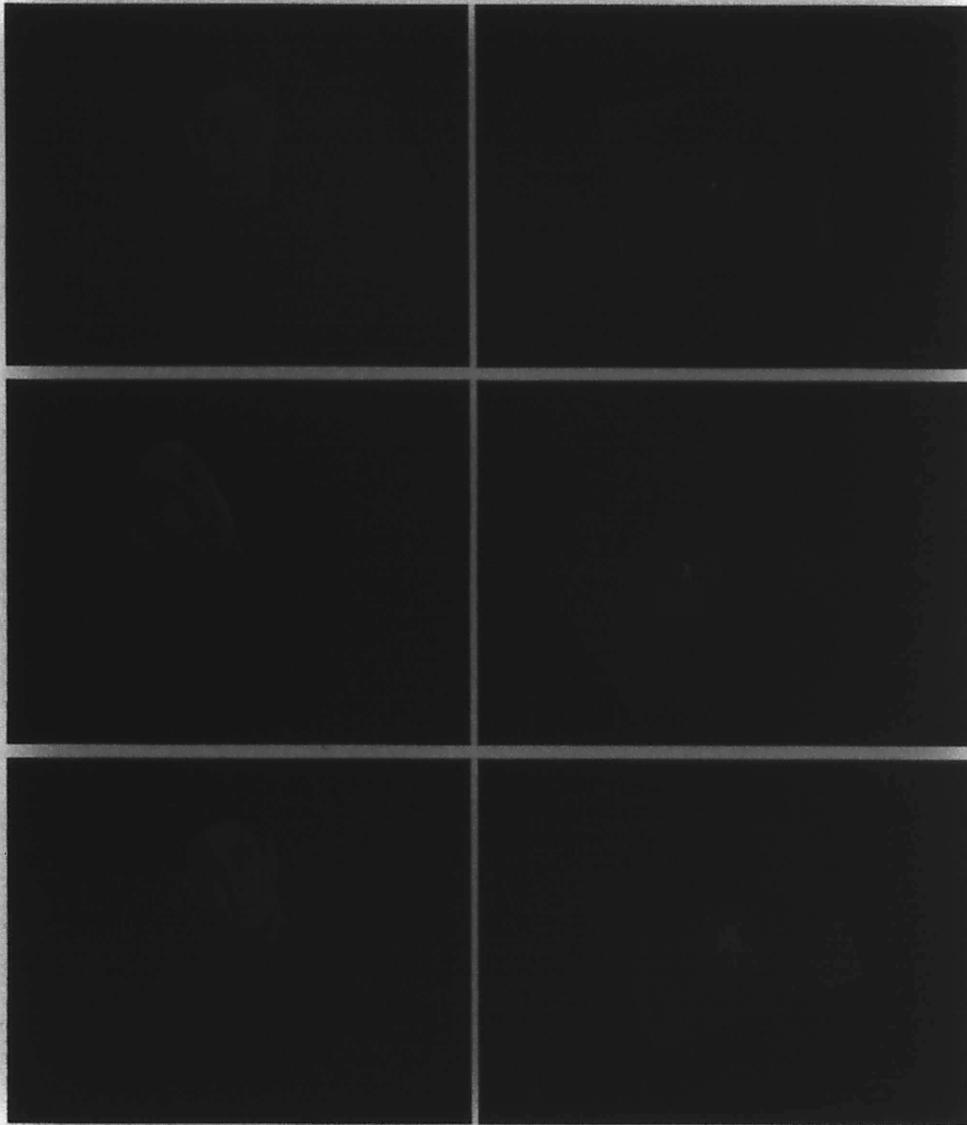


Figure 3.11: Some frames of the projection results of a 3D face with the cardboard configuration.

running time mainly distributes in feature detection, particle filter tracking and the projection image warping. Table 3.2 shows the partition of the running time in both configurations. The particle filter tracking consumes the major part of the time. It varies with the number of particles used. Fig. 3.12 shows the tracking time against the number of particles in the Wiimote configuration. Approximately, the tracking time increases linearly with the number of particles. In our system, the number of particles is set to 80 and 60 in the two configurations respectively. The maximum number of line segment features is also an influence on the processing time in the cardboard configuration. We fix it at 20 in our experiments.

Table 3.2: Running time per frame with two configurations

Process	Wiimote	Cardboard
Feature detection	about 5 ms	about 10 ms
Particle filter tracking	30 ~ 35 ms	40 ~ 45 ms
Projection image warping	about 5 ms	about 5 ms

3.7 Discussions

There are several limitations of our system. First, there is limitation on the resolution of the projection since the projection image is inevitably down-sampled when the sphere is away from the projector. Such limitation makes the small details of the displaying object

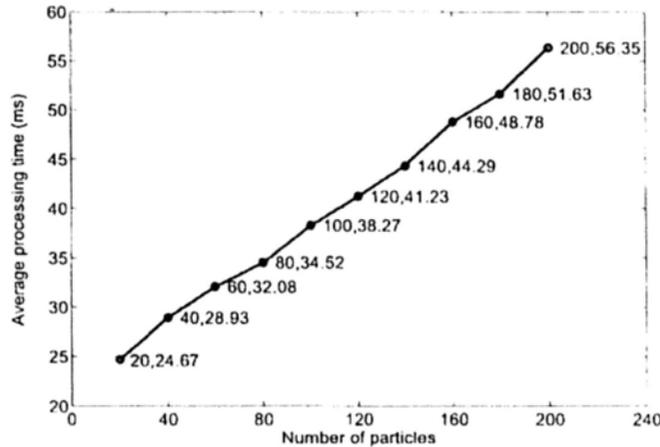


Figure 3.12: The processing time of the particle filter tracking algorithm against the number of particles.

unobservable or blurred. Second, the depth field of the projector is another problem. We use a single projector with quite limited depth of field, making the projection in focus only within a particular range of depth. Third, the display result has latency (about 160ms), especially when the sphere is moving quickly. One cause to the latency is the tracking algorithm, and another is the physical latency of the projector and camera. High quality projector and camera may be a solution to alleviate the problems mentioned above. However, the cost will be increased.

□ **End of chapter.**

Chapter 4

Projection On A Flexible Screen

In this chapter, we propose a low-cost hand-held flexible display system which employs a projector to project display information onto a hand-held flexible surface (e.g. an ordinary white paper) that can be twisted freely. While the user twists the projection surface, the system projects well-tailored information onto the surface corresponding to the deformation so that the viewer sees the information as if it was printed on the paper. The ultimate goal is to develop an interactive viewing tool for displaying content on flexible surface that can be deformed by the user, i.e., when the user twists the paper, the display content on the paper deforms simultaneously. This system has a lot of potential in the entertainment and education fields. A pair of cameras is employed to track the pattern printed on the back of paper. The cameras and the projector are calibrated off-line via a simple and convenient method. A real-time algorithm is proposed to recover the 3D surface of the paper. The display content is then pre-

warped according to the recovered surface and projected onto the front of the paper. Two demonstrative applications are elaborated to illustrate the potential of the proposed system. Our system is easy to set up and runs in real-time. Experimental results show that the flexible display is created with satisfactory accuracy and robustness. This work is published in [33].

4.1 Introduction

Traditional display systems usually display information on static flat monitors and the viewer interacts with it using indirect pointing devices such as keyboard and mouse. The shape of the screen is fixed and the control of the display such as the viewing angle is limited. As display technology is widely used in different disciplines, static display technology is not sufficient for many emerging applications. For example, in the medical field, a common way for clinicians to analyze medical volumetric data such as MRI and CT is to view the cross-sectional slices of the data obtained. With a traditional static display system, the slices can only be displayed on a fixed screen and the interaction is achieved via keyboard and mouse; the user's viewing experience is limited and the interaction is unnatural. An alternative way is to interact with these cross-sectional slices directly using a projector and a hand-held screen. The projection frustum forms a virtual object model in space, the user inserts the portable

screen into the frustum and the corresponding cross-sectional slice of the volumetric data is displayed. The user can observe any cross-sectional slices he wants. This mobile type of display provides the user with an immersive experience and a more natural and direct way of interaction. Also the human internal organs are usually not planar but curved, so a flexible slicing tool that not only displays planar slices but also curved slices may help the doctor diagnose the disease. Though one planar slicing tool [10] and several flexible display systems like [26] [27] have been proposed, they allow none or very limited deformation, and some of them require attaching sensors to the surface. These reasons motivate us to develop a more flexible, low cost, easy-to-setup and real-time hand-held display system. Such a system is expected to have a lot of potential in practice, not only in medical fields but also in entertainment or manufacturing industries. For example, it can be used as a model-preview tool in design to preview the appearance of flexible models and how it can be twisted by users before it is put into production.

The proposed system is based on computer vision technology and the devices used consist of a projector and three webcams. Neither special hardware nor sensors are needed. The configuration of our system is shown in Fig. 4.1. The projector and one webcam are fixed on a rig and another two webcams are placed on the floor. An ordinary white paper with printed checker pattern on the back is used as

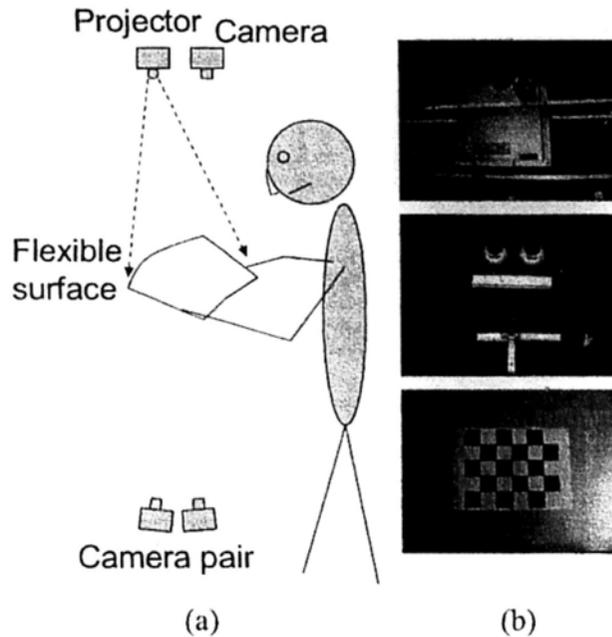


Figure 4.1: The configuration of our system: (a) The overall diagram of our system; (b) The facing down projector and camera, the facing up tracking camera pair, and an ordinary paper with checker pattern printed on the back as the flexible surface.

the projection screen. When the user manipulates the paper screen within the viewing field of the projector and the cameras, the camera pair on the floor tracks the checker pattern and the projector will project pre-warped images onto the top of the paper. The facing down camera on the top is included as an observation camera when calibrating the projector since the facing up cameras on the floor cannot observe the projection result. If we reuse one of the tracking cameras as the observation camera, the total number of cameras needed is actually two.

The potential use of the proposed system is demonstrated with

two applications. The first one is a flexible image projection application that can be used as a model previewing tool to view different appearances of curved model surfaces. The second one is a flexible slicing tool that not only views planar but also curvilinear cross-sections of medical volumetric data. The use of the system is not limited to these two examples. Other applications can be implemented using the similar method depending on one's imagination.

The major contribution of this work is the proposal of a new flexible display system and an effective approach to realize it with several low-cost and off-the-shelf devices. The advantage of the proposed method mainly lies in a well-designed calibration method and an efficient algorithm to track and recover the deformation of the flexible surface. The rest of this chapter is organized as follows: we first give an overview of the system architecture in Section 4.2. In Section 4.3 we introduce the calibration of the system. In Section 4.4 we describe how to track and recover the surface of the paper. In Section 4.5 we present some potential applications of our system. Implementation and experimental results are detailed in Section 4.6. We discuss this chapter in Section 4.7.

4.2 System Overview

Our system is an integration of three parts. Fig. 4.2 shows the overview of the system. The first part is the calibration module.

In order to recover the flexible surface and guide the projection to fit the deformation, we need to calibrate the geometric relationships among the two tracking cameras and the projector. In our approach, it is unnecessary to explicitly estimate the relative poses between them. Instead we simply estimate two projection matrices. A simple and convenient calibration method is proposed. The second part is tracking and recovering the surface of the paper. To simplify the tracking process and enable real time recovery, a checker pattern is printed on the back of the paper and a stereo camera pair is used. We track corners of the checker pattern and recover their 3D positions. The surface of the paper is then approximated by a triangulated mesh of the 3D corners. We employ a flexible triangulation model that can enhance the deformation ability of a fixed triangulation model. The last part is to project the display content onto the paper. Based on the calibrated projection matrix and the recovery result, the display content is pre-warped to fit the surface of the paper. In Section 4.3, 4.4, and 4.5, we describe each module in detail.

4.3 System Calibration

The calibration step finds the geometric relationships among the two tracking cameras and the projector. In our approach, we first calibrate one tracking camera using the OpenCV toolbox [24] and choose it as a reference camera. Then we calibrate two geometric re-

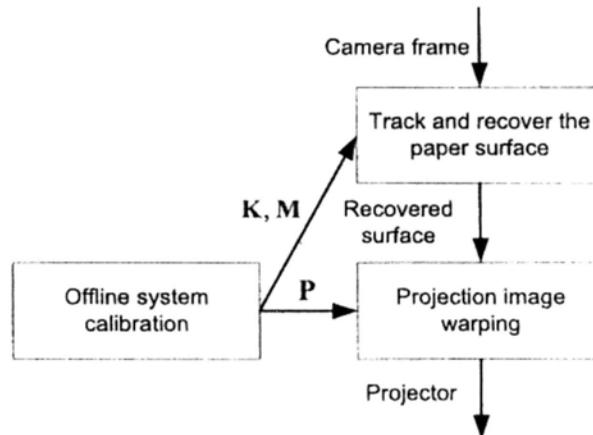


Figure 4.2: The overview of our system.

relationships, one between the reference camera and the projector and the other between the two tracking cameras. Without the need to know explicit geometric parameters, we simply estimate two projection matrices, one from the 3D camera coordinate of the reference camera to the projector image plane, and the other to the second tracking camera image plane. Both projection matrices are constant and independent from the deformation of the paper. While designing the calibration method, we keep in mind that the calibration process should be as easy as possible since the layout of the cameras and the projector might change frequently in practice. It should not take the user too much time and labor to calibrate the system.

4.3.1 The projective model

The projective model of the projector is similar to the camera model except for the projection direction. The projection from a 3D world

point to the 2D projector image pixel is related by a 3×4 perspective projection matrix. We assume that the world coordinate system is identical to the reference camera coordinate system here. Then any 3D point in the reference camera coordinate system, for example a point $\mathbf{X}(x, y, z)$ on the paper, corresponds to its projector pixel $\mathbf{x}(u, v)$ via a projection matrix \mathbf{P} :

$$s\tilde{\mathbf{x}} = \mathbf{P}\tilde{\mathbf{X}} \quad (4.1)$$

and

$$\mathbf{P} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \quad (4.2)$$

where $\tilde{\mathbf{x}}$, $\tilde{\mathbf{X}}$ are the homogenous coordinates and s is a scale factor.

Similarly, the 3D point \mathbf{X} in the reference camera coordinate system, and its projection \mathbf{y} in the second tracking camera are also related via a 3×4 projection matrix \mathbf{M} :

$$s\tilde{\mathbf{y}} = \mathbf{M}\tilde{\mathbf{X}} \quad (4.3)$$

and

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \quad (4.4)$$

The target of the calibration is then subject to estimate the two projection matrices \mathbf{P} and \mathbf{M} .

4.3.2 Calibration method

To estimate the projection matrix \mathbf{P} , the main idea of the proposed method is to collect a number of correspondences between the 3D points in the reference camera coordinate and their 2D projections in the projector image. The collecting process is conducted as follows: we hold a thin cardboard with identical checker pattern printed on both sides between the projector and the reference camera. A cross with known position is projected to the top side of the cardboard. The reference camera and the observation camera can observe the checker pattern on each side but only the observation camera can observe the cross. When the user moves the cardboard slowly, our calibration program will detect the checker pattern in both cameras and the cross in the observation camera automatically. If two checker patterns and the cross are all detected, the program reports a correspondence and asks the user if it is acceptable. This allows the user

reference camera. Since the checker patterns on the two sides are identical, they are assumed to have the same 3D positions and thus the checker corners in the two cameras are related by a homography:

$$s\tilde{\mathbf{x}} = \mathbf{H}\tilde{\mathbf{y}} \quad (4.5)$$

and

$$\mathbf{H} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \quad (4.6)$$

where $\mathbf{x}(u, v)$ and $\mathbf{y}(\alpha, \beta)$ are the corresponding corners in the reference camera and observation camera respectively. The homography matrix has 8 unknowns (up to a scale factor) and four corresponding corners are enough to estimate it. Substituting each pair of corresponding checker corners into (5.3) and rearranging it to the form in (4.7), we can estimate the homography by Singular Value Decomposition (SVD). The image position of the cross in the reference camera is then calculated via the homography given the the detected image position of the cross in the observation camera.

$$\begin{pmatrix} \alpha & \beta & 1 & 0 & 0 & 0 & -u\alpha & -u\beta & -u \\ 0 & 0 & 0 & \alpha & \beta & 1 & -v\alpha & -v\beta & -v \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{pmatrix} = 0 \quad (4.7)$$

Next we calculate the 3D coordinates of the cross in the reference camera coordinate. We first calculate the 3D positions of the checker corners via the pose estimation algorithm[61] given the intrinsic parameters of the reference camera and the physical width of the checker. After obtaining the 3D positions of the checker corners, we can construct the planar equation of the cardboard:

$$ax + by + cz + d = 0 \quad (4.8)$$

where a, b, c, d are the coefficients of the planar equation. Since the 3D cross point is on the cardboard, it should satisfy the planar equa-

tion (4.8). Meanwhile, the 3D cross point and its 2D projection should satisfy the projection equation of the reference camera:

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (4.9)$$

and

$$\mathbf{K} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \quad (4.10)$$

where \mathbf{K} is the calibrated intrinsic parameter matrix of the reference camera and (u, v) have been obtained via the homography transformation in (5.3). From (4.8) and (4.9), we can solve the 3D coordinates of the cross point. Until now, we have obtained full information for each correspondence to estimate the matrix \mathbf{P} .

The projection matrix in Eq. (4.4) has 12 unknowns (up to a scale factor), so a minimum number of 6 correspondences are enough to solve it. Substituting each pair of the 2D and 3D coordinates of the cross into Eq. (4.1), we can obtain a solution using SVD in the same way we estimate the homography. In order to compensate for the detection errors of the cross and the checkers, a fine adjustment is car-

ried out. It minimizes the following sum of squared back-projection errors:

$$\sum_i \left(u_i - \frac{p_{11}x_i + p_{12}y_i + p_{13}z_i + p_{14}}{p_{31}x_i + p_{32}y_i + p_{33}z_i + p_{34}} \right)^2 + \left(v_i - \frac{p_{21}x_i + p_{22}y_i + p_{23}z_i + p_{24}}{p_{31}x_i + p_{32}y_i + p_{33}z_i + p_{34}} \right)^2 \quad (4.11)$$

Taking the SVD solution as initialization, we use the Levenberg-Marquardt method[29] to minimize the error. After this step, the accuracy of the estimated projection matrix is further improved.

The calibration of the projection matrix M can be done in a similar way. Each corner pair of the checker pattern in the two tracking cameras forms a correspondence. The calibration process is thus easier since the correspondence can be directly observed.

The proposed calibration approach is easy, flexible and automatic. The whole process involves little labor of the user. It takes about a few minutes to complete a whole calibration, including collecting the correspondences and estimating the two projection matrices.

4.4 Paper Surface Tracking and Recovery

We propose a real-time surface tracking and recovery algorithm to recover the 3D surface of the paper in each frame. In our implementation, to simplify the tracking and save computation time, a checker pattern is printed on the back of the paper, which can be easily de-

tected and tracked by the Lucas-Kanade tracker[7]. Moreover, the calibrated tracking camera pair is used to solve the depth ambiguity. The recovery task is then subject to recover the 3D positions of the checker corners in each frame based on their tracked image positions. Assuming there are totally n corners, and their tracked image positions in the two cameras are $\mathbf{x}_i(u_i, v_i), \mathbf{y}_i(\alpha_i, \beta_i), i = 1 \dots n$, the corresponding 3D positions $\mathbf{X}_i(x_i, y_i, z_i)$ are then the unknown variables to be estimated.

To allow more deformation freedom, we apply a flexible triangulation to the corners. In detail, each checker is triangulated through introducing a diagonal line. However, different from existing fixed triangulation models, the choice of which diagonal line to triangulate along is not fixed beforehand but to be determined during the recovery process. In other words, the model allows each checker to deform along either one of its two diagonal lines. The flexible triangulation model enhances the deformation ability of the fixed triangulation model, especially for our small size checker pattern. We introduce a variable ω for each checker to indicate along which diagonal to triangulate. The variable has two possible values, 1 or -1. The value 1 indicates the checker deforming along the left-top to right-bottom diagonal line while -1 indicates the deformation along the other diagonal line. Supposing there are m checkers in the pattern, the indication variables $\omega_i, i = 1 \dots m$ are also the unknowns

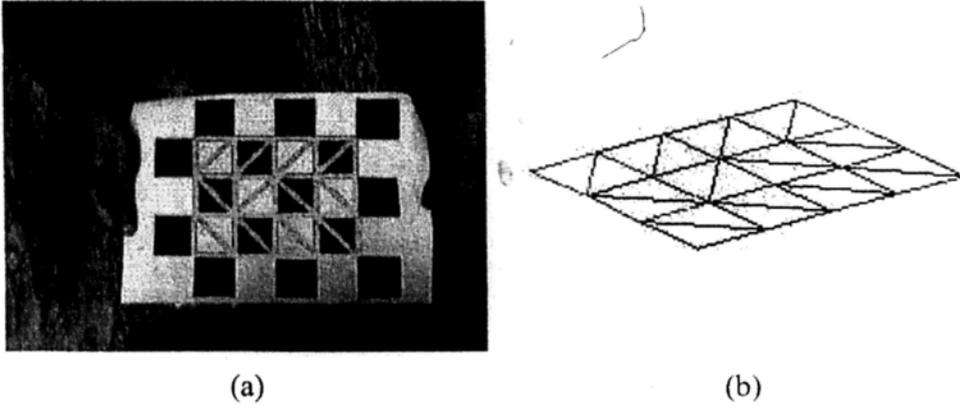


Figure 4.4: An example of the flexible triangulation of a 4×3 checker pattern. The triangulation in (b) is reprojected to the paper in (a).

to be estimated. An example of a flexible triangulation to a 4×3 checker pattern is shown in Fig. 4.4.

To solve the 3D positions and the indication variables, we minimize an energy function composed of two parts, the back-projection errors of the checker corners in two tracking cameras and a smoothness term to regularize the triangulation model. Substituting each corner into Eq. (4.9) and Eq. (4.3), we can formulate the energy function of back-projection errors as:

$$\begin{aligned}
 E_p = & \sum_{i=0}^n \left(\left(\frac{\mathbf{k}_1^T \mathbf{X}_i}{\mathbf{k}_3^T \mathbf{X}_i} - u_i \right)^2 + \left(\frac{\mathbf{k}_2^T \mathbf{X}_i}{\mathbf{k}_3^T \mathbf{X}_i} - v_i \right)^2 \right) + \\
 & \sum_{i=0}^n \left(\left(\frac{\mathbf{m}_1^T \tilde{\mathbf{X}}_i}{\mathbf{m}_3^T \tilde{\mathbf{X}}_i} - \alpha_i \right)^2 + \left(\frac{\mathbf{m}_2^T \tilde{\mathbf{X}}_i}{\mathbf{m}_3^T \tilde{\mathbf{X}}_i} - \beta_i \right)^2 \right)
 \end{aligned} \tag{4.12}$$

where $\mathbf{k}_1^T, \mathbf{k}_2^T, \mathbf{k}_3^T$ are three row vectors of \mathbf{K} and $\mathbf{m}_1^T, \mathbf{m}_2^T, \mathbf{m}_3^T$ are

three row vectors of M . To regularize the triangulation model, an intuitive idea is to preserve the the original edge length of each triangle. However, such a regularization term is difficult to optimize. Salzmann *et. al*[49] proposed to preserve the orientation of the edge in consecutive frames and obtained good results. We employ the same constraint in our formulation and formulate it as a quadratic term. Assuming that the surface $\{\mathbf{X}_i^t, i = 1 \dots n\}$ at time t is known, for each edge $\overline{\mathbf{X}_i \mathbf{X}_j}$ in the triangulation model, the edge orientation constraint is formulated as the difference of orientation between consecutive frames, namely:

$$\delta_{ij} = \|\mathbf{X}_i^{t+1} - \mathbf{X}_j^{t+1} - \theta_{ij}^t\| \quad (4.13)$$

and

$$\theta_{ij}^t = L_{ij} \frac{\mathbf{X}_i^t - \mathbf{X}_j^t}{\|\mathbf{X}_i^t - \mathbf{X}_j^t\|} \quad (4.14)$$

where L_{ij} is the original length of the edge. According to our triangulation model, there are two types of the edges. One is the side edge of the checker and another is the diagonal line. However, for the second type, we should choose the diagonal line to regularize according to the value of the indication variable. If it is 1, we constrain the left-top to right-bottom diagonal line; If -1 , we constrain the other diagonal line. The total smoothness term is the sum over

all edges, namely:

$$\begin{aligned} \mathbf{E}_r = & \sum_{k=1}^m \left(\frac{\omega_k + 1}{2} \delta_{ac}^2 + \frac{1 - \omega_k}{2} \delta_{bd}^2 \right) \\ & + \sum_{(i,j) \in \Omega} \delta_{ij}^2 \end{aligned} \quad (4.15)$$

where a, b, c, d are indices of the four corners of the k^{th} checker, ac and bd are the two diagonals, Ω is the set of side edges. The recovery is then subject to minimize the sum of the two energy functions.

$$\min_{\mathbf{X}, \omega_k} \mathbf{E}_p + \lambda \mathbf{E}_r \quad (4.16)$$

where λ is a weight of the smoothness term. There are totally $3n+m$ variables to be solved. Simultaneous minimization over \mathbf{X} and ω is difficult because the indication variables are discrete and there are enormous combinations of them even with a small size checker pattern. Our solution is to separate \mathbf{X} and ω and minimize over them alternately. Specifically, when minimizing over \mathbf{X} , we keep ω constant, and vice visa. The two minimization phases in one iteration are detailed as follows:

minimization over \mathbf{X} All \mathbf{X}_i are involved in both \mathbf{E}_p and \mathbf{E}_r . They are quadratic in \mathbf{E}_r but non-quadratic in \mathbf{E}_p . To simplify the optimization, we reformulate \mathbf{E}_p to quadratic form. The idea is to restrict the back-projection errors under a bound γ , and rewrite it to a

quadratic form:

$$\begin{aligned} & (\mathbf{k}_1^T \mathbf{X}_i - u_i \mathbf{k}_3^T \mathbf{X}_i)^2 + (\mathbf{k}_2^T \mathbf{X}_i - v_i \mathbf{k}_3^T \mathbf{X}_i)^2 \leq \gamma^2 (\mathbf{k}_3^T \mathbf{X}_i)^2 \\ & (\mathbf{m}_1^T \tilde{\mathbf{X}}_i - \alpha_i \mathbf{m}_3^T \tilde{\mathbf{X}}_i)^2 + (\mathbf{m}_2^T \tilde{\mathbf{X}}_i - \beta_i \mathbf{m}_3^T \tilde{\mathbf{X}}_i)^2 \leq \gamma^2 (\mathbf{m}_3^T \tilde{\mathbf{X}}_i)^2 \end{aligned} \quad (4.17)$$

E_p then becomes:

$$\begin{aligned} E_p = & \sum_{i=0}^n \left((\mathbf{k}_1^T \mathbf{X}_i - u_i \mathbf{k}_3^T \mathbf{X}_i)^2 + (\mathbf{k}_2^T \mathbf{X}_i - v_i \mathbf{k}_3^T \mathbf{X}_i)^2 \right. \\ & \left. - \gamma^2 (\mathbf{k}_3^T \mathbf{X}_i)^2 \right) + \\ & \sum_{i=0}^n \left((\mathbf{m}_1^T \tilde{\mathbf{X}}_i - \alpha_i \mathbf{m}_3^T \tilde{\mathbf{X}}_i)^2 + (\mathbf{m}_2^T \tilde{\mathbf{X}}_i - \beta_i \mathbf{m}_3^T \tilde{\mathbf{X}}_i)^2 \right. \\ & \left. - \gamma^2 (\mathbf{m}_3^T \tilde{\mathbf{X}}_i)^2 \right) \end{aligned} \quad (4.18)$$

and the minimization becomes:

$$\min_{\mathbf{X}_i} E_p + \lambda E_r \quad (4.19)$$

All terms in the total energy are quadratic, so the energy function can be easily minimized by solving the linear equations:

$$\frac{\partial (E_p + \lambda E_r)}{\partial \mathbf{X}_i} = 0 \quad (4.20)$$

minimization over ω The variables ω are only involved in the diagonal line of the regularization term. So we can ignore other terms.

The minimization becomes:

$$\min_{\omega_k} \sum_{k=1}^m \left(\frac{\omega_k + 1}{2} \delta_{ac}^2 + \frac{1 - \omega_k}{2} \delta_{bd}^2 \right) \quad (4.21)$$

Since each ω_k is independent, the optimization is actually a comparison of δ_{ac} and δ_{bd} for each checker. If δ_{ac} is smaller, ω_k is then set to 1. Otherwise, ω_k is set to -1.

The initial values of \mathbf{X} and ω are set to the result of previous frame. For the first frame, a tricky method is used. We require the paper in the first frame to be planar. So \mathbf{X} in the first frame can be obtained simply by the pose estimation algorithm[61]. For ω , we simply set all of them to 1 in the first frame. Although the paper may not be ideally planar in practice, it works well in our experiment. The above alternate minimization converges quickly and we usually run a few iterations for each frame.

4.5 Applications

From the tracking and recovery algorithm, we obtain the 3D surface of the paper in each frame, represented as a triangulated mesh of 3D corners. In combination with the calibration result, we can make the flexible surface a versatile interface for visualizing images and data. In this section, we describe two demonstrative applications to illustrate the use of our proposed system.

4.5.1 Flexible image projection

The first one is to display a "flexible" image on the paper, i.e. when the user twists the paper, the image bends simultaneously with the paper deformation, as if it was printed on the paper. This application can be viewed as an example of Shader lamp [48], in which the object to be modified is the projection surface itself. It would be useful as a model previewing tool to view different appearances of a curved surface. This kind of flexible projection can be widely used in entertaining and educational field to produce a more immersive user experience.

The display relies on a pre-warping of the display content before projected to the paper. Given the display content image \mathcal{S} , the warping of the projection image \mathcal{Q} is conducted as follows: for each triangle of the surface, we first project its three vertices to the projection image plane using the projection matrix \mathbf{P} . For example, a triangle composed of $\mathbf{X}_i, \mathbf{X}_j, \mathbf{X}_k$ is projected to $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$. Then for each pixel \mathbf{x} in the projected triangle, we find its correspondence point \mathbf{X} on the paper. We write \mathbf{X} in barycentric coordinates in terms of its three vertices:

$$\mathbf{X} = \xi_1 \mathbf{X}_i + \xi_2 \mathbf{X}_j + \xi_3 \mathbf{X}_k \quad (4.22)$$

Since point \mathbf{X} projects to the pixel \mathbf{x} via the projection matrix \mathbf{P} , we can obtain the barycentric coordinates by solving the following

linear equations:

$$\begin{aligned} s\tilde{\mathbf{x}} &= P\tilde{\mathbf{X}} \\ \xi_1 + \xi_2 + \xi_3 &= 1 \end{aligned} \tag{4.23}$$

The target is then to set the content of $Q(\mathbf{x})$ to what should be displayed on \mathbf{X} . We apply the same triangulation of the surface to S , and for each \mathbf{x} , with the barycentric coordinates ξ_1, ξ_2, ξ_3 calculated we can find the corresponding pixel \mathbf{y} in S . The color of $S(\mathbf{y})$ is then copied to $Q(\mathbf{x})$. By reversing the projection direction, the content of each triangle in S will be projected to a corresponding triangle region on the surface.

A depth image Z with the same size of the projection image is used to handle the possible mutual occlusion of the triangles. Each $Z(\mathbf{x})$ keeps the minimum depth among all the points that project to \mathbf{x} in the projection image. We initialize it with a very large depth. During the warping, for each X , if it is nearer than the depth kept in $Z(\mathbf{x})$, we do the warping and replace $Z(\mathbf{x})$ with the depth of X . Otherwise, it means X is occluded and we simply ignore it.

Through the above warping, the display content can be shown to deform with the paper simultaneously when projecting the pre-warped images.

4.5.2 3D volume visualization

The second application is to use the flexible screen as 3D volume data visualization tool, which would be very useful for viewing cross sections of medical volumetric data such as MRI and CT. Instead of displaying the volumetric data on a fixed screen, the proposed system can be used as a slicing tool to examine the slices of the volume data in their actual positions. With this slicing tool, we can simulate a virtual volume placed at a certain position in front of the projector. When the user moves the paper within the virtual volume, the slice of the volume data is shown on the paper, as if the user is holding the actual slice of data. This would give the user a more intuitive and immersive experience, and also more freedom in interaction compared with traditional keyboard-mouse display system. Moreover, since most real volume data rarely follows a perfect plane, e.g., spine, or kidney, it is desirable that the viewer be able to view curved slices of data. With the help of our system, the observer can see details of the inner surface simply by adjusting the position and shape of the paper held by his hands. This is particularly useful when the doctor is analyzing the health condition of a patient or when a medical teacher is teaching the student about the structure of the human body. We believe the proposed system will have great potential in medicine and education.

The generation of the projection image is the same as the first

application except for the setting of $Q(\mathbf{x})$. The volume data is first loaded as a 3D texture and aligned at a certain position in front of the projector. It is also scaled to match the size of the projection screen. After that, we solve the intersection of \mathbf{X} with the volume data. The intersected voxels of data are then interpolated to give the value that should be display at \mathbf{X} . This is also the value that should be set to $Q(\mathbf{x})$. Through this process, the cross section image is created and then projected onto the surface.

4.6 Experimental Results

We have built a prototype system with the following devices: an off-the-shelf projector with resolution of 1280×1024 , and three Logitech Quickcam Pro 4000 webcams with resolution of 320×240 . A dual core 2.16 GHz PC with 1 GB memory stick is used as the testing platform. Since we are not using any special devices, the cost of our system is low. Experimental results show that the display system achieves satisfactory accuracy and robustness with these ordinary devices.

System calibration We use a thin but hard cardboard with 3×2 checker pattern printed on both sides (see Fig. 4.3) to collect correspondences. The width of each checker is 50 mm. We collect totally 48 correspondences to calibrate the projector camera pair and

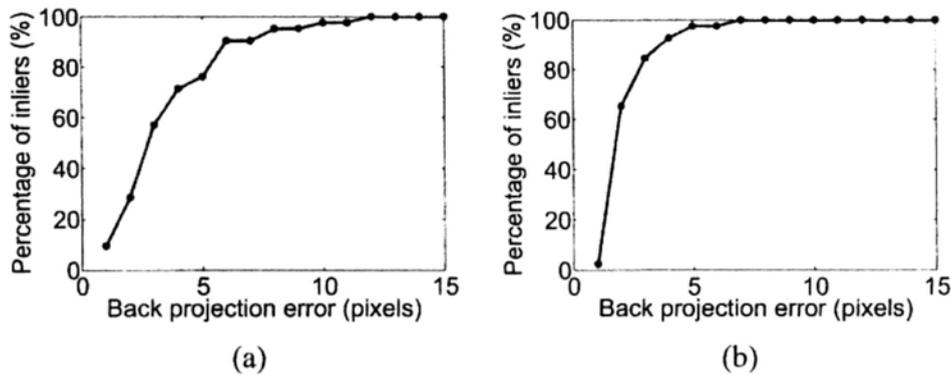


Figure 4.5: The distributions of the back projection error for the two calibrations. (a) Estimation of P . (b) Estimation of M .

72 correspondences to calibrate the tracking camera pair by changing the position and orientation of the cardboard. The whole process takes about 10 minutes. Most of the time is spent in eliminating the false detections of the cross. The calibration time can be further reduced by improving the detection. The accuracy of the estimated projection matrix is measured by the distribution of the back projection error, which is the percentage of the points with back projection error below some pixel level (inliers). The evaluation is conducted on another stand-alone correspondence set. The error distributions of the two calibrations are shown in Fig. 4.5. The back projection error corresponding to 80% inliers for the tracking camera pair is 2.6 pixels and that for the projector camera pair is 5.3 pixels. It is an acceptable accuracy for our display application.

Paper surface recovery The parameters of the recovery algorithm are set as follows in our experiments: the weight λ of the smoothness term is set to 1×10^5 and the back-projection error bound γ is set to 2 pixels. We run the alternate minimization between \mathbf{X} and ω for 3 iterations. To evaluate the performance of the recovery algorithm, we generate a sequence of 200 synthetic surfaces by simulating a paper bending process. Some frames are shown in Fig. 4.6. The lattice is 4×3 and the width of each checker is 50 mm, which is the same as our checker pattern. The 3D corners are then projected to 2D with the intrinsic parameter matrix \mathbf{K} of the reference camera and the calibrated projection matrix \mathbf{M} . Gaussian noises with standard deviation $\sigma = 2$ are added to the 2D projections. We test our algorithm with fixed and flexible triangulation models on the same synthetic data. The accuracy of the recovery is measured in two aspects: the mean distance between the recovered corners and their ground-truth positions, and the back-projection errors. The result is shown in Fig. 4.7. We can see that the proposed method achieves good accuracy in both mean error distance and back-projection errors, and the result with flexible triangulation model is more accurate and stable.

To evaluate the performance of our recovery algorithm in real scenarios, we test it with live webcam captures. Fig. 4.8 shows several frames of tracking and recovering a paper printed with 4×3

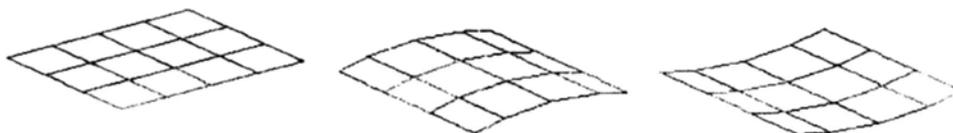


Figure 4.6: Some frames of the synthetic 4×3 surface sequence.

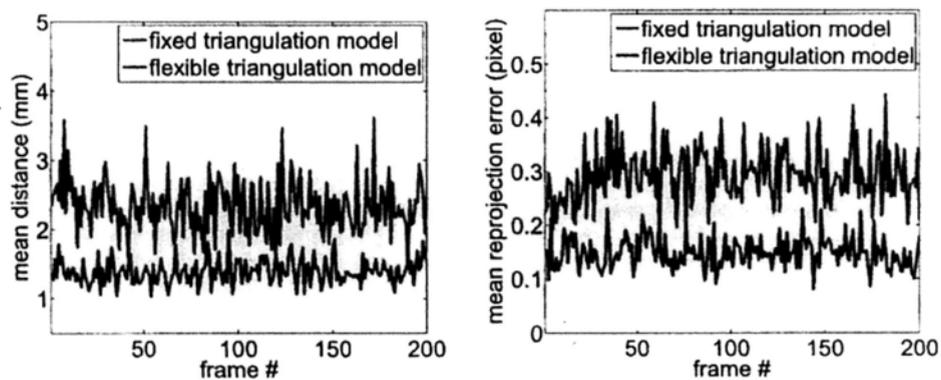


Figure 4.7: The performance of our algorithm on a synthetic sequence. The flexible triangulation achieves better accuracy and stability.

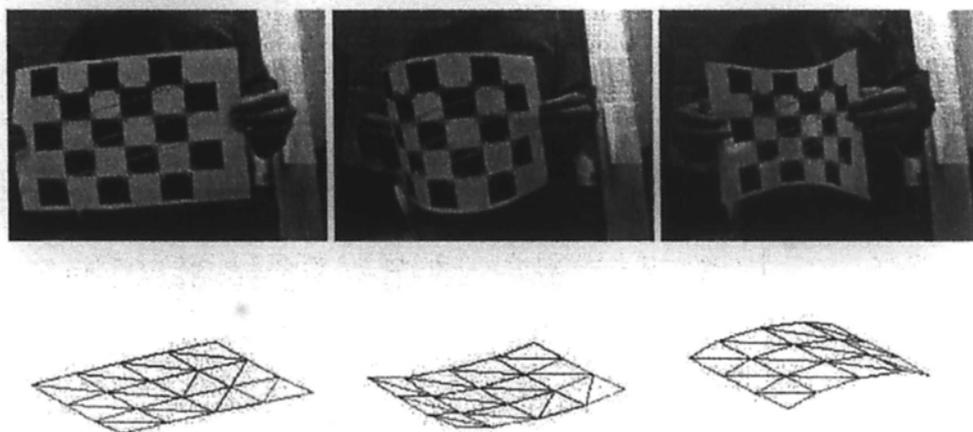


Figure 4.8: Recovering the surface of a paper printed with 4×3 checker pattern in live webcam capture. The first row shows the tracked check patterns. The second row shows the recovered surface in another perspective.

checker pattern. The recovered surfaces are shown in another perspective (from the user's view). The performance of our algorithm on real data is difficult to evaluate quantitatively since the ground-truth 3D corners of the check pattern are difficult to measure. Here, we simply evaluate the recovering accuracy of the curvature of the paper since the ground-truth curvature of the paper can be measured by the height and width of the arch. We compute the curvature of the recovered surface according to the 3D corners, and then compare it with the manually measured data. Five set of deformations are evaluated and the recovering error is plotted in Fig. 4.9. In general, our algorithm can recover the paper with a maximum curvature about 0.5 (its corresponding recovering error is about 0.05), which is enough for most of applications.

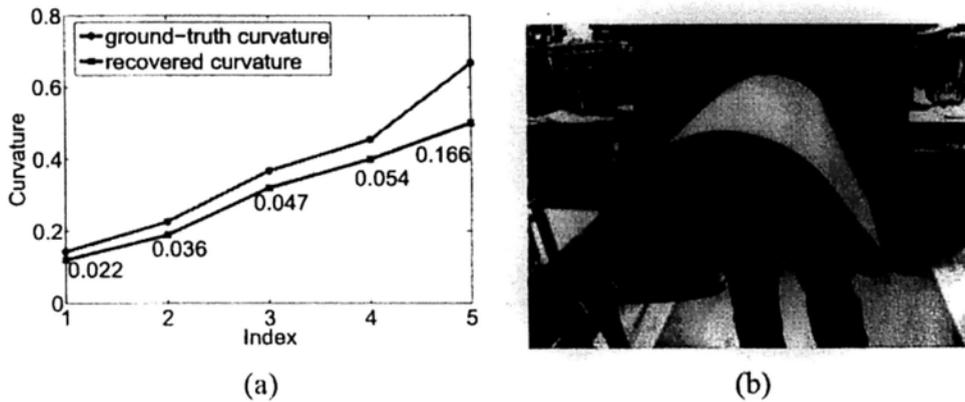


Figure 4.9: (a) The recovering accuracy of the curvature. The errors between recovered curvature and ground-truth are also plotted. (b) The approximate maximum acceptable deformation of the paper.

The working area of the camera pair is also investigated. In general, the working area is the intersection of the field of view of the two cameras, but limited to a range in depth. If the paper is too close to the camera, it cannot be observed by both cameras. On the other hand, if the paper is too far away from the camera, the checker pattern on the back of the paper may appear too small in the camera to be tracked. An illustration diagram of the working area is shown in Fig. 4.10. To find out its range, we move the paper to everywhere it can be tracked by the camera pair. The position of the paper is estimated for each frame based on the recovered 3D paper surface and the range of the working area is then evaluated as the maximum allowed moving range of the paper. It is approximately 0.9 m in height, and 0.9 m in width at the top of the working area for our prototype configuration. The working area is not fixed, i.e., it can

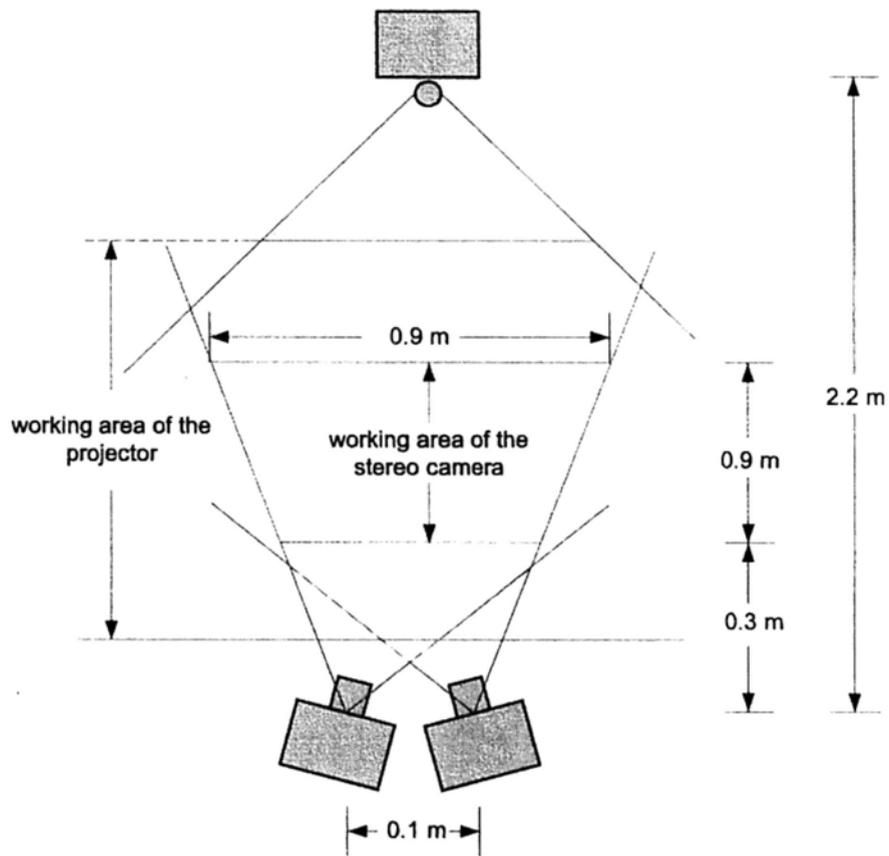


Figure 4.10: The working area of the system.

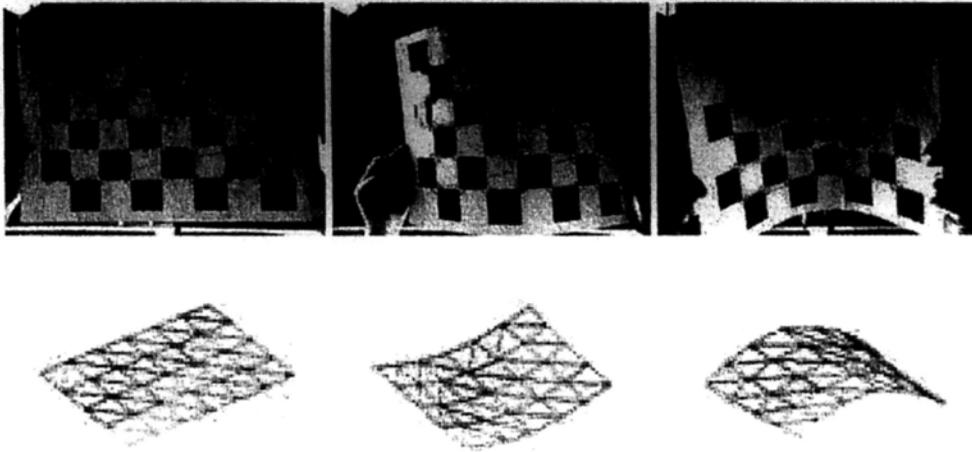


Figure 4.11: Tracking and recovering the surface of an A3 paper printed with a 6×4 checker pattern in a live webcam capture. The first row shows the tracked checker patterns. The second row shows the recovered surface in another perspective.

vary in depth with the size of the paper. Specifically, if the paper is bigger, the working area will go further from the camera, and vice versa. It is thus no problem to track a bigger paper, e.g. an A3 paper. Fig. 4.11 shows several results of tracking and recovering the surface of an A3 paper printed with a 6×4 checker pattern on the back. The grid size of the checker pattern should be compatible with the paper size. It cannot be too big or too small. Otherwise, the approximation of the flexible surface may be unacceptable, or it may not be tracked accurately and robustly. Usually, a 4×3 grid for an A4 paper and a 6×4 grid for an A3 paper is a good choice. The recovery errors with these two kinds of papers are comparable in our experiments.

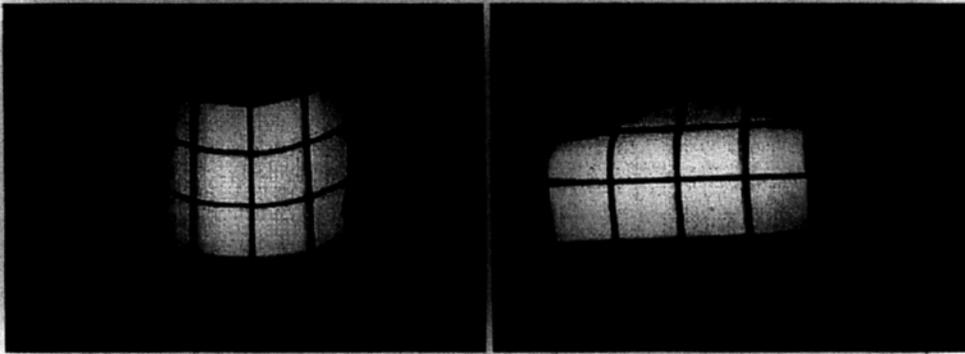


Figure 4.12: The green grid is projected while the black grid is printed. The coincidence of the green grid with the black grid indicates the projection accuracy of the system.

Display results To evaluate the projection accuracy, we project a green grid onto the paper on which a black grid is printed. In this case, the black grid is the ground-truth. Therefore, by observing and measuring the offset between the green grid and the black grid, we can access the projection accuracy. An evaluating video is recorded and the projection errors between the green and black grid are manually measured. The average projection error of the corner is about 2.3 pixels. Two evaluating frames are shown in Fig. 4.12.

Some results of the flexible image display application are shown in Fig. 4.13. The cooperation of the surface recovery and the image pre-warping routine can successfully project correct content on the surface under different kinds of deformations. Fig. 4.14 illustrates some results of virtually slicing a MRI brain. From these images, it can be easily seen that a curved slice of the brain can be exhibited to the user owing to our system. In both experiments, we find that

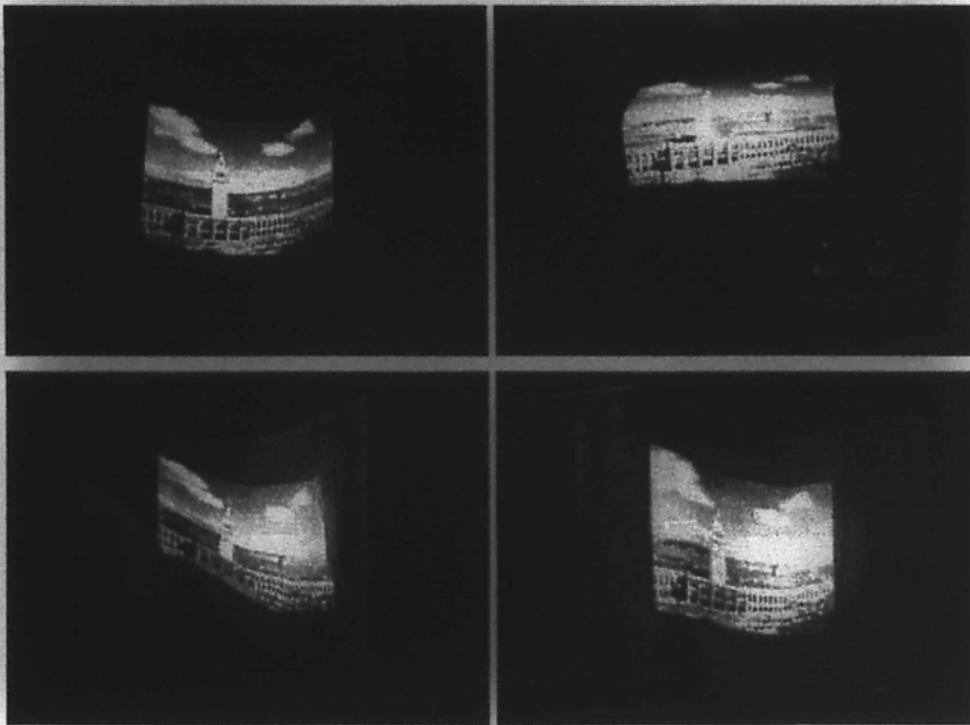


Figure 4.13: Some results of flexible image projection.

the system can warp the projection image correctly and create the flexible display with satisfactory accuracy and robustness. It runs smoothly and no obvious latency and flickering effect is observed. Continuous display result can be watched online (<http://www.youtube.com/watch?v=Kfdlo1Z75XQ>).

Our system can achieve real-time performance on the above platform. The running time of each frame is mainly occupied by the checker pattern tracking (about 5 ms), the surface recovery (about 10 ms) and the projection image warping (about 15 ms). The surface recovery is actually fast while the projection image warping is

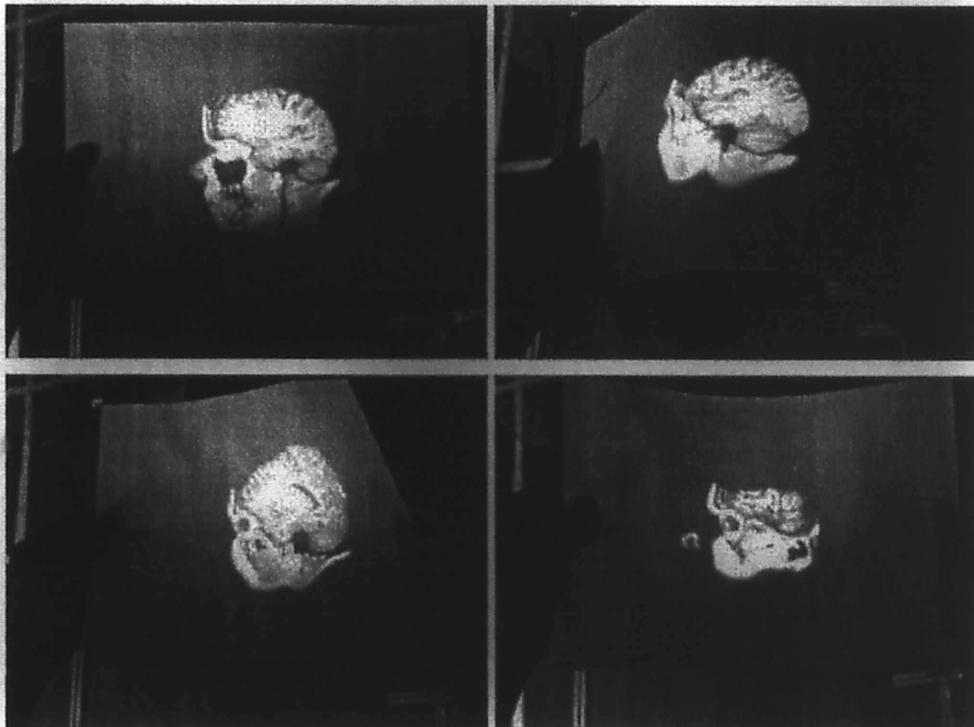


Figure 4.14: Some results of slicing a MRI brain.

slow. When running with live webcam capture, our system can still achieve real time processing smoothly (a frame rate about 18 fps). This speed is acceptable even when carrying out some fast deformations (see supplementary video). However, limited by the frame rate of the webcam and the KLT algorithm (which expects tiny variance between consecutive frames), our system cannot accurately handle drastic deformations and will exhibit perceivable latency. But if cameras with higher frame rates (like embedded cameras) can be used, this problem will be greatly relieved, and the recovery ability of the system can be further enhanced.

4.7 Discussions

Although the performance of the system is satisfactory, there are still several aspects that can be further improved. First, there is some limitation on the working area of the system, which is primarily confined to the working area of the camera pair. Although it is sufficient for many visualization applications like the two demonstrated in this chapter, it may be limited for some interactive applications that require large movement of the projection surface. Possible ways to enlarge the working area include using cameras with bigger field of view and focus area, or using an array of cameras to ensure that the pattern can always be seen by two of them. Second, there is limitation on the depth field of the projector. The projector used in our

system has quite limited depth of field, which makes the projection to be in focus only within a particular range of depth. A solution to this problem is to use multiple projectors. Third, there may be applications that require fine representation of the surface. The proposed system may fail as we are using a coarse checker pattern to approximate the surface for the benefit of easy tracking. Increasing the size of the checker pattern or using texture-abundant patterns could achieve a better approximation, but it would increase difficulty in feature tracking and matching, and also increase the running time greatly. A fast and robust tracking algorithm may be needed to make the system real time for large scale applications.

The 3D volume visualization is not new, but most existing systems like [10] can only deal with a planar slicing surface. The novelty of our application is that we extend it to allow flexible slicing using low-cost off-the-shelf devices. Although in most cases the planar slicing is enough for clinicians, it is good to provide them with curvilinear slicing in case that they may want to see the curved slice of the data, for example, a section of the spine or a layer of the kidney. The advantage of the proposed system lies in that it can not only deal with flexible slicing, but can also naturally handle planar slicing without any additional effort. By replacing the deformable paper with a rigid cardboard with similar patterns printed on the back, the proposed system becomes a planar slicing tool similar to

[10] without any modification to the program. This gives our system greater freedom as a 3D volume visualization tool. With the proposed system, the clinician can visualize the medical data under "planar" mode in most cases, and they can simply change to "flexible" mode in a breeze when necessary. The accuracy presented may not be enough for quantitative medical image analysis, but it is enough for visual analysis or illustration purpose. As described in the manuscript, its major application lies in education or exhibition, for which the accuracy is enough. The accuracy can also be further improved by optimizing the implementation.

End of chapter.

Chapter 5

Mobile Projection Keystone

Correction

Keystone correction is an essential operation for projector based applications, especially in mobile scenarios. In this chapter, we propose a hand-held movable projection method that can freely project keystone-free content on a general flat surface without adding any markings or boundary on it. Such a projection system can give the user greater freedom of display control such as viewing angle, distance etc, without suffering from keystone distortion. Compared with traditional static projection systems that keep the projector and screen in a fixed position, our projection scheme can give the user greater freedom of display control (such as viewing angle, distance etc), and produce undistorted display at the same time. We attach a camera to the projector to form a camera-projector stereo pair. A green frame with the same resolution as the projector screen

is projected onto the screen. Particle filter is employed to track the green frame and the correction of the display content is then achieved by rectifying the projection region of interest into a rectangular area. We built a prototype system to validate the effectiveness of the method. Experimental results show that our method can continuously project distortion free content in real time with good performance. This work is published in [32].

5.1 Introduction

As computer technique advances, the physical size of a low-cost projector is much reduced, which can be incorporated into many mobile devices such as phones or cameras. These mobile projection devices provide us with enhanced viewing experience, through which our eyesight will no longer be limited to a small screen, nor will it be confined within a narrow angle. For example, using a digital camera with a projection module on it, we can shoot a picture and immediately project the image to share with a group of friends, instead of asking all people to stare at the small screen on the camera.

The promising future of mobile projection is unquestionable. However, a big obstacle of it being widely used in a mobile environment is keystone distortion: when we project an image onto a screen at oblique positions, the projection region will become a trapezoid instead of a rectangle. This kind of distortion gives the user an un-

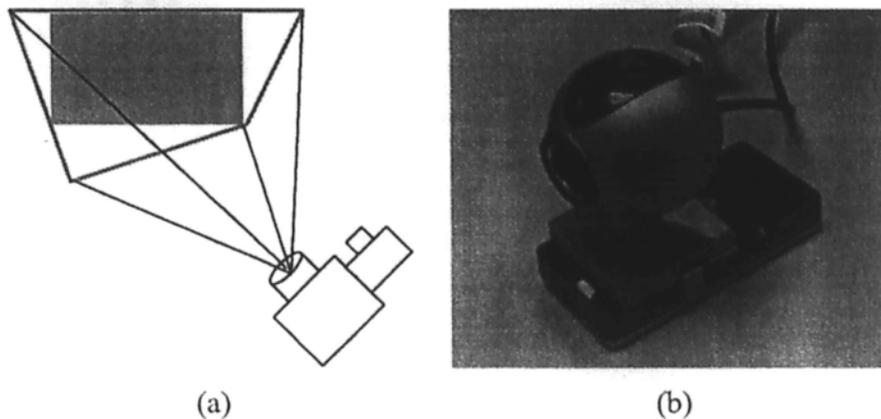


Figure 5.1: (a) Keystone correction for mobile projector. (b) The mobile projector attached with a webcam in our prototype system.

pleasant experience and the correcting of it becomes a stringent need, especially in a mobile scenario where the mobile projector might be moving continuously. In this scenario, a good keystone correction method should be equipped with the following features: (1) screen independence: no specially designed or position-fixed screen should be required, i.e., the user can project on any normal flat surface; (2) continuous processing in real time: since the pose of the projector is not fixed, continuous correction instead of one-time correction is expected to be performed in real-time for the best user experience.

Motivated by this, we propose a method which can continuously correct the distortion and display the content of interest in a rectangular area on a markless screen. The only additional device used is a webcam attached with the projector (see Fig. 5.1), which is quite natural since we are observing more and more mobile devices with

both embedded projector and camera recently. The projector-camera pair which has a stereo relationship between them is calibrated beforehand. Without printing any markings or boundary on the screen, a green frame is projected onto it instead. The camera tracks the frame and finds the pose between the projector and the screen. The correction of the display content is then achieved by recovering the 3D projection region and rectifying the projection region of interest into an inscribed rectangle in the 3D projection region. The original display image is pre-warped so that it will be projected into this rectangle. While the projector is moving, our method keeps tracking the green frame using particle filter and handling the correction. As a result, the user can enjoy a keystone free viewing experience even when he or she is moving the projector.

Our method is intended for markless mobile projection, which distinguishes itself from existing approaches concentrating on one-time correction for static projectors. For example, Sukthankar *et al.* [52] proposed to correct keystone with a fixed camera-projector pair by using homographies among the projector, camera and screen. However, in their proposed method, a fixed screen is needed, and the correction algorithm relies on detecting the screen boundary. This method is not suitable for mobile projection since blank surfaces (walls, floors etc) without boundaries or markings are usually used as the projection surface of mobile projectors. Raskar *et al.* [44] pro-

posed a correction method without requiring a boundary or markings on the screen. However, their algorithm requires a full calibration of the projector and camera. The complexity and high computation cost of the algorithm prevent it from being widely used in mobile projection applications. Li *et al.* [30] proposed an efficient keystone correction method in which not only keystone correction but also auto zooming and screen fitting are achieved. However, this method still requires a bounded screen. In Table 5.1, we compare these methods with the proposed one in terms of what type of projectors is used, computation time, whether requires marking, and commercial feasibility. To sum up, mobility, marklessness, and real-time correction are the key features of our method; previous methods are not suitable for our mobile projection purpose.

Table 5.1: Overall comparison between the proposed method and others

	Projector	Time ¹	Markless	Commercial use
Sukthankar's [52]	static	about 200ms	no	overhead projection
Raskar's [44]	static	about 1s	yes	overhead projection
Li's [30]	static	about 120ms	no	overhead projection
Our method	movable	about 60ms	yes	mobile devices

Our main contribution is that our system is the first to deal with the movable markless projector keystone problem with a simple but effective solution. The novelty of the method mainly lies in a parti-

¹The time listed for other three methods is the computation time needed for one full correction process, while it is the per-frame computation time for our method. They are estimated with our platform.

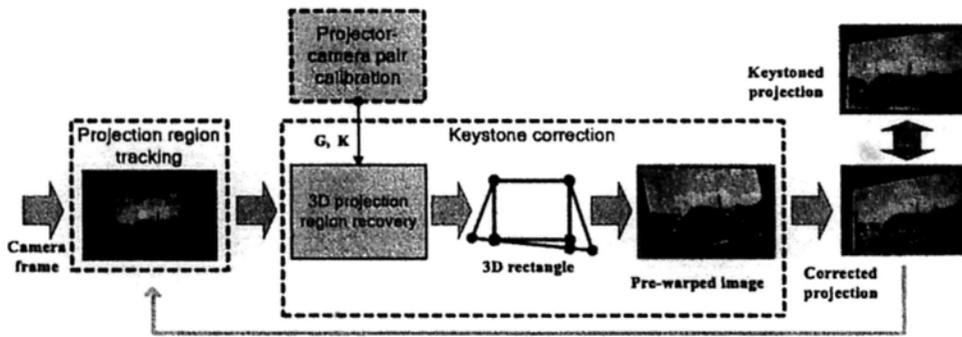


Figure 5.2: The keystone correction flowchart. The dashed boxes highlight the three modules of the method.

cle filter based tracking scheme artfully introduced without full calibration of the projector, and a coplanarity enforcement process to improve the accuracy of the recovered projection region. The remainder of the chapter is organized as follows. In Section 5.2, we give an overview of the proposed method. The technical details are described in Section 5.3, 5.4, and 5.5. Experimental results are given in Section 5.6. We discuss this chapter in Section 5.7.

5.2 System Overview

Our method is an integration of three modules, the calibration module, the tracking module, and the correction module. The calibration module is run offline, which finds the relationship between the projector and camera. The tracking module takes the camera frame captured as input, and tracks the projection region. The correction module rectifies the keystone distortion based on the calibration re-

sult and the tracked projection region. The work-flow of the method is illustrated in Fig. 5.2. In following sections, we describe each module in detail.

5.3 Projector-Camera Pair Calibration

We use a calibrated camera with known intrinsic parameters. In order to correct the keystone distortion, we need to calibrate the geometric relationship between the projector and camera. Since the projector is moving in our system, a fixed relationship that is independent from the motion of the projector is needed. Our solution is simply to estimate the projection matrix from the 3D camera coordinate to the projector image plane.

Ideally, the projective model of a projector is similar to the camera model except for the projection direction. The projection from a 3D world point to the 2D projector image pixel is also via a 3×4 perspective projection matrix. So for each 3D point $\mathbf{X}^c(x, y, z)$ in the camera coordinate system, it relates its corresponding projector image pixel $\mathbf{x}^p(u, v)$ by a projection matrix \mathbf{G} :

$$\lambda \tilde{\mathbf{x}}^p = \mathbf{G} \tilde{\mathbf{X}}^c \quad (5.1)$$

where $\tilde{\mathbf{x}}^p, \tilde{\mathbf{X}}^c$ are homogeneous coordinates, λ is a scale factor, \mathbf{G} describes the intrinsic parameters of the projector and the relative

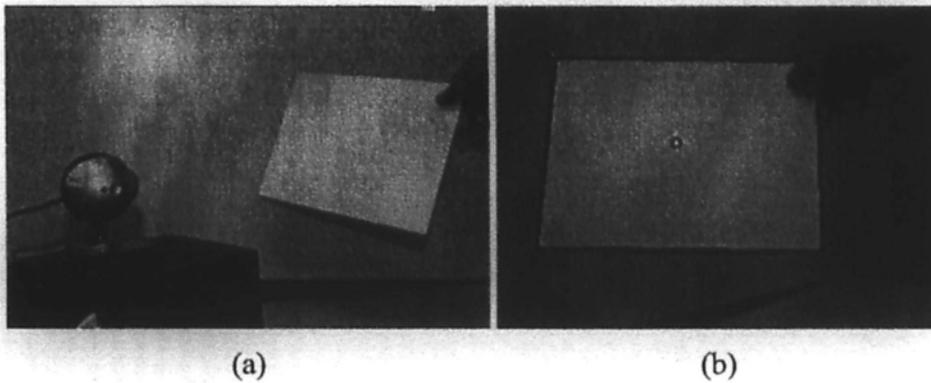


Figure 5.3: (a) A cardboard is held to collect the cross projected onto it. (b) The cardboard and the cross in the camera are detected to calculate the 3D-2D correspondence.

pose between the projector and camera. A full calibration like that proposed in [44] involves estimating all the intrinsic and pose parameters, which is rather complicated and a stable result is not easy to obtain. However, in our method, owing to our novel keystone correction algorithm, we do not need to estimate all these parameters explicitly, but simply estimate the projection matrix G .

A simple method proposed in [28] is employed to estimate the projection matrix. The main idea is to collect a number of correspondence points between the 3D points in the camera coordinate system and their 2D projections in the projector image. As shown in Fig. 5.3 (a), the user holds an ordinary cardboard with known size in front of the projector. A cross with a known position in the projector is projected onto the cardboard at the same time. The calibration module automatically detects the cardboard and the cross in the camera. If both are successfully detected, the program reports

a correspondence. A successful detection is shown in Fig. 5.3 (b). Based on the detection, the 3D positions of the cardboard and the cross in the camera coordinate system can be easily calculated via a Perspective-4-Points (P4P) algorithm proposed in [61]. In this way, a 3D-2D correspondence is established. The projection matrix is then estimated based on a number of such 3D-2D correspondences using Singular Value Decomposition (SVD).

5.4 Projection Region Detection and Tracking

To assist the detection and tracking of the projection region, we add a green frame along the border of the projector screen in the projection image. An illustration of the final projection image with a green border is shown in Fig. 5.5. The whole detection process can be divided into two stages. In the initial stage, we detect a quadrangle fulfilling several criteria as the initial position of the projection region. After that, we track its position in the subsequent frames using particle filter. The tracking process is introduced so as to obtain a smooth and coherent correction.

5.4.1 Detection

The detection is performed on the edge map obtained by the Canny edge detector, similar to that in [28]. Owing to the fixed relationship between the projector and camera, the projection region appears

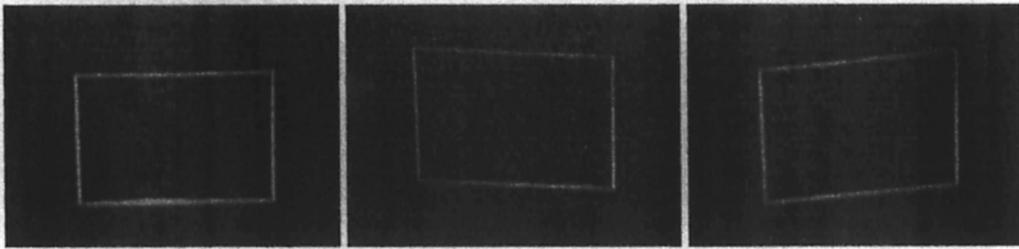


Figure 5.4: Some typical shapes of the green frame.

nearly rectangular in the camera. Moreover, its shape only varies within a small range regardless of the pose of the projector. Our experimental investigation also confirms this. Some typical shapes are shown in Fig. 5.4. The detection algorithm is thus simple. We use a Hough Transform line detector to extract a set of line segments, and then test which four segments form a desired quadrangle with the following criteria: (1) each side of the formed quadrangle should be longer than a threshold; (2) opposite sides should have similar lengths; (3) each angle should be within the range from 30° to 150° ; (4) the overlapping ratio of the line segments to the four sides of the formed quadrangle should be bigger than a threshold; (5) the quadrangle is approximately located around the center of the camera image. If a quadrangle satisfying all the criteria is detected, we regard it as the initial projection region and proceed to the tracking stage.

5.4.2 Particle filter tracking

From the detection result, we have obtained the 2D positions of the four corners of the projection region. Direct tracking of the four corners would incur redundancy in the tracking state since the projector camera pair is actually dominated by a homography:

$$\lambda \tilde{\mathbf{x}}^p = \mathbf{H} \tilde{\mathbf{x}}^c \quad (5.2)$$

where \mathbf{H} is the homography matrix from the camera to the projector, \mathbf{x}^p is the corner of the projector screen, \mathbf{x}^c is the corner of the projector region in camera image, λ is a scale factor. Moreover, according to [20], \mathbf{H} is further expressed as:

$$\mathbf{H} = \mathbf{J} \left[\mathbf{R} - \frac{\mathbf{t} \mathbf{n}^T}{d} \right] \mathbf{K}^{-1} \quad (5.3)$$

where \mathbf{J} is the intrinsic parameter matrix of the projector, \mathbf{R} and \mathbf{t} are the rotation and translation of the camera relative to the projector, \mathbf{n} is the normal of the screen relative to the camera, d is the distance of the screen from the camera, \mathbf{K} is the known intrinsic parameter matrix of the camera. We can see that the projector-camera homography is actually ruled by \mathbf{n} and d since \mathbf{J} , \mathbf{R} , \mathbf{t} , \mathbf{K} are all fixed. So we can treat \mathbf{n} and d instead of four corners as the tracking state vector \mathbf{s} :

$$\mathbf{s} = [\mathbf{n}^T, d] \quad (5.4)$$

Particle filter [1] is employed to estimate the posterior density of \mathbf{s} , which is represented as a set of particles. Each particle has a weight to indicate how confident it is to represent the real projection region. The two main components of a particle filter are the state dynamic model and the observation model. The dynamic model defines how the particles propagate from previous frame to current frame while the observation model determines how weights are assigned to particles providing the observation at that frame. We detailed the two models and the initialization method in our algorithm as follows:

dynamic model Since the projector is moving in free motion, a simple random walk model based on an uniform density about the the previous state is used, i.e, the state at k^{th} frame obeys an uniform distribution in the neighborhood of the state at $k - 1^{th}$ frame:

$$p(\mathbf{s}_k | \mathbf{s}_{k-1}) = Uniform(\mathbf{s}_{k-1} - \mathbf{e}, \mathbf{s}_{k-1} + \mathbf{e}) \quad (5.5)$$

where \mathbf{e} represents the uncertainty about the movement of the projector.

observation model To evaluate the likelihood of each particle, we first re-project the projector screen to the camera image plane according to $[\mathbf{n}^T, d]$ represented by the particle using Eq. (5.2), (5.3). Though we do not calibrate the projector parameters explicitly, i.e,

we did not find \mathbf{J} , \mathbf{R} , \mathbf{t} but \mathbf{G} instead, it is still feasible to do the re-projection. We reformulate Eq. (5.3) to Eq. (5.6):

$$\mathbf{H} = \left[\mathbf{JR} - \frac{\mathbf{Jt}\mathbf{n}^T}{d} \right] \mathbf{K}^{-1} \quad (5.6)$$

\mathbf{JR} and \mathbf{Jt} can therefore be obtained from \mathbf{G} :

$$[\mathbf{JR}, \mathbf{Jt}] \propto [\mathbf{G}_{3 \times 3}, \mathbf{G}_{3 \times 1}] \quad (5.7)$$

where $\mathbf{G}_{3 \times 3}$ is the first three columns of \mathbf{G} , $\mathbf{G}_{3 \times 1}$ is the last column of \mathbf{G} . In this way, we can obtain the homography matrix. The projector screen is then back-projected to the camera image by the inverse homography matrix:

$$\tilde{\mathbf{x}}^c = \lambda \mathbf{H}^{-1} \tilde{\mathbf{x}}^p \quad (5.8)$$

After the re-projection, we evaluate the particle's likelihood by comparing the re-projected quadrangle with the edge map. Specifically, we check how many edge points are on the four sides of the re-projected quadrangle. The checking is performed along each side for every 5 pixels. If there is an edge point whose perpendicular distance to the side is within 5 pixels, we consider that the side has an on-edge point. The likelihood of that side is then assigned as the proportion of on-edge points among total points on that side, and the likelihood of that quadrangle is the sum of the likelihoods of all four

sides. After all particles are evaluated, we choose the particle with the maximum likelihood as the final result of the current frame.

Initialization

First, the detected quadrangle in the detection stage is used to initialize the particle filter. Its 3D position in the camera coordinate system is recovered using the method in Section 5.5.1. The normal and distance to the camera are used as the initial state of the particle filter.

5.5 Automatic Keystone Correction

The correction algorithm contains three steps. As shown in Fig. 5.2, we first recover the 3D position of the projection region based on the 2D tracking result. Second, we look for an inscribed rectangle inside the 3D projection region. Finally, the original projection image is pre-warped so that it will be projected into the inscribed rectangle on the screen. This process repeats for each camera frame.

5.5.1 Recovering 3D projection region

Having obtained the 2D camera position of the projection region, we proceed to recover its 3D position in the camera coordinate system. Based on the calibration result of the projector-camera pair, we can recover the 3D position of each corner of the projection region from

its corresponding pixels in the camera and projector image. The four corners in the camera are already tracked and their positions in the projector can be simply obtained according to the resolution of the projector screen. Moreover, the correspondence between them is fixed no matter how we move the projector-camera pair.

Supposing that the 3D coordinates of the four corners of the projection region to be solved are $\mathbf{X}_i^c, i = 1 \dots 4$ respectively, they and their 2D positions in the camera $\mathbf{x}_i^c(u, v)$ and projector images $\mathbf{x}_i^p(\alpha, \beta)$ should satisfy Eq. (5.1) and the projection equations of the camera in Eq. (5.9):

$$\lambda \bar{\mathbf{x}}^c = \mathbf{K} \mathbf{X}^c \quad (5.9)$$

where λ is a scale factor, and \mathbf{K} is the known intrinsic parameter matrix of the camera. Each projection equation can be re-arranged into two linear equations. Hence, there are totally 4 linear equations with 3 unknowns (3D coordinates of the corner). A least square solution can be obtained by SVD.

However, the SVD solution cannot guarantee the coplanarity of the four corners since they are solved separately. Geometrically, four coplanar points should satisfy the following condition:

$$\overrightarrow{\mathbf{X}_1^c \mathbf{X}_4^c} \cdot (\overrightarrow{\mathbf{X}_1^c \mathbf{X}_2^c} \otimes \overrightarrow{\mathbf{X}_1^c \mathbf{X}_3^c}) = 0 \quad (5.10)$$

Directly incorporating the condition into Eq. (5.1) and Eq. (5.9) will

result in a nonlinear equation that is difficult to solve. Instead, we carry out a post refinement to the SVD solution, which minimizes the sum of squared back-projection errors in the camera and projector, plus the coplanarity constraint:

$$\begin{aligned}
 & \sum_{i=1}^4 \left\| u_i - \frac{\mathbf{k}_1^T \mathbf{X}_i^c}{\mathbf{k}_3^T \mathbf{X}_i^c} \right\|^2 + \left\| v_i - \frac{\mathbf{k}_2^T \mathbf{X}_i^c}{\mathbf{k}_3^T \mathbf{X}_i^c} \right\|^2 \\
 & + \sum_{i=1}^4 \left\| \alpha_i - \frac{\mathbf{g}_1^T \tilde{\mathbf{X}}_i^c}{\mathbf{g}_3^T \tilde{\mathbf{X}}_i^c} \right\|^2 + \left\| \beta_i - \frac{\mathbf{g}_2^T \tilde{\mathbf{X}}_i^c}{\mathbf{g}_3^T \tilde{\mathbf{X}}_i^c} \right\|^2 \\
 & + \omega \left\| \overrightarrow{\mathbf{X}_1^c \mathbf{X}_4^c} \cdot (\overrightarrow{\mathbf{X}_1^c \mathbf{X}_2^c} \otimes \overrightarrow{\mathbf{X}_1^c \mathbf{X}_3^c}) \right\|^2
 \end{aligned} \tag{5.11}$$

where $\mathbf{k}_1^T, \mathbf{k}_2^T, \mathbf{k}_3^T$ are the row vectors of \mathbf{K} , $\mathbf{g}_1^T, \mathbf{g}_2^T, \mathbf{g}_3^T$ are the row vectors of \mathbf{G} , ω is a weight. Taking the SVD solution as initialization, we use the Levenberg-Marquardt algorithm [29] to minimize the above function. The optimization stops if a pre-defined accuracy of coplanarity is reached. Since the SVD solution is already close to coplanarity, the above optimization stops within a few iterations.

5.5.2 Looking for inscribed rectangle

Next, according to the obtained 3D positions of the quadrangle's corners, we look for an inscribed rectangle inside the quadrangle whose top side resides in that of the quadrangle. Then the inscribed rectangle is exactly where we expect the projection image appears on the screen finally. Unlike [44] which uses tilt sensors to align

the rectangle horizontally, the projection region of interest in our approach can be adjusted to the most suitable viewing direction by the user, making it the best choice in a mobile scenario.

5.5.3 Pre-warping projection image

By substituting the four corners of the rectangle computed in the previous step into Eq. (5.1), we can obtain their corresponding points in the projection image. The region enclosed by the four points then becomes the effective projection region, and correcting the keystone effect is achieved by warping the original display image into this region. To perform pre-warping, we use a similar homography mapping as in [44] to map the original display image into this effective region. The homography can be calculated from the correspondences between four corners of the effective region and the original display image. Fig. 5.5 shows an example of the original projection image and the pre-warped projection image.

5.6 Experimental Results

A prototype system is built according to our proposed method. The testing platform is a PC installed with a 2.16GHz dual core processor and 1GB memory stick. The projector-camera pair is comprised of an Optoma mobile projector with resolution of 1280×1024 and a Logitech Quickcam Pro 4000 webcam with resolution of 320×240 .



Figure 5.5: (a) The original projection image. (b) The pre-warped projection image with a green frame added along the border.

Projector-camera calibration A cardboard with size of 200×150 mm is used to collect correspondences. By changing the position and orientation of the cardboard, totally 42 correspondences are collected to estimate the projection matrix. The collecting process takes about 5 minutes. Most of the time is spent in eliminating the false detection of the cross. The time can be further reduced by improving the detection. In order to compensate for the detection error of the cross and obtain a stable solution, we use a RANSAC estimation scheme in our algorithm. For each run of RANSAC, 6 correspondences are randomly selected to estimate the projection matrix. The estimate with most inliers is then accepted as the final result. We run RANSAC estimation for 10000 iterations and it takes about half a minute. The accuracy of the estimated projection matrix is measured by the distribution of the back-projection error, i.e., the per-

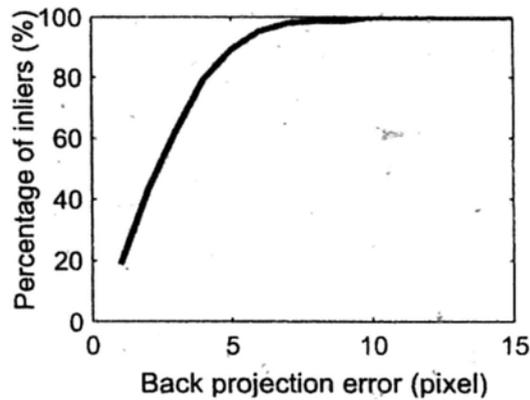


Figure 5.6: The distribution of back-projection errors of the estimated projection matrix.

centage of the points with back-projection error below some pixel level (inliers). The evaluation is conducted on another stand-alone correspondence set. The error distribution is shown in Fig. 5.6. The back-projection error corresponding to 80% inliers is 4.2 pixels. It is an acceptable accuracy for our general projection application.

Projection region detection and tracking We first test the robustness of the detection algorithm against different tilt angles of the projector. By moving the projector, a video sequence of 280 frames containing different tilt angles of the projector is recorded to evaluate the detection rate. We run the detection algorithm on this video frame by frame. The detection rate is then calculated as the proportion of the correctly detected frames. It is 0.98 in our experiment.

To evaluate the performance of the particle filter tracking of the projection region, we synthesize a random motion trajectory of the

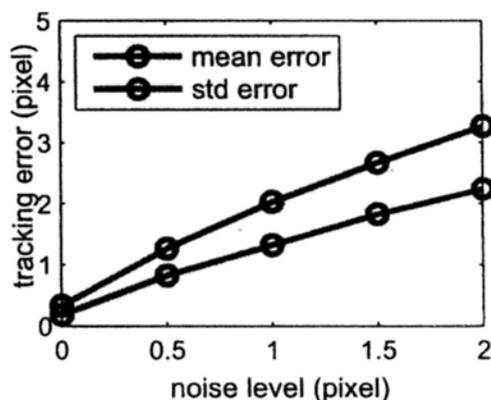


Figure 5.7: The mean and std image error of the tracked projection region w.r.t different noise levels on simulation data.

projector relative to a virtual screen. The projector screen is projected onto the screen and then back-projected to the camera. Zero mean gaussian noise with different standard deviations is added to the projection in camera. A video sequence of 320 frames containing different poses is created. We run our algorithm on the synthetic video and evaluate the averaged error of the four corners between the tracked ones and the synthetic ground-truth. The corresponding result w.r.t different noise levels is shown in Fig. 5.7.

The tracking performance on real data is also tested. A video sequence of 300 frames containing free movements of the projector is recorded to evaluate the tracking accuracy. We manually label the position of the quadrangle every ten frames, and evaluate the error between the tracking results and the manually labeled positions. The mean and std error are 3.4 and 3.6 pixels respectively. The trajectory and orientation of the projector for every ten frames w.r.t. a coordi-

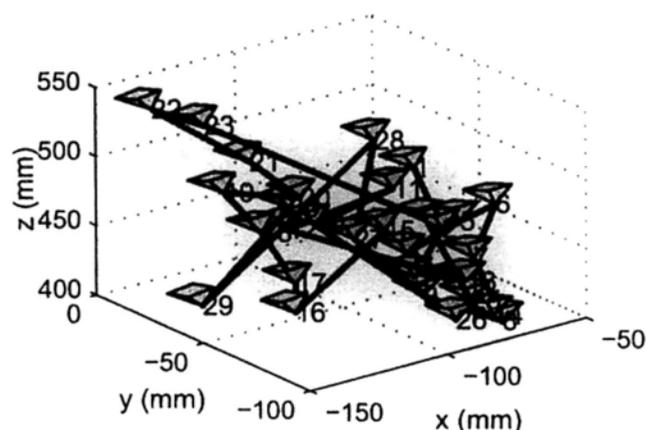


Figure 5.8: The recovered trajectory of the projector on real data.

nate system on the projection screen is plotted in Fig.5.8. From these experiments, we can see that the algorithm can track the projection region with good accuracy and robustness in both synthetic and real scenarios.

Keystone correction We project a picture to test the keystone correction performance. The user casually poses the projector-camera pair towards an ordinary flat board. To show the advantage of the keystone correction algorithm, we demonstrate a comparison between the projection result before and after keystone correction in Fig. 5.9. It can be clearly seen that the keystone distortion is removed, and no obvious perceivable quality degradation is observed. To measure the correction error, we manually measure the shape of the full projection region with the green frame and the corrected

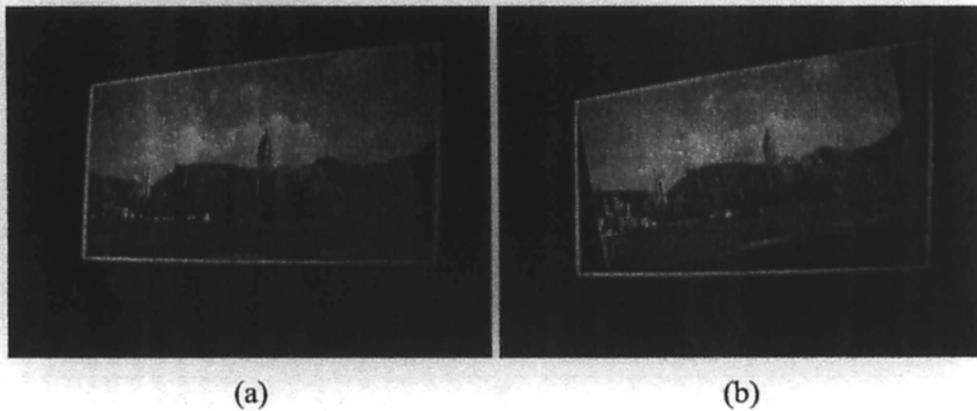


Figure 5.9: (a) The projection result without keystone correction. (b) The projection result after keystone correction.

rectangular projection region using a ruler and then solve the inscribed rectangle of the green frame. The average difference between the four corners of the manually measured rectangular projection region and those of the inscribed rectangle is considered as the correction error. It is 1.8 mm for Fig. 5.9, while the projection region is around the size of an A4 paper. More projection results are shown in Fig. 5.10, which all have a correction error below 4 mm. The projection content of interest resides in a rectangular area after correction. If it were not for the correction, the projection content would distort in the trapezoid area enclosed by the green frame. When the user moves the projector around and freely changes its pose, our system can still effectively correct the keystone distortion. Continuous results can be watched online at <http://www.cse.cuhk.edu.hk/~khwong/demo>.

We carry out simulation experiments to test the keystone correc-

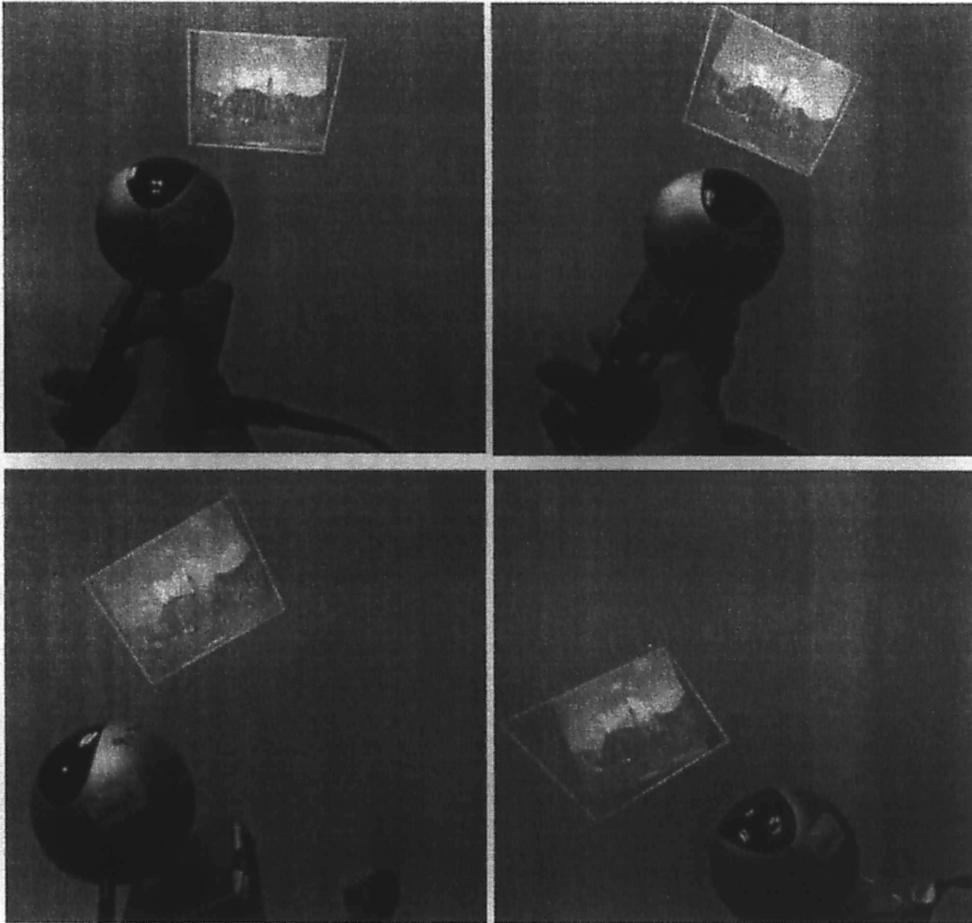


Figure 5.10: Some correction results in real projection.

tion performance with different tilt angles of the projector relative to the screen. The projector is adjusted to different pitch (horizontal tilt) and roll (vertical tilt) angles. Both angles are tested every 10 degrees within the range from -45 to 45 degree, so totally 10×10 poses are evaluated. At each pose, the projector screen is projected onto the virtual screen and then back-projected to the camera. Zero mean Gaussian noise with $std = 2.0$ is added to the projection in the camera. We run the keystone correction algorithm on the back-projected quadrangle. The effective projection region obtained is then projected to the virtual screen. The accuracy of the corrected result is then evaluated by measuring the difference between the corrected rectangular region on the virtual screen and the ground-truth rectangle that should be projected onto. The averaged error of the four corners is evaluated. The corresponding result is plotted in Fig. 5.11. In general, the correction error becomes larger when the projector becomes more oblique. According to our real projection experiment, the maximum allowable correction error for human eye is about 4 mm (around A4 size projection region), which is found by investigating the correction errors of a number of eye acceptable projection results. Our algorithm can obtain good correction with a maximum tilt angle of 30 degrees in real projection, which is enough for most real applications.

We compare our keystone correction method with the method

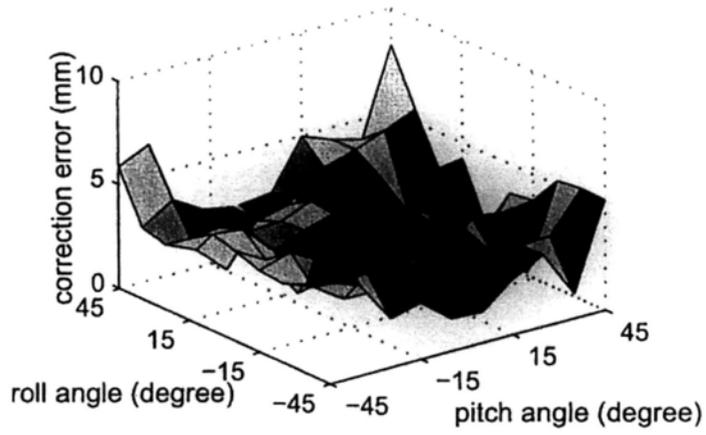


Figure 5.11: The error of keystone correction against different poses of the projector.

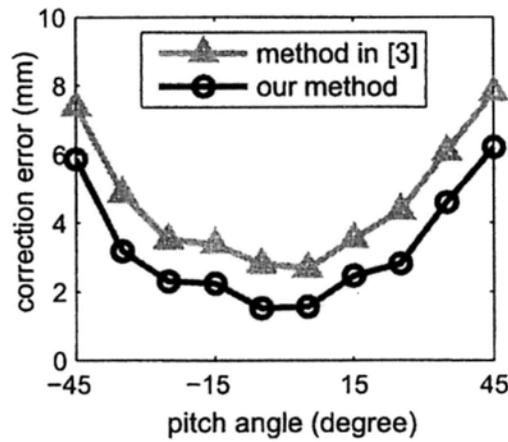


Figure 5.12: Comparison of our keystone correction module with the static projector keystone correction method proposed in [30].

mentioned in [30] for static projector keystone correction. The main difference between [30] and our method is that it requires a framed static screen and relies on the detection of the screen boundary. Similar to the above simulation, we place the projector at different pitch angles towards the virtual screen. For simulation of [30], a virtual frame with a known position is further added to the screen, and its position is disturbed with zero mean Gaussian noise with $std = 2.0$. The correction results of both algorithms are evaluated and compared. The corresponding result is shown in Fig. 5.12. Our algorithm outperforms [30] at all angles.

Speed Our current implementation can achieve a frame rate about 16 fps on our platform. No obvious latency is observed. The per-frame processing time is about 60 ms. The pre-warping step occupies most of the time (about 36 ms) due to the large resolution of the projection image. With a smaller projection resolution, the processing time will be dramatically reduced.

5.7 Discussions

To apply our method into real embedded projection devices, two problems should be further studied:

First, we project a green frame to guide the keystone correction in the current implementation, which is not user-friendly. To eliminate

it, we can mount four IR lasers with the projector. By detecting the spots of the lasers, we can recover the relative position between the screen and the projector similarly.

Second, all of the algorithms are implemented in the PC in the experiment. It is necessary to implement all of them into an embedded platform. As introduced, our method mainly contains three modules, the calibration module, the tracking module, and the keystone correction module. The calibration module is off-line, and its output is a 3×4 projection matrix. We can do the calibration beforehand and save the projection matrix in the embedded circuit. The particle filter tracking and the correction module mainly consist of some algebraic computation, which are not difficult to implement in embedded platforms. So in theory, it is feasible to implement them in an embedded system. According to our experiment in the PC, the major computation cost of the proposed method is occupied by the pre-warping step in the correction module, due to the large resolution of the projection image (1280×1024). This cost can be drastically decreased in embedded platforms since the resolution of projection image should be much smaller, usually 320×240 for most cell phones and PDAs. As the processing power of embedded devices (like smart phone) becoming stronger and stronger, we are confident that our method can be run smoothly in embedded platforms.

End of chapter.

Chapter 6

Conclusion and Future Work

In this thesis, we study several important and interesting problems in projector-based interactive visual processing, including 3D object interaction, flexible display, and mobile projector keystone correction, etc.

In Chapter 3, we propose a movable projector-based 3D object manipulation system using low-cost devices and computer vision technologies. The object tracking techniques and a commercially available tracking product (Wiimote) are used to track the translation and rotation of the sphere. The generation of the projection image is based on the translation and rotation of the sphere as well as the pre-calibrated geometric relationships. Extensive experiments show that our system can robustly track the movement of the sphere and correctly generate the projection image. It successfully creates the effect with satisfactory accuracy and robustness under different environments. It is believed to have a lot of applications in education

and entertainment area.

In Chapter 4, we propose a flexible projector-based hand-held display system using ordinary devices and computer vision technology. A projector and a camera pair are used as the projection and tracking device and an ordinary white paper is used as the flexible projection surface. No sensors or special hardware are needed in our system. An off-line flexible, easy and automatic calibration method is employed to calibrate the system. A real-time tracking and recovery algorithm is proposed to track and recover the 3D surface of the paper. The display content is pre-warped and projected to the paper based on the calibration result and the recovered surface. Two model applications are elaborated to demonstrate the potential of our system. Experimental results show that our system can successfully create the flexible display on the deformable surface with satisfactory accuracy and robustness. Future work will be carried out to improve the freedom of control and interactivity of the system.

Another interesting extension of the proposed method is to extend it for mobile applications. For example, the famous SixthSense system [40] can only display information on planar or nearly planar surfaces like walls, and floors etc. However, real world objects usually have non-planar or even flexible surfaces. It would be necessary that the system be able to project information on these natural surfaces. A possible solution is to recover the shape of the surface using

a variation of our algorithm before the real projection. Specifically, we can project a checker pattern onto the surface and then detect it by the camera. The detected checker pattern and the checker pattern in the projector will become a stereo correspondence. By using the algorithm proposed in Chapter 4, we can reconstruct the shape of the 3D non-rigid surface and then pre-warp the real projection image so that it will be correctly projected onto the non-rigid surface without distortion. Moreover, due to the efficiency of our algorithm, the process can be executed in real-time.

In Chapter 5, we propose an effective keystone correction method for a mobile projection system with continuous real-time keystone correction. Since our calibration method and correction mechanism are screen independent, no special display screen is needed for our method and the user can freely project the content onto where he or she likes. Mobility is the most distinguishing feature of our method, while experimental results obtained have also proved its accuracy and real-time processing capability. As a result, our prototype system is especially suitable for products like integrated camera-projector pair or mobile phones with build-in projectors. One limitation is the addition of the green frame projected to the screen, while may be a hurdle for some applications. Future work will be carried out to introduce invisible IR lasers to remove the requirement of having the green frame in the viewing area, and implement

the method on an embedded platform.

Through these investigations, the power of projectors has been well exhibited. However, the great potential of projectors is far more beyond those we discussed in this thesis. In addition, with the rapid evolution of hand-held devices such as PDAs and smart cell phones, we believe mobile devices with built-in projectors will become ubiquitous in the future. We also believe the applications using desktop projectors will be shifted to mobile platforms. It will certainly open a big research area for mobile projectors and related technologies.

□ **End of chapter.**

Bibliography

- [1] M. Arulamplam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transaction on Signal Processing*, 50(2):174–188, 2002.
- [2] M. Ashdown and Y. Sato. Steerable projector calibration. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Workshop*, 2005.
- [3] D. Bandyopadhyay, R. Raskar, and H. Fuchs. Dynamic shader lamps: Painting on movable objects. In *In Proceedings of Int. Symp. On Augmented Reality*, pages 207–216, 2001.
- [4] O. Bimber, D. Iwai, G. Wetzstein, and A. Grundhofer. The visual computing of projector-camera systems. In *SIGGRAPH 2008*, pages 1–25, 2008.
- [5] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d-faces. In *Proceedings of ACM SIGGRAPH*, pages 187–194, 1999.

- [6] S. Borkowski, O. Riff, and J. L. Crowley. Projecting rectified images in an augmented environment. In *International workshop on Projector Camera System*, 2003.
- [7] J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker.
- [8] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *Proceedings of CVPR*, 2000.
- [9] M. Brown, A. Majumder, and R. Yang. Camera-based calibration techniques for seamless multiprojector displays. *IEEE Trans. TVCG*, 2005.
- [10] A. Cassinelli and M. Ishikawa. Volume slicing display. In *Proceedings of ACM SIGGRAPH ASIA, Art Gallery and Emerging Technologies*, pages 88–88, 2009.
- [11] C. Cruz-Neira, D. J. Sandin, and T. A. Defanti. Surround-screen projection-based virtual reality: the design and implementation of the cave. In *Proceedings of ACM SIGGRAPH*, pages 135–142, 1993.
- [12] O. Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.

- [13] O. D. Faugeras, Q. T. Luong, and S. J. Maybank. Camera self-calibration: theory and experiments. In *European Conference on Computer Vision*, pages 321–334, 1992.
- [14] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*, 1981.
- [15] S. J. Gibbs, C. Arapis, and C. J. Breiteneder. Teleport – towards immersive copresence. *Multimedia System*, 7(3):214–221, 1999.
- [16] M. Grossberg, H. Peri, S. Nayar, and P. Belhumeur. Making one object look like another: controlling appearance using a projector-camera system. In *CVPR*, 2004.
- [17] S. Gupta and C. Jaynes. Active pursuit tracking in a projector-camera system with application to augmented reality. In *CVPR workshop*, 2005.
- [18] R. Hartley. Kruppa’s equations derived from the fundamental matrix. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 133–135, 1997.
- [19] R. I. Hartley. An algorithm for self calibration from several views. In *Intl. Conf. on Computer Vision and Pattern Recognition*, pages 908–912, 1994.

- [20] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [21] M. Hereld, I. Judson, and R. Stevens. Introduction to building projection based tiled display systems. *IEEE Computer Graphics and Applications*, 20(4):22–28, 2000.
- [22] R. Horaud, B. Conio, and O. Le Boulleux. An analytic solution for the perspective 4-point problem. In *Computer Vision Graphics and Image Processing*, 1989.
- [23] S. Ilic and P. Fua. Implicit meshes for surface reconstruction. *IEEE Trans. on PAMI*, 28(2):328–333, 2006.
- [24] Intel. OpenCV. <http://www.intel.com/technology/computing/opencv>.
- [25] D. Kondoh and R. Kijima. Proposal of a free form projection display using the principle of duality rendering. In *Proceedings of VSMM*, pages 346–352, 2002.
- [26] J. Konieczny, C. Shimizu, G. Meyer, and D. Colucci. A handheld flexible display system. In *Proceedings of IEEE Visualization*, pages 59–597, 2005.
- [27] J. C. Lee, S. E. Hudson, and E. Tse. Foldable interactive displays. In *ACM symposium on User Interface Software and Technology*, pages 287–290, 2008.

- [28] M. Leung, K. Lee, K. Wong, and M. Chang. A projector-based movable hand-held display system. In *Proceedings of International Conference On Computer Vision and Pattern Recognition*, pages 1109–1114, 2009.
- [29] levmar. levmar:levenberg-marquardt nonlinear least squares algorithms in c/c++. <http://www.ics.forth.gr/lourakis/levmar/>.
- [30] B. Li and I. Sezan. Automatic keystone correction for smart projectors with embedded camera. In *Intl. Conf. on Image Processing*, pages 2829–2832 Vol. 4, 2004.
- [31] K. Li, H. Chen, Y. Chen, D. Clark, P. Cook, S. Damianakis, G. Essl, A. Finkelstein, T. Funkhouser, T. Housel, A. Klein, Z. Liu, E. Praun, J. Singh, B. Shedd, J. Pal, G. Tzanetakis, and J. Zheng. Building and using a scalable display wall system. *Computer Graphics and Application, IEEE*, 20(4):29–37, 2000.
- [32] Z. R. Li, K. H. Wong, Y. B. Gong, and M. M. Chang. An effective method for movable projector keystone correction. *IEEE Transactions on Multimedia*, 13(1), 2001.
- [33] Z. R. Li, K. H. Wong, Y. B. Gong, and M. M. Chang. A low-cost projector-based hand-held flexible display system. *Multimedia Tools and Applications Journal, Springer*, 2011.

- [34] Z. R. Li, K. H. Wong, M. C. Leung, H. F. Ko, K. K. Lee, and M. M. Chang. An interactive hand-held spherical 3d object display system. *Multimedia Systems*, 2011.
- [35] M. L. Liu and K. H. Wong. Pose estimation using four corresponding points. *Pattern Recognition Letters*, 20, 1999.
- [36] K. L. Low, G. Welch, A. Lastra, and H. Fuchs. Life-sized projector-based dioramas. In *ACM symposium on Virtual reality software and technology*, pages 91–101, 2001.
- [37] D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Pattern Analysis and Machine Intelligence*, 13, 1991.
- [38] Q. T. Luong and O. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *The International Journal of Computer Vision*, 22(3):261–289, 1997.
- [39] S. J. Maybank and O. D. Faugeras. A theory of self-calibration of a moving camera. *The International Journal of Computer Vision*, 8(2):123–152, 1992.
- [40] P. Mistry and P. Maes. Sixthsense: a wearable gestural interface. In *ACM SIGGRAPH ASIA 2009 Sketches*, 2009.

- [41] T. Okatani and K. Deguchi. Autocalibration of a projector-camera system. *IEEE Trans. PAMI*, 2005.
- [42] C. Pinhanez. The everywhere displays projector. In *Ubiquitous Computing*, pages 315–331, 2001.
- [43] A. Raij and M. Pollefeys. Auto-calibration of multi-projector display walls. In *ICPR*, pages 14–17, 2004.
- [44] R. Raskar and P. Beardsley. A self-correcting projector. In *Intl. Conf. on Computer Vision and Pattern Recognition*, pages 504–508, 2001.
- [45] R. Raskar, M. S. Brown, R. G. Yang, W. C. Chen, G. Welch, H. Towles, W. B. Seales, and H. Fuchs. Multi-projector displays using camera-based registration. In *IEEE Visualization*, pages 161–168, 1999.
- [46] R. Raskar, K. Low, and G. Welch. Shader lamps: Animating real objects with image based illumination. *Technical report, Chapel Hill, NC, USA*, 2000.
- [47] R. Raskar and van B. Jeroen. Low-cost multi-projector curved screen displays. In *International Symposium Society for Information Display (SID)*, 2005.
- [48] R. Raskar, G. Welch, K. L. Low, and D. Bandyopadhyay. Shader lamps: Animating real objects with image-based illu-

- mination. In *Proceedings of Eurographics Workshop on Rendering Techniques*, pages 89–102, 2001.
- [49] M. Salzmann, R. Hartley, and P. Fua. Convex optimization for deformable surface 3d tracking. In *Proceedings of ICCV*, 2007.
- [50] M. Salzmann, J. Pilet, S. Ilic, and P. Fua. Surface deformation models for nonrigid 3d shape recovery. *IEEE Trans. on PAMI*, 29(8):1481–1487, 2007.
- [51] N. Sukaviriya, M. Podlaseck, R. Kjeldsen, A. Levas, G. Pingali, and C. Pinhanez. Embedding interactions in a retail store environment: The design and lessons learned. In *IFIP INTERACT*, 2003.
- [52] R. Sukthankar, R. Stockton, and M. Mullin. Smarter presentations: Exploiting homography in camera-projector systems. In *Intl. Conf. on Computer Vision*. IEEE, 2001.
- [53] J. Summet and R. Sukthankar. Tracking locations of moving hand-held displays using projected light. In *Proceedings of Pervasive*, pages 37–46, 2005.
- [54] B. Triggs. Autocalibration from planar scenes. In *European Conference on Computer Vision*, pages 89–105, 1998.

- [55] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment: a modern synthesis. In *International Workshop on Visual Algorithms: Theory and Practice*, 1999.
- [56] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.
- [57] K. Y. Wong, D. Schnieders, and S. Li. Recovering light directions and camera poses from a single sphere. In *Proceedings of ECCV*, pages 631–642, 2008.
- [58] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+3d active appearance models. In *Proceedings of CVPR*, 2004.
- [59] R. Yang, D. Gotz, J. Hensley, H. Towles, and M. Brown. Pixelflex: a reconfigurable multi-projector display system. In *IEEE Visualization*, pages 167–554, 2001.
- [60] Y. K. Yu, K. H. Wong, and M. M. Chang. Pose estimation for augmented reality applications using genetic algorithm. *IEEE Transactions on System, Man and Cybernetics*, 35(6):1295–1301, 2005.

- [61] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(11), 2000.
- [62] Z. Zhang. Camera calibration with one-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 892–899, 2004.
- [63] J. Zhu, S. C. H. Hoi, Z. Xu, and M. R. Lyu. An effective approach to 3d deformable surface tracking. In *Proceedings of ECCV*, 2008.