Use of Projector-Camera System for Human-Computer Interaction

DAI, Jingwen

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Mechanical and Automation Engineering

Supervised by

Prof. Ronald Chung

The Chinese University of Hong Kong
September 2012

Abstract

The use of a projector in place of traditional display device would dissociate display size from device size, making portability much less an issue. Associated with camera, the projector-camera system allows simultaneous video display and 3D acquisition through imperceptible structured light sensing, providing a vivid and immersed platform for natural human-computer interaction. Key issues involved in the approach include: (1) *Simultaneous Display and Acquisition*: how to make normal video projector not only a display device but also a 3D sensor even with the prerequisite of incurring minimum disturbance to the original projection; (2) *3D Information Interpretation*: how to interpret the spare depth information with the assistance of some additional cues to enhance the system performance; (3) *Segmentation*: how to acquire accurate segmentation in the presence of the incessant variation of the projected video content; (4) *Posture Recognition*: how to infer 3D posture from single image. This thesis aims at providing improved solutions to each of these issues.

To address the conflict between imperceptibility of the embedded codes and the robustness of code retrieval, noise-tolerant schemes to both the coding and decoding stages are introduced. At the coding end, specifically designed primitive shapes and large Hamming distance are employed to enhance tolerance toward noise. At the decoding end, pre-trained primitive shape detectors are used to detect and identify the embedded codes – a task difficult to achieve by segmentation that

is used in general structured light methods, for the weakly embedded information is generally interfered by substantial noise.

On 3D information interpretation, a system that estimates 6-DOF head pose by imperceptible structured light sensing is proposed. First, through elaborate pattern projection strategy and camera-projector synchronization, pattern-illuminated images and the corresponding scene-texture image are captured with imperceptible patterned illumination. Then, 3D positions of the key facial feature points are derived by a combination of the 2D facial feature points in the scene-texture image localized by AAM and the point cloud generated by structured light sensing. Eventually, the head orientation and translation are estimated by SVD of a correlation matrix that is generated from the 3D corresponding feature point pairs over different frames.

On the segmentation issue, we describe a coarse-to-fine hand segmentation method for projector-camera system. After rough segmentation by contrast saliency detection and mean shift-based discontinuity-preserved smoothing, the refined result is confirmed through confidence evaluation.

Finally, we address how an HCI (Human-Computer Interface) with small device size, large display, and touch input facility can be made possible by a mere projector and camera. The realization is through the use of a properly embedded structured light sensing scheme that enables a regular light-colored table surface to serve the dual roles of both a projection screen and a touch-sensitive display surface. A random binary pattern is employed to code structured light in pixel accuracy, which is embedded into the regular projection display in a way that the user perceives only regular display but not the structured pattern hidden in the display. With the projection display on the table surface being imaged by a camera, the observed image data, plus the known projection content, can work together to probe the 3D world immediately above the table surface, like deciding if there is a

finger present and if the finger touches the table surface, and if so at what position on the table surface the finger tip makes the contact. All the decisions hinge upon a careful calibration of the projector-camera-table surface system, intelligent segmentation of the hand in the image data, and exploitation of the homography mapping existing between the projector's display panel and the camera's image plane.

摘要

用投影機替代傳統的顯示器可在較小尺寸的設備上得到較大尺寸的顯示，從而彌補了傳統顯示器移動性差的不足。投影機－照相機系統通過不可感知的結構光，在顯示視頻內容的同時具備了三維傳感能力，從而可為自然人機交互提供良好的平臺。投影機－照相機系統在人機交互中的應用主要包括以下四個核心內容：（1）同時顯示和傳感，即如何在最低限度的影響原始投影的前提下，使得普通視頻投影機既是顯示設備又是三維傳感器；（2）三維信息的理解：即如何通過利用額外的信息來彌補稀疏點云的不足，從而改善系統性能；（3）分割：即如何在不斷變化投影內容的影響下得到準確的分割；（4）姿態識別：即如何從單張圖像中得到三維姿態。本文將針對上述四個方面進行深入的研究和探討，並提出改進方案。

首先，為了解決嵌入編碼不可見性与編碼恢復魯棒性之間的矛盾，本文提出一種在編解碼兩端同時具備抗噪能力的方法。我們使用特殊設計的幾何圖元和較大的海明距離來編碼，從而增強了抗噪聲干擾能力。同時在解碼端，我們使用事先通過訓練得到的幾何圖元檢測器來檢測和識別嵌入圖像的編碼，從而解決了因噪聲干擾使用傳統結構光中的分割方法很難提取

嵌入編碼的困難。

其次在三維信息的理解方面，我們提出了一個通過不可感知結構光來實現六自由度頭部姿態估計的方法。首先，通過精心設計的投影策略和照相機－投影機的同步，在不可感知結構光的照射下，我們得到了模式圖和與之相對應的紋理圖。然後，在紋理圖中使用主動表觀模型定位二維面部特徵，在模式圖中通用結構光方法計算出點雲坐標，結合上述兩種信息來計算面部特征點的三維坐標。最后，通過不同幀中對應特征點三維坐標間的相關矩陣的奇異值分解來估計頭部的朝向和位移。

在分割方面，我們提出一種在投影機－照相機系統下由粗到精的手部分割方法。首先手部區域先通過對比度顯著性檢測的方法粗略分割出來，然後通過保護邊界的平滑方法保證分割區域的一致性，最后精确的分割結果由置信度分析得到。

最後，我們又探討如何僅使用投影機和照相機將在普通桌面上的投影區域轉化成觸摸屏的方案。我們將一種經過統計分析得到的隨機二元編碼嵌入到普通投影內容中，從而在用戶沒有感知的情況下，使得投影機－照相機系統具備三維感知的能力。最終手指是否觸及桌面是通過投影機－照相機－桌面系統的標定信息，精准的手部區域分割和手指尖定位，投影機投影平面与照相機圖像平面的單應映射以及嵌入投影的編碼來確定。

Acknowledgement

In what follows, grateful acknowledgement is given to everyone who assisted and inspired me throughout my PhD study.

First and foremost, I would like to thank my supervisor, Professor Ronald Chung, for his patience, support, encouragement and investment of time over the past three years. I was so fortunate and proud to have him as my supervisor who engages himself rigorously and earnestly in scientific research and truly cares about training his students to become better scientists. Furthermore, it was Professor Chung who gave me the opportunity to go to different flagship conferences in China, the United States and Europe, thereby allowing me to have the opportunity to present my work and have fruitful discussions with the researchers all over the world. I would also like to express my great appreciation to Professor Jianbo Su, my master supervisor in Shanghai Jiao Tong University. During my PhD study, he was always there to provide valuable advice and instruction when necessary. The discussions about potential research direction were always enlightening to me.

I would also like to express my great appreciation to my officemate, Dr. Zibin Wang, who imparted his knowledge to me and offered kind help to me during my first two years of PhD study. I deeply acknowledge the help and support of Ms. Mu Fang, Ms. Wuyuan Xie and Mr. James Hui, it was happy to have them in the lab, and to have inspiring chats about study, research, entertainment, and Hong Kong life.

I am also deeply indebted for the help and support from the Department of Mechanical and Automation Engineering, which has provided an incredibly supportive environment. I would like to thank the administrative staff and the technical support staff: Ms. Yuet-lin Kan, Joyce Wong, Maggie Chan, Winnie Wong, Ms. Kit-wah Yau, Allan Mok, Thomas Lau, Karina Djin, and Ms. Yuk-kuen Chan.

I would like to express my gratitude and love to Ms. Jiang Zhao, who is my fiancée and soon will be my wife, for supporting me with all her heart. Her critical thinking, determination and perseverance endowed me great motivation and encouragement.

Last but not least, I would like to give my sincere thanks to my beloved mother and father, for taking care of me, loving me and always helping me out of difficulties that I encountered during my study, without a word of complaint. Without their support, I would not have even made this thesis into a reality.

To my parents

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The increasing capabilities and declining cost make video projectors widespread and established presentation tools. Being able to generate images that are larger than the actual display device virtually anywhere is an interesting feature for many applications that cannot be provided by desktop screens. Many researchers discovered this potential by applying projectors in unconventional ways to develop new and innovative information displays that go beyond simple screen presentations.

The adoption of structured light illumination has been proven to be an effective and accurate visual means for 3D reconstruction [147, 82]. The system consists of a projector that projects controlled patterns to the target object, and a camera capturing images of the illuminated object. Once correspondences between positions on the projector's pattern panel and positions on the camera's image plane are established through the use of some elaborately designed coding strategies on

the illuminated patterns, simple triangulation over the light rays from the projector and the corresponding light rays to the camera would recover 3D information about the target object. Recently, the availability of pico projectors with average dimensions of $4 \times 2 \times 1$ inches has widely extended the application area of structured light system.

On the other hand, HCI (Human-Computer Interface) has been traversing from firstly punch card and LEDs, then paper tape and CRO display, more recently mouse-plus-keyboard and LCD panel, and now fingers and touch-sensitive display panel over the history of development. Technologies have been ever improving, with the data-input mechanism growing only more natural, and the display only more vivid. Indeed for the input-output interface of computers, scarcely anything could be more natural than using our body to manipulate the computers (such as head pose, facial expression, hands, body gesture etc.).

Motivated by the aforementioned facts, this thesis is mainly focus on using projector-camera system in natural human-computer interaction. By endowing the projector-camera system the capability of simultaneous video content display and 3D acquisition, the ProCams will provide a vivid, natural and accurate platform for human-computer interaction.

1.2 Challenges

1.2.1 Simultaneous Display and Acquisition

In a variety of projector-camera systems we have often wished to operate cameras and projectors simultaneously. Unfortunately, conflicting lighting requirements have made such systems very difficult to realize: cameras need brightly lighted environments, whereas projectors need dark rooms. An additional difficulty

for cameras, especially for those performing 3D acquisition, has been the lack of strong features in many parts of the environment. Skin, clothes often image with nearly uniform surface structure, making depth acquisition that relies on stereo correspondence particularly difficult to perform. Using structured light to illuminate the scene solves this problem, however it is highly distracting and therefore not suitable for human-populated environments.

Some researchers designed structured light system in the non-visible spectrum [48]. That way the media for regular projection and structure light sensing can be made separate. However, additional hardware could be reduced and device size could be diminished if structured light and regular projection can be achieved through the same projector.

Therefore, how to make normal video projector not only a display device but also a 3D sensing is one challenge for the use of projector-camera system in human-computer interaction.

1.2.2 3D Information Interpretation

Through multiple projection-capture cycle using temporal coding scheme, the projector-camera system has the capability to derive dense even pixel-wise point cloud. Nevertheless, for the applications of human-computer interaction, due to the fast movement of the human body, it is impossible to use time multiplexing methods to acquire 3D information. Instead, the spatial multiplexing methods is adopted, since one single image is necessary. But the disadvantage is that the spatial coding scheme could derive sparse and even inaccurate point cloud, which will influence the performance of HCI system.

As a result, how to interpret the spare depth information with the assistance of some additional information (such as 2D texture image, epipolar constraints, smooth constraints, continuity constraints and homography constraints etc.) to

enhance the system performance is an interesting problem.

1.2.3 Segmentation

The segmentation, as the first step for most natural human-computer interfaces , plays an important role in the robustness, accuracy and efficiency of a HCI system. However, in the projector-camera scenario, it is a challenging task in the presence of the incessant variation of the projected video content and the shadow cast by the human body. Moreover, as the initial step of the HCI system, it is not allowed allocate too much resource (computing time and computing power) to segmentation task, especially in real-time applications and mobile applications.

Thus, how to segment the body part precisely and efficiently in complex foreground and background is a question deserving to be considered.

1.2.4 Posture Recognition

In typical projector-camera system setup, only one camera has the sensing capability, so in a certain time instant, only one image could be captured. For most touch based interface, the touch action detection is a principal task. Even for this simple task, the challenge is, from a single image alone there is generally difficulty in even distinguishing whether there is a physical contact between finger and table surface, let alone identifying where the touch takes place or what the touch gesture is. The inevitable self-occlusion of the fingers will aggravate the ambiguity. The facility of acquiring certain 3D information about the illuminated workspace would be of much aid.

Hereby, how to make use of the 3D information to relax the ambiguity is another challenge.

1.3 Objective

The main objective of this thesis consists in developing a projector-camera system that can provide a platform for the user interacting with computer in a natural way. All the three key issues: 3D information interpretation, display and sensing, and human action recognition. In details, the objectives can be generalized as follows:

1. Propose a novel approach of 6-DOF head pose estimation from imperceptible structured light sensing to evaluate the validity of projector-camera system for human-computer interaction and to study the way of combining 2D texture information with 3D depth information.

2. Propose an approach of embedding codes into projection display for structured light based sensing, with the purpose of letting projector serve as both a display device and a 3D sensor.

3. Propose a novel segmentation method specialize for the projector-camera scenario to derive the accurate position of human body (e.g. hand and arm) operating under the projector illumination.

4. Develop a system that transfer arbitrary planar surface to touch-sensitive display by mere a projector and camera.

1.4 Organization of the Thesis

A comprehensive study of related previous work is addressed in Chapter 2. The survey is conducted from two aspects: projector-camera system and natural human-computer interaction. For projector-camera system, the most prevalent projection technologies are introduced. And the recent researches on projector-camera system (ProCams) are reviewed, including system calibration, traditional and embed-

ded structured light sensing, and ProCams in interaction. For the natural human-computer interaction, two channels, head pose and hand gesture are mainly reviewed.

In Chapter 3, we describe a method of estimating head pose estimation from imperceptible structured light sensing. First, through elaborate pattern projection strategy and camera-projector synchronization, pattern-illuminated images and the corresponding scene-texture image are captured with imperceptible patterned illumination. Then, 3D positions of the key facial feature points are derived by a combination of the 2D facial feature points in the scene-texture image localized by AAM and the point cloud generated by structured light sensing. Eventually, the head orientation and translation are estimated by SVD of a correlation matrix that is generated from the 3D corresponding feature point pairs over different frames. Extensive experiments show that the proposed method is effective, accurate and rapid in 6-DOF head pose estimation, making it suitable for real-time application.

In Chapter 4, we describe an approach of embedding codes into projection display for structured light based sensing, with the purpose of letting projector serve as both a display device and a 3D sensor. The challenge is to make the codes imperceptible to human eyes so as not to disrupt the content of the original projection. There is the temporal resolution limit of human vision that one can exploit, by having a higher than necessary frame rate in the projection and stealing some of frames for code projection. Yet there is still the conflict between imperceptibility of the embedded codes and the robustness of code retrieval that has to be addressed. We introduce noise-tolerant schemes to both the coding and decoding stages. At the coding end, specifically designed primitive shapes and large Hamming distance are employed to enhance tolerance toward noise. At the decoding end, pre-trained primitive shape detectors are used to detect and identify the embedded codes – a task difficult to achieve by segmentation that is used in

general structured light methods, for the weakly embedded information is generally interfered by substantial noise. Extensive experiments including evaluations of code imperceptibility, decoding accuracy and sensitivity analysis show that the proposed system is effective, even with the prerequisite of incurring minimum disturbance to the original projection.

One goal of projector-camera system is let human finger be used like a mouse to click and drag objects in the projected content. It requires segmentation of the human palm and fingers in the image data captured by the camera, which is a challenging task in the presence of the incessant variation of the projected video content and the shadow cast by the palm and fingers. In Chapter 5, we describe a coarse-to-fine hand segmentation method for projector-camera system. After rough segmentation by contrast saliency detection and mean shift-based discontinuity-preserved smoothing, the refined result is confirmed through confidence evaluation. Extensive experimental results are shown to illustrate the accuracy and efficiency of the approach.

In Chapter 6, we address how an HCI (Human-Computer Interface) with small device size, large display, and touch input facility can be made possible by a mere projector and camera. The realization is through the use of a properly embedded structured light sensing scheme that enables a regular light-colored table surface to serve the dual roles of both a projection screen and a touch-sensitive display surface. A random binary pattern is employed to code structured light in pixel accuracy, which is embedded into the regular projection display in a way that the user perceives only regular display but not the structured pattern hidden in the display. With the projection display on the table surface being imaged by a camera, the observed image data, plus the known projection content, can work together to probe the 3D world immediately above the table surface, like deciding if there is a finger present and if the finger touches the table surface, and if so at what

position on the table surface the finger tip makes the contact. All the decisions hinge upon a careful calibration of the projector-camera-table surface system, intelligent segmentation of the hand in the image data, and exploitation of the homography mapping existing between the projector's display panel and the camera's image plane. Extensive experimentation including evaluation of the display quality, touch detection accuracy, trajectory tracking accuracy, multi-touch capability and system efficiency are shown to illustrate the feasibility of the proposed realization.

Chapter 7 summarizes the contributions in this thesis and related publications. How future work could extend on the results of this thesis is also discussed.

Chapter 2

Background

2.1 Projector-Camera System

Systems that utilize controllable lighting systems with light sensing devices facilitate a wide range of applications. Examples include 3D scanning, flexible display walls, novel display interaction, reflectance field capture, optical communication, and artistic creations. While the term ”projector-camera system” is used, such systems encompass any approach that employs a controllable light-source, ranging from LEDs to an array of light projectors, with any light sensing device, ranging from a simple photo-sensor to an array of high-resolution wide-field-of-view cameras. In the rest part of this section, a brief review on projection technologies and some researches in ProCams is presented.

2.1.1 Projection Technologies

Projectors share a common history with cameras. The first known record of what might portray the idea of projecting an image on a surface is a drawing by Johannes de Fontana from 1420. The drawing was of a nun holding something that might be a lantern. The lantern had a small translucent window that contained an image of a devil holding a lance. These drawings are likely to have inspired the creation of the earliest image projector, a device called a *magic lantern* [172]. In the 1950s to the 1970s, the type of projector called slide projectors were common as a form of entertainment; family members and friends would gather to view slideshows. Late in the 20th century, slides and transparencies were replaced with digital images.

A video projector is an image projector that receives a video signal and projects the corresponding image on a projection screen using a lens system. All video projectors use a very bright light to project the image. Video projectors are widely used for many applications such as, conference room presentations, classroom training, home theatre and concerts. Since the video projector is one of the key components of projector-camera system (ProCams), here I will summarize the main technologies in recent video projectors.

LCD Projector

LCD (Liquid Crystal Display) projectors contain three separate LCD glass panels, one for red, green, and blue components of the image signal being transfered to the projector. As the light passes through the LCD panels, individual pixels can be opened to allow light to pass or closed to block the light. This activity modulates the light and produces the image that is projected onto the screen. The conceptual diagram of the LCD technology is shown in Fig. 2.1.

The lamp provides white light that passes through a polarizing filter. Polariz-

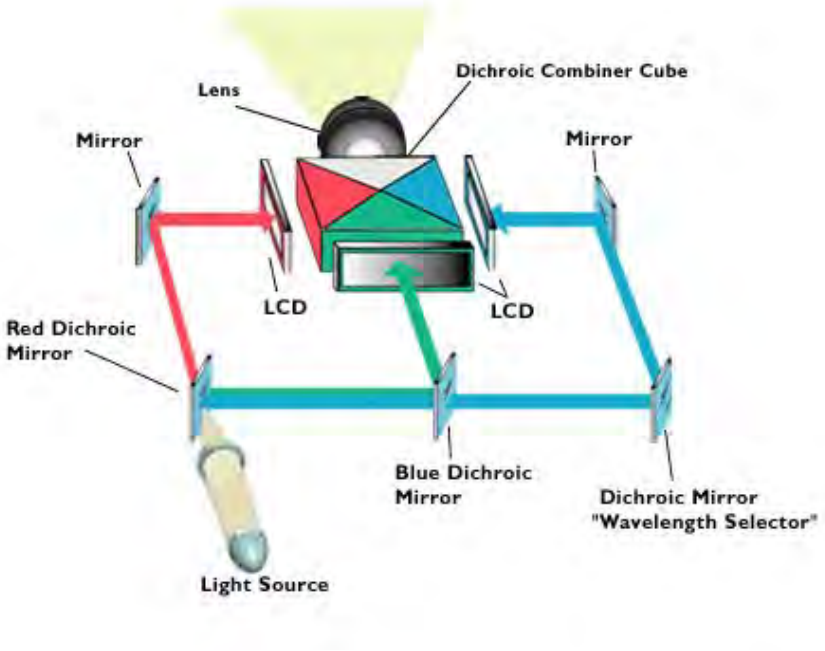


Figure 2.1: Conceptual diagram of the LCD technology.

ing works by accepting light that is traveling on the same plane. All other light will be blocked. From the polarizing filter the light is then passed through a series of dichroic mirrors. Dichroic mirrors work by only allowing certain colors in the light spectrum to be reflected, while others pass through. The dichroic mirrors in LCD projectors separate the light into the three primary colors: green, red and blue. These three colors are then sent to a separate LCD panel. From there the LCD panels send the light through the dichroic prism which recombines the light and sends it out the main lens in the LCD projector to the surface against which it is projected. Each LCD is only capable of controlling one color.

LCD panels in LCD projectors work by allowing the polarized light to travel through a pane of glass into the liquid crystal inside the display. The liquid crystals bend the light, and it is traveling on a different plane then when it entered through the polarizing filter. If you apply an electrical current to the liquid crystal they

will align, allowing the light to pass through on the same plane as when it entered. If you add a second polarizing filter at the other end of the liquid crystal you can then effectively block all light from passing through. Each LCD panel has a separate system to control the electrical current that passes through the liquid crystal, allowing each to be controlled individually.

DLP Projector

DLP (Digital Light Processing) is a proprietary technology developed by Texas Instruments. It works quite differently than LCD. Instead of having glass panels through which light is passed, the DLP chip is a reflective surface made up of thousands (or millions) of micromirrors. Each mirror represents a single pixel. The micromirror architecture is illustrated in Fig. 2.2.

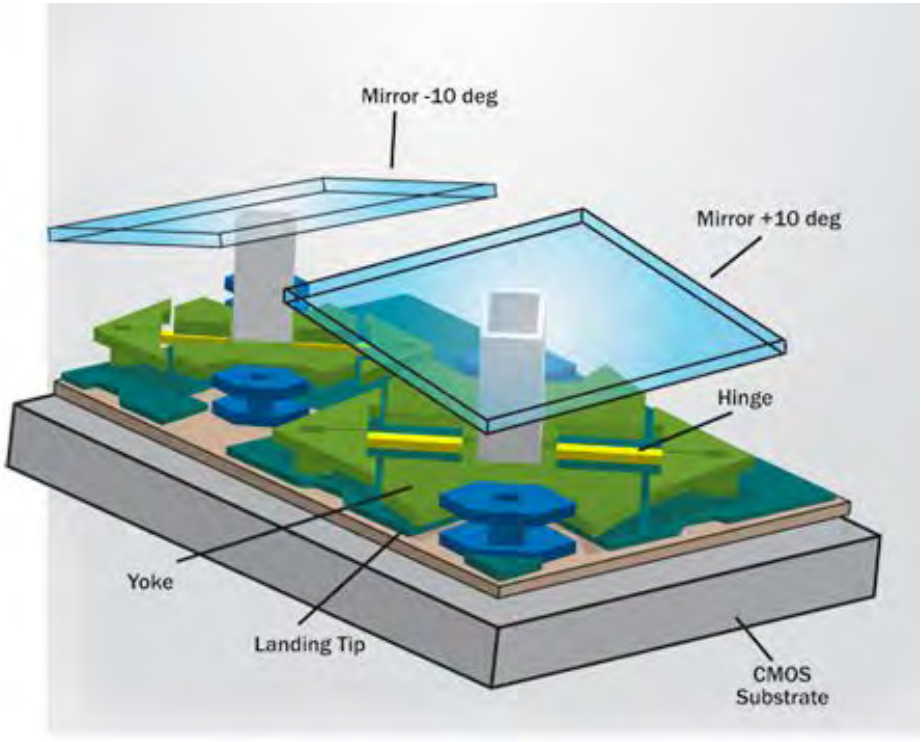


Figure 2.2: Micromirror architecture.

Before any of the mirrors switch to their on or off positions, the chip will rapidly decode a bit-streamed image code that enters through the semiconductor. It then converts the data from interlaced to progressive, allowing the picture to fade in. Next, the chip sizes the picture to fit the screen and makes any necessary adjustments to the picture, including brightness, sharpness and color quality. Finally, it relays all the information to the mirrors, completing the whole process in just 16 microseconds.

The mirrors are mounted on tiny hinges that enable them to tilt either toward the light source (ON) or away from it (OFF) up to $\pm 12°$, and as often as $5,000$ times per second. When a mirror is switched on more than off, it creates a light gray pixel. Conversely, if a mirror is off more than on, the pixel will be a dark gray. The light they reflect is directed through a lens and onto the screen, creating an image. The mirrors can reflect pixels in up to $1,024$ shades of gray to convert the video or graphic signal entering the DLP into a highly detailed grayscale image. DLPs also produce the deepest black levels of any projection technology using mirrors always in the off position.

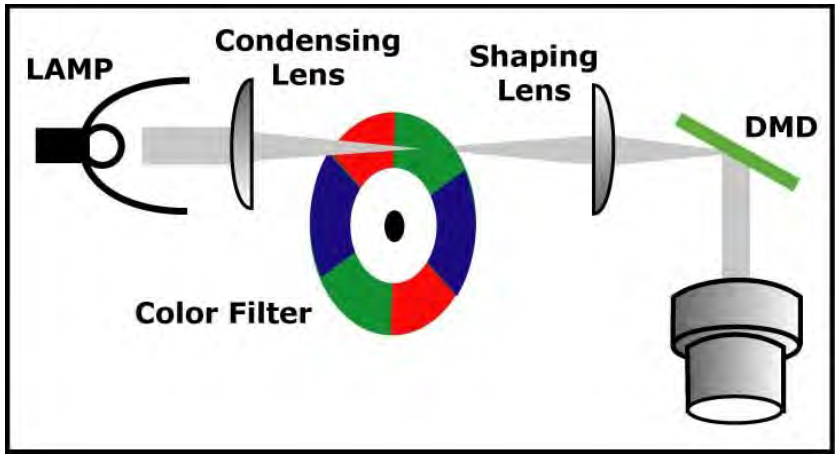


Figure 2.3: Conceptual diagram of the DLP technology.

As shown in Fig. 2.3, to add color to that image, the white light from the

lamp passes through a transparent, spinning color wheel, and onto the DLP chip. The color wheel, synchronized with the chip, filters the light into red, green and blue. The on and off states of each mirror are coordinated with these three basic building blocks of color. A single chip DLP projection system can create 16.7 million colors.

Each pixel of light on the screen is red, green or blue at any given moment. The DLP technology relies on the viewer's eyes to blend the pixels into the desired colors of the image. For example, a mirror responsible for creating a purple pixel will only reflect the red and blue light to the surface. The pixel itself is a rapidly, alternating flash of the blue and red light. Our eyes will blend these flashes in order to see the intended hue of the projected image.

Up to now, the LCD and DLP are the most mature technologies in consumer market. Fig. 2.4 lists the comparison of the two technologies.

	DLP	LCD
Pros.	○ Sealed imaging chip ○ Filter-free ○ No convergence problems ○ Contrast advantages ○ No image persistence ○ No degradation of image quality over time ○ Less pixilation effect on low resolution products ○ Leads in miniaturization	○ Better price/performance in HT products ○ Fewer artifacts/greater image stability ○ Sharper image with data display ○ Greater installation flexibility in HT products ○ Better light efficiency, less power usage
Cons.	● Color wheels can produce rainbow artifacts ● Color saturation ● Dithering artifacts ● Restricted compatibility with zoom lenses	● Unknown lifespan of LCD panels ● Lower contrast ratings in business products ● Susceptible to dust spots

Figure 2.4: Pros and Cons of DLP and LCD.

LCoS projector

Liquid crystal on silicon (LCoS or LCOS) is a ”micro-projection” or ”micro-display” technology typically applied in projection. It is a reflective technology similar to DLP projectors; however, it uses liquid crystals instead of individual mirrors. By way of comparison, LCD projectors use transmissive LCD chips, allowing light to pass through the liquid crystal. In LCoS, liquid crystals are applied directly to the surface of a silicon chip coated with an aluminized layer, with some type of passivation layer, which is highly reflective. The conceptual diagram of LCoS is shown in Fig. 2.5.

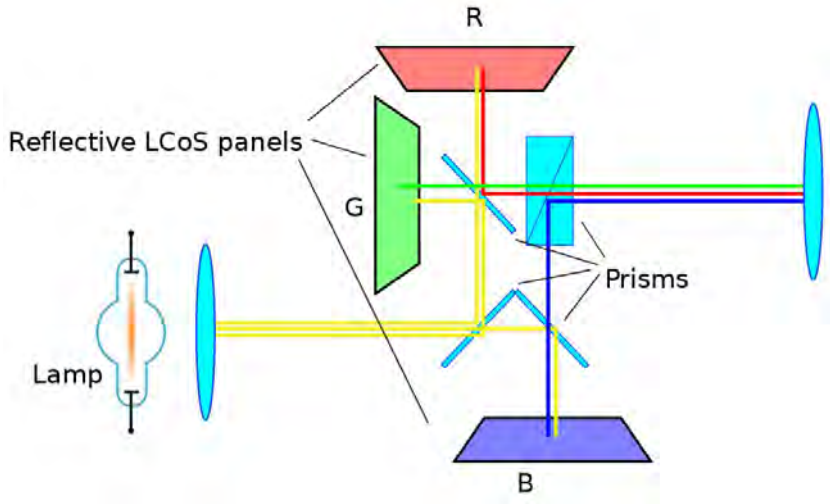


Figure 2.5: Conceptual diagram of the LCoS technology.

LCoS technology can typically produce higher resolution and higher contrast images than LCD technologies, which makes it less expensive to implement in such devices as projection televisions.

LED projectors

LED projectors use one of the above mentioned technologies for image creation, with a difference that they use an array of Light Emitting Diodes (LED) as the

light source, negating the need for lamp replacement. By using LED, the heat problem caused by traditional light source is resolved to a certain extend, and the exemption of fans leads in miniaturization. Almost all the pico projectors employ LED as light source.

Laser video projector

A laser video projector is a video projector that modulates a laser beam in order to project a raster-based image. The systems work either by scanning the entire picture a dot at a time and modulating the laser directly at high frequency, much like the electron beams in a CRT, or by optically spreading and then modulating the laser and scanning a line at a time, the line itself being modulated in much the same way as with DLP. When well implemented this technology produces the broadest color gamut available in practical display equipment today, because lasers produce truly monochromatic primaries.

Due to the special features of laser projectors, such as a high depth of field, it is possible to project images or data onto a screen at an arbitrary distance. Such laser projection techniques are used in hand-held projectors, and for flight simulators and other virtual reality applications.

2.1.2 Researches in ProCams

There continues to be growing interest in systems that combine projection technology with computer vision. Examples include automatically calibrated display walls, interactive display surfaces, intelligent environments and performance art. A characteristic of these systems is their ability to passively sense an environment in support of real-time control of projected light. Research in this area spans a number of disciplines including computer vision, computer graphics, HCI and

display technologies. In the rest of this section, several directions related to our work in ProCams domain will be reviewed briefly.

ProCams Calibration

Projector-camera systems are adopted in many applications such as measurement and spatial augmented reality. Calibration is a crucial step involving the determination of the intrinsic parameters of both the camera and the projector that constitute the device and the extrinsic parameters between the two instruments. Most current methods to geometrically calibrate projector-camera systems operate in two phases: camera is calibrated firstly and then the projector [135, 92, 55].

To reduce the amount of work, methods have been proposed to integrate both calibrations together. They either exploit structured light [100, 154] or color channels [101, 139], or both for one-shot structured light.

In [154], a calibration design that makes use of a liquid-crystal display (LCD) panel as the calibration plane are presented. Whereas patterns displayed on it are used for camera calibration, patterns projected onto and reflected by it when it is set to total dark are used for projector calibration. The LCD panel's planarity is of industrial grade and is thus far more dependable; The pattern shown on the LCD panel is programmable and is thus convenient to produce in high precision.

In [140], a user-friendly method to perform full geometric calibration of projector-camera system is proposed. It is based on fiducial markers typically used for augmented reality applications. Half of the markers are physically printed, and half are projected using the projector. Each marker on its own carries information and can easily be identified, this allows the machine to pre-warp markers that are projected, in a way that they do not interfere with printed markers. Moreover, markers do not need color, and unlike structured light users can hold the calibration board in their hands.

Structured Light Sensing

The most straightforward application of projector-camera system is structured light sensing. The adoption of structured light illumination has been proven an effective and accurate visual means for 3D reconstruction. The system consists of a projector that illuminates controlled pattern or patterns to the target object, and a camera grabbing image or images of the illuminated object. Once correspondences between positions on the projector's pattern panel and positions on the camera's image plane are established, simple triangulation over light rays from the projector and the corresponding light rays to the camera would recover 3D information about the target object.

The coding methods in structured light were classified into two classes: temporal and spatial codification. The temporal coding schemes generate the codewords by projecting a sequence of patterns along time, so the pattern structure can be very simple. Since multiple images are used, dense surface points can be reconstructed by this means. Spatial codification represents the codeword in a unique pattern. And its structure is often complex. Single or fewer pattern images are usually used in spatial codification. That makes its output 3D model with lower data density. In addition, there is also another hybrid class that combined the features of temporal and spatial coding strategies. More detailed discussions about structured light sensing technologies are summarized in the surveys [81, 82].

Embedded Structured Light

Temporal modulation of projected images is about integrating coded patterns into the projection, in a way that the coded patterns are not noticeable to the users under limitation of human vision. Synchronized cameras, however, are able to detect and extract these codes. The principle was firstly described by Raskar et al. [136], and has been enhanced by Cotting et al. [34]. It is referred to as

embedded imperceptible pattern projection. Extracted code patterns, for instance, allow simultaneous acquisition of scene depth and texture.

The first applicable imperceptible pattern projection technique was presented in [34], in which DLP (digital light processing) projector was utilized. A core component of DLP projector is a CMOS IC named DMD (Digital Micro-mirror Device), whose top surface is composed of a dense array of tiny mirrors, each corresponding to a single pixel in the 2D image to be projected. Each of the mirrors could be turned to one of two stable positions (on/off, or active/inactive). The projected intensity of a pixel is determined by the percentage of time that its mirror is active rather than inactive. In [34], imperceptible structured light was achieved by letting a specific time slot called BIEP (binary image exposure period), of the DLP projection sequence, be occupied exclusively for displaying a binary pattern within a single color channel (multiple color channels are used in [35] to differentiate between multiple projection units). A camera that is synchronized exactly to this projection sequence can capture the embedded binary codes. In the selected BIEP the mirror flip sequences (each over a particular projection pixel) are not necessarily evenly distributed over all possible intensities. Thus, the intensity of each original projected pixel might have to be modified to ensure that the mirror state is active (which encodes the desired binary value at the particular pixel). This, however, can result in non-uniform intensity fragmentation and substantial reduction of the tonal values. In practice, dithering techniques are used to diffuse the artifacts.

Another scheme of integrating imperceptible code patterns is to modulate the intensity of the projected image I spatially over the pattern domain. The result is the code image I_{cod}. In addition, a compensation image I_{com} is computed in such a way that $(I_{cod} + I_{com})/2 = I$. If both images are projected alternately in a high enough speed, human observers will perceive I due to the slower

temporal integration nature of the human visual system. This has been demonstrated in [136]. The problem with this rather simple technique is that the code could be visible during eye movements or code transitions, as the code image and compensation image could be a little misaligned in the human visual perception under such circumstances. In [56] properties of human perception, like adaptation limitations to local contrast changes, are taken into account for adapting the coding parameters depending on local characteristics, such as spatial frequencies and local luminance values of image and code. This makes a truly imperceptible temporal coding of binary information possible. For binary codes, I is regionally decreased ($I_{cod} = I - \Delta$ to encode a binary 0) or increased ($I_{cod} = I + \Delta$ to encode a binary 1) in intensity by the amount of Δ, while the compensation image is computed with $I_{com} = 2I - I_{cod}$. The code can then be recovered from the two corresponding images (C_{cod} and C_{com}) captured by the camera, by examining the sign of $C_{cod} - C_{com}(< \mid = \mid >)0$. Thereby, Δ is one coding parameter that is locally adapted.

In [127], another technique for adaptively embedding complementary patterns into projected images is presented. In this work, the embedded code intensity is regionally adapted depending on the spatial variation of neighboring pixels and their color distribution in the YIQ color space. The final code contrast of Δ is then calculated depending on the estimated local spatial variations and color distributions. In [144], the binary temporal coding technique was extended to encoding intensity values as well. For this, the code image is computed with $I_{cod} = I\Delta$ and the compensation image with $I_{com} = 2(I - \Delta)$. The code can be extracted from the camera images with $\Delta = 2C_{cod}/(C_{cod} + C_{com})$. In general, temporal coding is not limited to the projection of two images only. Multiple code and compensation images can be projected if the display frame-rate is high enough, which requires high speed projectors and cameras.

ProCams based interaction

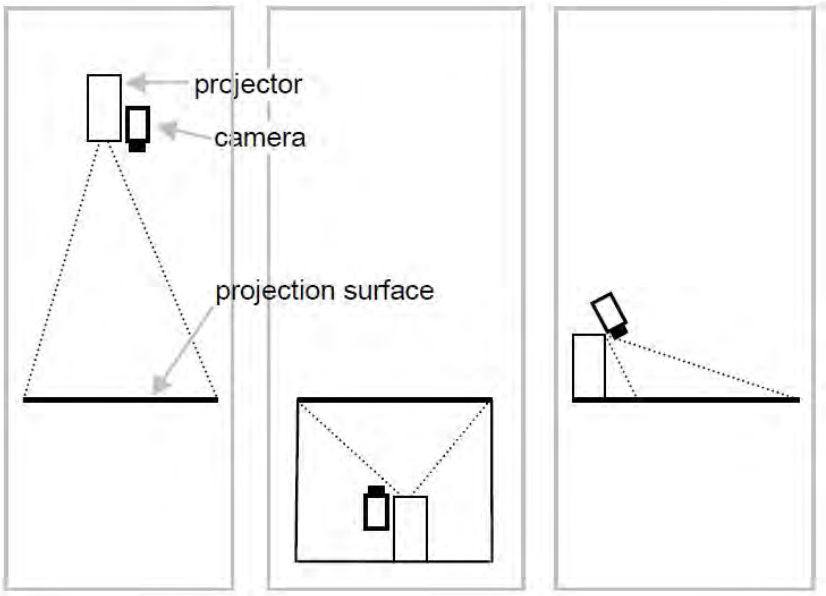


Figure 2.6: Projection-vision systems configurations: front projection with projector and camera mounted above (left), rear projection with projector and camera in a cabinet (middle), camera and projector sit off to the side of the active surface (right). [174]

There has been a great variety of interactive table and wall research prototype systems. Here we limit discussion to projection-vision based touch screens.

One popular approach is to mount a camera and projector high on a shelf or on the ceiling [171, 163, 21]. Such mounting configurations are typically necessary because of the throw requirements of projectors and the typical focal length of video cameras. Such a configuration has the following drawbacks:

- Ceiling installation of a heavy projector is difficult, dangerous, requires special mounting hardware. and is best left to professionals.
- Once the installation is complete, the system and the projection surface cannot be moved easily.

- Often minor vibrations present in buildings can create problems during operation and make it difficult to maintain calibration [171].
- The user's own head and hands can occlude the projected image as they interact with the system.

A second approach is to place the projector and camera behind a diffuse projection screen [112, 161]. While this enables the construction of a self-contained device, allows the placement of codes on the bottom of the objects, and eliminates occlusion problems, this approach also has drawbacks:

- It is difficult to construct such a table system with large display area which also allows users room enough to put their legs under the table surface.
- Because the camera is looking through a diffuse surface, the imaging resolution is limited. High resolution capture of documents, for example, is impossible.
- A dedicated surface is required, and the resulting housing for the projector and camera can be quite large. This presents manufacturing and distribution problem for a real product.

A third approach is to place camera and projector sitting off to the side of the active surface [174]. This configuration requires short-throw projector and geometrical distortion correction.

Three different projection-vision system configurations are illustrated diagrammatically in Fig. 2.6.

Finally, there are a number of systems which embed sensing electronics into the surface itself [38, 137]. These systems typically result in very fast and precise detection of touch compared to vision based approaches, but lack much of the flexibility in terms of other objects to be sensed. Others support only objects with

special embedded hardware devices and do not detect touch [128]. These systems usually rely on overhead projection.

Computer vision-based tables are capable of interesting sensing capabilities, including detection and recognition of objects place on the surface. In this paper we present novel techniques to enable a variety of sensing capabilities and interactions, many of these capabilities have been studied in previous work.

Robust finger tracking has been studied in the context of table systems [94, 98], but generally 'clicking' or 'pen down' is implemented by dwelling or other gesture recognition. True detection of touch can be detected roughly with two cameras [33, 108, 173]. In [174], the analysis of shadows is we explored to detect touch and infer hover height. A related formulation uses shadows to infer the height of objects above a surface but is unsuited to the case where the object is touching the surface and so occludes its own shadow, while another approach using observing shadows using an illuminant coaxial with the camera is unable to infer precise depth or hover information.

After the release of PrimeSense's [6] depth-sensing camera-based Microsoft Kinect [4], depth-sensing cameras have been used in various interactive surface applications. LightSpace [176] used an array of depth-sensing cameras to track users's manipulations on multiple surfaces. In [175], the touch event was determined by using a per-pixel depth threshold derived from a histogram of the static scene. Omnitouch [63] detected surface touch by counting the pixel number in a flood filling operation in depth map.

Besides the planar surface, some curved surfaces are projected for interaction purpose. Sphere [20] is a multi-user, multi-touch-sensitive spherical display in which an infrared camera used for touch sensing shares the same optical path with the projector used for the display. Instrumented with a single depth camera, a stereoscopic projector, and a curved screen, MirageTable [19] is an interactive

system designed to merge real and virtual worlds into a single spatially registered experience on top of a table. LightGuide [151] is a system that explores a new approach to gesture guidance where we project guidance hints directly on a user's body.

2.2 Natural Human-Computer Interaction

Human-Computer Interaction **(HCI)** involves the study, planning, and design of the interaction between people (users) and computers. It is often regarded as the intersection of computer science, behavioral sciences, design and several other fields of study. Interaction between users and computers occurs at the user interface (or simply interface), which includes both software and hardware; for example, characters or objects displayed by software on a personal computer's monitor, input received from users via hardware peripherals such as keyboards and mouses, and other user interactions with large-scale computerized systems such as aircraft and power plants. The Association for Computing Machinery (ACM) defines human-computer interaction [65] as "a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them."

Compared with traditional Human-Computer Interaction, the natural Human-Computer Interaction is (1) effectively invisible, or becomes invisible with successive learned interactions, to its users, and (2) is based on nature or natural elements. The word natural is used because most computer interfaces use artificial control devices whose operation has to be learned. A natural HCI relies on a user being able to quickly transition from novice to expert. Thus, "natural" refers to a goal in the user experience - that the interaction comes naturally, while interacting with the technology, and that the interface itself is natural.

If the users could interact with the computer through the approaches that used to communicate with other people, such as speech, body motion, head pose, facial expression, hand gesture etc., it will be the most natural way for users to manipulate the computers. On the side of computer, the computer should have the capability to comprehend what the user says, to recognise what the gesture the user is showing, to estimate what the use's head pose is, to understand the changes of user's facial expression, and to interpret the meaning of user's hand gesture indicates.

For all the natural channels mentioned above, many researches in the literature address each topic extensively. Here we mainly provide a literature review on head pose estimation and hand gesture recognition to show the relation between our research and the state of the art.

2.2.1 Head Pose

The capacity to estimate the head pose of another person is a common human ability that presents a unique challenge for computer vision systems. In a computer vision context [118], head pose estimation is the process of inferring the orientation of a human head from digital imagery. It requires a series of processing steps to transform a pixel-based representation of a head into a high-level concept of direction. Like other facial vision processing steps, an ideal head pose estimator must demonstrate invariance to a variety of image-changing factors. These factors include physical phenomena like camera distortion, projective geometry, multi-source non-Lambertian lighting, as well as biological appearance, facial expression, and the presence of accessories like glasses and hats.

Head pose estimation is most commonly interpreted as the ability to infer the orientation of a person's head relative to the view of a camera. It is often assumed that the human head can be modeled as a disembodied rigid object. Under

this assumption, the human head is limited to three DOF in pose, which can be characterized by *pitch*, *roll*, and *yaw* angles as pictured in Fig. 2.7.

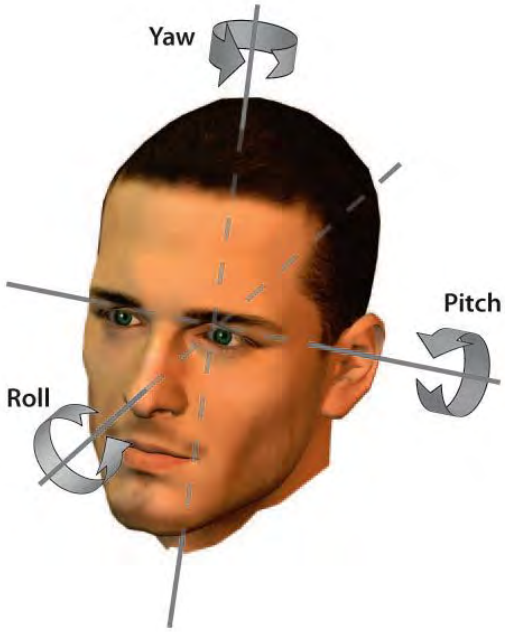


Figure 2.7: The three degrees of freedom of a human head can be described by the egocentric rotation angles *pitch*, *roll*, and *yaw* [118].

In literature, the head pose estimation methods can be divided into the following seven categories based on the fundamental approach that underlies the implementation:

Appearance Template Matching

Appearance template matching methods employ image-based comparison metrics to match a view of a person's head to a set of exemplars(each labeled with a discrete pose). And the queried image is given the same pose that is assigned to the most similar of these templates. D. Beymer [22] made use of normalized cross-correlation at multiple image resolutions as similarity measurement, while S. Niyogi and W. Freeman [122] applied mean squared error(MSE) over a sliding

window.

Compared with more complicated methods, appearance templates have some advantages. Firstly, the templates can be extended to a larger set at any time, allowing system to adapt to changing conditions. Moveover, appearance templates do not require negative training examples or facial feature points. Appearance templates are also well suited for both high and low-resolution imagery. However, there are many disadvantages with appearance templates. Without using some interpolation methods, they are only capable to estimate discrete pose orientations. Localization error can degrade the accuracy of the head pose estimation, since they assume that the head region has already been detected. As more templates are added into exemplar set, more computationally expensive image comparisons will need to be compute. Despite those limitations, the most significant problem with appearance templates is that they operate under the faulty assumption that pairwise similarity in the image space can be equated to similarity in pose.

Detector Array

Frontal face detection has been a hot topic in computer vision during last decade, and many approach [42, 60, 167] have been introduced. Given the success of these approaches, it seems like a natural extension to estimate head pose by training multiple face detectors, each to specific a different discrete pose. Detector array are similar to appearance templates in that they operate directly on an image patch. Instead of comparing an image to a large set of individual templates, the image is evaluated by a detector trained through many images with a supervised learning algorithm.

Huang et al. [75] used three SVMs to compose detector array for estimate three discrete yaws. Zhang et al. [185] trained five FloatBoost classifiers for head pose estimation in a far-field multi-camera setting.

Since each detector is capable of making the distinction between head and non-head, so an advantage of detector array methods is that a separate head detection and localization step is not required. Another improvement is that unlike appearance templates, detector arrays can ignore the appearance variation that does not correspond to pose change through training algorithm. Like appearance templates, detector arrays are also suitable for high and low-resolution images.

Disadvantages to detector arrays also exit. It is burdensome to train many detectors for each discrete pose, since besides many positive face examples, many negative non-face examples are also necessary for detector training, which require substantially more training data. Another disadvantage is the degree of freedom, in practice, the detector arrays approaches have been limited to one DOF and fewer than 12 detectors. Furthermore, since the majority of the detectors have binary outputs, there is no way to derive a reliable continuous estimate from the result, allowing only coarse head pose estimates. Finally, the computational requirements increase linearly with the number of detectors, making it difficult to implement a real-time system with large array.

Geometric Methods

Geometric approaches to head pose estimation use head shape and the precise configuration of local features to estimate pose, since these factors, such as the location of the face in relation to the contour of the head, strongly influence the human perception of head pose, suggesting that these are extremely salient cues regarding the orientation of the head.

Early approaches focused on estimating the head from a set of facial feature locations. It is assumed that these features are already known, and that pose can be estimated directly from the configuration of these points. The configuration of facial features can be exploited in many ways to estimate pose. A. Gee and R.

Cipolla [8] employed five facial points (the outside corners of each eye, the out corners of the mouth, and the tip of the nose), the facial symmetry axis is found by connecting a line between the midpoint of the eyes and the midpoint of mouth. Assuming a fixed ratio between these facial points and a fixed length of the nose, the facial direction can be determined under weak-perspective geometry from the 3D angle of the nose. T. Horprasert et al. [160] used a different set of five points (the inner and outer corners of each eye, and the tip of the nose) to estimate pose. Under the assumption that all four eye points are assumed to be coplanar, yaw can be determined from the observable difference in size between the left and right eye due to projective distortion from the known camera parameters. Roll can be found simply from the angle of this line from horizon. Pitch is determined by comparing the distance between the nose tip and the eye-line to an standard model. Wang and Sung [74] recently proposed another geometric method using the inner and outer corners of each eye and the corners of the mouth.

The most significant advantages of the geometric methods are fast and simple. With only a few facial features, a decent estimate of head pose can be derived. However, the obvious difficulty lies in detecting the features with high precision and accuracy, but the more subtle challenges stem from handling outlying or missing feature detections. Considering that geometric approaches depend on the accurate detection of facial points, they are typically more sensitive to occlusion than appearance-based methods that use information from the entire facial region.

Flexible Models

In flexible models approaches, a nonrigid model is fit to the image such that it conforms to the facial structure of each individual. In addition to pose labels, these methods require training data with annotated facial features, but it enables them to make comparisons at feature level rather than at the global appearance

level.

One of the flexible models for head pose estimation is Elastic Bunch Graph (EBG). To train this system, facial feature locations are manually labeled in each training image, and local feature descriptors such as Gabor jets can be extracted at each location, representing nonrigid, or deformable, objects. The technique called Elastic Graph Matching (EGM) is employed to compare a bunch graph to a new face image, the graph is placed over the image, and exhaustively or iteratively deformed to find the minimum distance between the features a every graph node location. In [87], a different bunch graph is created for every discrete pose, and each of these is compared to a new view of the head. The bunch graph with the maximum similarity assigns a discrete pose to the head.

Since EGM uses features located at specific facial points, there is significantly less inter-subject variability than with unaligned points, which makes it much more likely that similarity between the models will equate to similarity in pose. A disadvantage to this method is that pose estimate is discrete, requiring many bunch graphs to gain fine head pose estimates. Unfortunately, comparing many bunch graphs, each with many deformations, is computational expensive.

Another flexible models that have evolved for head pose estimation are the Active Shape Model (ASM) [156] and Active Appearance Model (AAM) [157], which learn the primary modes of variation in facial shape and texture from a 2D perspective. Given a rough initialization of face shape, the AAM can be fitted to a new facial image by iteratively comparing the rendered appearance model to the observed image and adjusting the model parameters to minimize a distance measure between these two images. Once the model has converged to the feature localizations, an estimate of head pose can be obtained by mapping the appearance parameters to a pose estimate. Cootes et al. [158] employed linear regression for yaw estimation by AAM. Baker et al. [141] proposed modified AAMs that expand

their utility to driver head pose estimation.

AAMs have good invariance to head localization error, since they adapt to the image and find the exact location of the facial features, allowing precise and accurate head pose estimation. The main limitation of AAMs is that all of the facial features are required to be located in each image frame. In practice, these approaches are limited to head pose orientations from which the outer corners of both eyes are visible. It is also not evident that AAM fitting algorithms could successfully operate for far-field head pose estimation with low-resolution facial images.

Nonlinear Regression Methods

Nonlinear regression methods estimate pose by learning a nonlinear functional mapping from the image space to one or more pose directions. The allure of these approaches is that with a set of labeled training data, a model can be built that will provide a discrete or continuous pose estimate for any new data sample.

Of the nonlinear regression tools used for head pose estimation, neural networks have been the most widely used in the literature [23, 187].

A locally linear map (LLM) is another popular neural network consisting of many linear maps [134]. To build the network, the input data is compared to a centroid sample for each map and used to learn a weight matrix. Head pose estimation requires a nearest-neighbor search for the closest centroid, followed by linear regression with the corresponding map. This approach can be extended with difference vectors and dimensionality reduction [71] as well as decomposition with Gabor wavelets [96].

The advantage of neural network approaches are numerous. These systems are very fast, only require cropped labeled faces for training, work well in near field and far-field imagery, and give some of the most accurate head pose estimation in

practice. The main disadvantage to these methods is that they are to prone to error from poor head localization.

Tracking Methods

Tracking methods operate by following the relative movement of the head between consecutive frames of a video sequence. Temporal continuity and smooth motion constraints are utilized to provide a visually appealing estimate of pose over time. These systems typically demonstrate a high level of accuracy, but initialization from a known head position is requisite. Typically, the subject must maintain a frontal pose before the system has begun, and must be re-initialized whenever the track is lost. As a result, approaches often rely on manual initialization or a camera view such that the subject's neutral head pose is forward-looking and easily re-initialized with a frontal face detector.

Tracking methods can operate in a bottom-up manner, following low-level facial landmarks from frame to frame [53, 183, 114, 124, 186]. Tracking can alternatively employ a model-based approach by finding the transformation of a model that best accounts for the observed movement of the head [107, 178].

The primary advantage of tracking approaches is their ability to track the head with high accuracy by discovering the small pose shifts between video frames. In this tracking configuration, these methods consistently outperform other head pose estimation approaches. An additional advantage with model-based tracking is the ability to dynamically construct an individualized archetype of a person's head. This allows these approaches to avoid the detrimental effects of appearance variation.

The difficulty with tracking methods is the requisite of an accurate initialization of position and pose to generate a new model or adapt an existing model. Without a separate localization and head pose estimation step, these approaches

can only be used to discover the relative transformation between frames. In this mode of operation, these methods are not estimating head pose in the absolute sense, but rather tracking the movement of the head.

Hybrid Methods

Hybrid approaches combine one or more of the aforementioned methods to estimate pose. These systems can be designed to overcome the limitations of any one specific head pose category. A common embodiment is to supplement a static head pose estimation approach with a tracking system. This method yields the high accuracy of pure tracking approaches without initialization and drift limitations. Many successful combinations have been presented by mixing an automatic geometric method with point tracking [89, 68, 121], PCA embedded template matching with a continuous density hidden Markov model [69], PCA embedded template keyframe matching with stereo tracking by grayscale and depth constancy [116], and color and texture appearance templates with image-based particle filtering [18].

Hybrid systems can also use two or more independent techniques and fuse the estimates from each system into a single result. In this case, the system gains information from multiple cues that together increase the estimation accuracy [177].

2.2.2 Hand Gesture

Most of the complete hand interactive systems can be considered to be composed by three layers [179]: detection (or segmentation), tracking and recognition. The detection layer is responsible for defining and extracting visual features that can be attributed to the presence of hands in the field of view of the camera(s). The

tracking layer is responsible for performing temporal data association between successive image frames. Moreover, in model-based methods, tracking also provides a way to maintain estimates of model parameters, variables and features that are not directly observable at a certain moment in time. Last, the recognition layer is responsible for grouping the spatiotemporal data extracted in the previous layers and assigning the resulting groups with labels associated to particular classes of gestures. In this subsection, related works in literature on these three subproblems are reviewed.

Detection and Segmentation

The primary step in gesture recognition systems is the detection of hands and the segmentation of the corresponding image regions. This segmentation is crucial because it isolates the task-relevant data from the image background, before passing them to the subsequent tracking and recognition stages. A large number of methods have been proposed in the literature that utilize a several types of visual features and, in many cases, their combination. Such features are skin color, shape, motion and anatomical models of hands. In [102], a comparative study on the performance of some hand segmentation techniques can be found.

- ***Color***

  Several color spaces have been proposed for hand detection including RGB, normalized RGB, HSV [149], YCrCb [27], YUV [14], etc. Several methods [91, 14, 150] utilize pre-computed color distributions extracted from statistical analysis of large datasets. Parametric models of the color distribution have also been used in the form of a single Gaussian distribution [72] or a mixture of Gaussians [143]. Generally, color segmentation can be confused by background objects that have a color distribution similar to human skin. A way to cope with this problem is based on background subtraction [52].

Moreover, skin color is only one of many cues to be used for to hand detection. For example, in cases where the faces also appear in the camera field of view, further processing is required to distinguish hands from faces [188]. Thus, skin color has been utilized in combination with other cues to obtain better performance.

- ***Shape***

  The characteristic shape of hands has been utilized to detect them in images in multiple ways. Much information can be obtained by just extracting the contours of objects in the image. In the 2D/3D drawing systems of [164], the user's hand is directly extracted as a contour by assuming a uniform background and performing real-time edge detection in this image. Local topological descriptors have been used to match a model with the edges in the image [142, 51, 83]. Certain methods focus on the specific morphology of hands and attempt to detect them based on characteristic hand shape features such as fingertips [14, 80].

- ***Trained detectors***

  Significant work has been carried out on finding hands in grey level images based on their appearance and texture. Several methods [181, 85, 84] attempt to detect hands based on hand appearances, by training classifiers on a set of image samples. More recently, methods based on a machine learning approach called boosting have demonstrated very robust results in face and hand detection [167, 41].

- ***Model-based detection***

  A category of approaches utilize 3D hand models for the detection of hands in images. One of the advantages of these methods is that they can achieve view-independent detection. Different models have been proposed, such as

point and line feature model [80, 182], edge feature model [79] and rigid joints model [77]. The more complicated 3D models should have enough degrees of freedom to adapt to the dimensions of the hand(s) present in an image [17, 97, 52].

- ***Motion***

  Motion is a cue utilized by a few approaches to hand detection. The reason is that motion-based hand detection demands for a very controlled setup, since it assumes that the only motion in the image is due to hand movement [78, 130].

Tracking

Tracking, or the frame-to-frame correspondence of the segmented hand regions or features, is the second step in the process towards understanding the observed hand movements. The importance of robust tracking is twofold. First, it provides the inter-frame linking of hand/finger appearances, giving rise to trajectories of features in time. These trajectories convey essential information regarding the gesture and might be used either in a raw form (e.g. in certain control applications like virtual drawing the tracked hand trajectory directly guides the drawing operation) or after further analysis (e.g. recognition of a certain type of hand gesture). Second, in model-based methods, tracking also provides a way to maintain estimates of model parameters variables and features that are not directly observable at a certain moment in time.

- ***Template based tracking***

  This class of methods exhibits great similarity to methods for hand detection. Members of this class invoke the hand detector at the spatial vicinity that the hand was detected in the previous frame, so as to drastically restrict

the image search space. The implicit assumption for this method to succeed is that images are acquired frequently enough [159, 50, 78, 156].

- ***Tracking based on the Mean Shift***

  The Mean Shift algorithm [180] is an iterative procedure that detects local maxima of a density function by shifting a kernel towards the average of data points in its neighborhood. The algorithm is significantly faster than exhaustive search, but requires appropriate initialization. The work in [36, 37, 57] is not restricted to hand tracking, but can be used to track any moving object. Mean-Shift tracking is robust and versatile for a modest computational cost. It is well suited for tracking tasks where the spatial structure of the tracked objects exhibits such a great variability that trackers based on a space-dependent appearance reference would break down very fast. On the other hand, highly cluttered background and occlusions may distract the mean-shift trackers from the object of interest.

- ***Particle filtering***

  Particle filters have been utilized to track the position of hands and the configuration of fingers in dense visual clutter [104, 76, 77]. In this approach, the belief of the system regarding the location of a hand is modeled with a set of particles. The approach exhibits advantages over Kalman filtering, because it is not limited by the unimodal nature of Gaussian densities that cannot represent simultaneous alternative hypotheses. A disadvantage of particle filters is that for complex models (such as the human hand) many particles are required, a fact which makes the problem intractable especially for high-dimensional models.

Recognition

The overall goal of hand gesture recognition is the interpretation of the semantics that the hand(s) location, posture, or gesture conveys. Basically, there have been two types of interaction in which hands are employed in the user's communication with a computer. The first is control applications such as drawing, where the user sketches a curve while the computer renders this curve on a 2D. Methods that relate to hand-driven control focus on the detection and tracking of some feature (e.g. the fingertip, the centroid of the hand in the image etc) and can be handled with the information extracted through the tracking of these features. The second type of interaction involves the recognition of hand postures, or signs, and gestures.

- ***Template matching***

  Template matching, a fundamental pattern recognition technique, has been utilized in the context of both posture and gesture recognition. For the recognition of hand postures, the image of a detected hand forms the candidate image which is directly compared with prototype images of hand postures [7, 49, 103]. The best matching prototype (if any) is considered as the matching posture. Clearly, because of the pixel-by-pixel image comparison, template matching is not invariant to scaling and rotation.

- ***Contour and silhouette matching***

  This class of methods [17, 17, 165] mainly refers to posture recognition and is conceptually related to template matching in that it compares prototype images with the hand image that was acquired to obtain a match. The defined feature space is the edges of the above images. The use of silhouettes [16] in gesture recognition has not been extensive, probably because different hand poses can give rise to the same or similar silhouette. Another

reason is that silhouette matching requires alignment (or else, point-to-point correspondence establishment across the total arc length), which is not always a trivial task. Also, matching of silhouettes using their conventional arc length descriptions (or ”signatures”) is very sensitive to deformations and noise.

- ***Model-based recognition methods***

  Most of the model-based gesture recognition approaches [181, 11, 133, 73] employ successive approximation methods for the estimation of their parameters. Since gesture recognition is required to be invariant of relative rotation, intrinsic parameters such as joint angles are widely utilized. The strategy of most methods in this category is to estimate the model parameters, e.g. by inference or optimization, so that the extracted features match a model.

- ***HMMs***

  HMM is a rich tool used for hand gesture recognition in diverse application domains. Probably, the first publication addressing the problem of hand gesture recognition is the celebrated paper by Yamato et al. [88]. In this approach, a discrete HMM and a sequence of vector-quantized (VQ)-labels have been used to recognize six classes of tennis strokes. Before applying the HMM, the image sequence goes through several preprocessing steps such as low-pass filtering to reduce the noise, background subtraction to extract the moving objects, and binarization of the moving objects in order to generate blobs. The blobs roughly represent the poses of the human. The features are the amounts of object (black) pixels. These features are vector quantized, such that the image sequence becomes a sequence of VQ-labels, which are then processed by a discrete HMM. Subsequently, several other applications of hand gesture recognition have been developed based on H-

MMs, such as sign language recognition [86, 129], graphic editor control [58], and robot control [10].

Chapter 3

Head Pose Estimation by Imperceptible Structured Light Sensing

In this chapter, we describe a method of estimating head pose estimation from imperceptible structured light sensing. First, through elaborate pattern projection strategy and camera-projector synchronization, pattern-illuminated images and the corresponding scene-texture image are captured with imperceptible patterned illumination. Then, 3D positions of the key facial feature points are derived by a combination of the 2D facial feature points in the scene-texture image localized by AAM and the point cloud generated by structured light sensing. Eventually, the head orientation and translation are estimated by SVD of a correlation matrix that is generated from the 3D corresponding feature point pairs over different frames. Extensive experiments show that the proposed method is effective, accurate and

rapid in 6-DOF head pose estimation, making it suitable for real-time application.

3.1 Introduction

Head pose estimation has always been an active topic in computer vision for its usefulness in a variety of applications. In human-computer interaction, head pose is an important cue for computer or robot to infer the intention of human [111]. For some face-related applications such as face alignment, face recognition, and facial expression recognition, estimating the pose of the face is considered as a precondition or preprocessing step [24]. Moreover, for driver-assistance systems, head pose estimation is essential for inferring the driver's focus of attention [119].

In the context of computer vision, head pose estimation is most commonly interpreted as the ability to infer the orientation and translation of a person's head with respect to the view of a camera. And the human head is assumed to be modeled as a disembodied rigid object, thus the human head motion is limited to six degrees of freedom (DOF), three for orientation characterized by pitch, roll, and yaw, and three for translation along three directions.

The adoption of structured light illumination has been proven to be an effective and accurate visual means for 3D reconstruction [147]. The system consists of a projector that projects controlled patterns to the target object, and a camera capturing images of the illuminated object. Once correspondences between positions on the projector's pattern panel and positions on the camera's image plane are established through the use of some elaborately designed coding strategies on the illuminated patterns, simple triangulation over the light rays from the projector and the corresponding light rays to the camera would recover 3D information about the target object. Recently, the availability of pico projectors with average dimensions of $4 \times 2 \times 1$ inches has widely extended the application area of

structured light system. Nikon Corp. has even integrated an ultra-small built-in projector into its latest digital camera COOLPIX S1000pj, making it possible to implement structured light system in hand-held consumer electronic products.

However, light projection's intrinsic characteristics could lead to disadvantages in specific circumstances: it could yield to loss or corruption of colorimetrical and textual information of the lighted surfaces, to the inconsistence of the optical flow, and moreover, to the offensive, indeed dangerous aspects of the illumination (think about the potential danger of LASER source; even video projection to human face for face measurement etc. could arouse discomfort).

In order to benefit from the merits of structured light vision while avoiding the drawbacks, some researchers designed structured light system in the non-visible spectrum [48]. Three major approaches are InfraRed Structured Light (IRSL), Filtered Structured Light (FSL), and Imperceptible Structured Light (ISL). ISL is easy to implement, since it requires similar equipments as those of regular projection: a digital projector, and cameras. The light source projects a light pattern followed by its complement (inverse pattern) onto the scene at high frequency, so that the resulting pattern is perceived by humans as uniform. The first camera is synchronized with the projection of the first pattern to get 3D information of the scene, just like in the traditional structured light methods; the second one has long integration time and observes the scene under uniform light to capture a gray-level or colored image representing scene texture.

This chapter describes a method of determining the 6-DOF head pose by the use of an imperceptible structured light system. The method is able to track accurate 3D positions of salient facial landmarks without the need of going through any training process. Firstly, through elaborate pattern projection strategy and camera-projector synchronization, a pattern-illuminated image and the corresponding scene-texture image are captured under illumination that appears as white light yet em-

beds coded patterns. Then, in the point cloud generated by structured light sensing, the facial feature points in the scene-texture image localized by AAM will have their 3D positions interpolated. Correspondences between such facial features in 3D, with those associated with the previous or reference image frame, can then be constructed. Finally, the head orientation and translation are estimated by SVD of a correlation matrix that is generated from such point pairs in 3D.

The remainder of this chapter is structured as follows: In Section 3.2, related works on head pose estimation and imperceptible structured light sensing are briefly reviewed. The essential processes of the proposed method including pattern projection strategy, facial landmark localization, and 6-DOF head pose estimation are described in Section 3.3. In Section 3.4, system setup and experimental results are shown. Conclusion and possible future work are offered in Section 3.5.

3.2 Previous Works

Head pose estimation has been extensively researched and increasingly used in human-computer interface and driver safety assistant system. Imperceptible structured light system (ILS) is an effective and low cost means of measuring 3D information, that is without disrupting, modifying, or putting in danger the environment. Below we offer a brief review of some of the key works related to head pose estimation and ILS.

3.2.1 Head Pose Estimation

Since there are many potential applications of head pose estimation, a variety of approaches to the problem have been proposed in the past decade. A comprehensive literature review has been recently carried out by Murphy-Chutorian and

Trivedi [118]. Below we outline some key works related to our work.

Since the head pose or motion is in 3D domain, inherently using 3D information and 3D face models are more direct and accurate for pose estimation, than using 2D texture information (such as points, edges etc.) in images. Morency et al. use depth and intensity view-based eigenspaces to build a prior model from the first frame that is then robustly tracked [116]. Jimenez et el. build a 3D face model using some points chosen by SMAT in the first frame [90]. Through the stereo correspondence of the two cameras, the 3D coordinates of these points are extracted, and the points are tracked in the following frames and 3D pose are calculated at each frame by RANSAC and POSIT.

Some methods have been presented that work on range-scan data. Based on a novel shape signature to identify noses in range images, Breitenstein et al. generate candidates for the nose positions, and then generate and evaluate many pose hypotheses [26]. The pose is estimated using an error function that is employed to compare the input range image with the pre-computed pose image of an average face model.

In the above methods, though 3D acquisition systems are there to provide accurate and dense data, the vast amount of data also demands the use of powerful parallel processors (GPU), or else there could be difficulty in processing the data in real time.

Besides, hybrid methods that combine one or more methods have shown good performance in pose estimation. Murphy-Chutorian et al. present a system based on localized gradient orientation histograms, that are integrated with support vector machines for regression [119]. However, some training processes on some previously prepared training sets are needed for the learning based method, which are generally tedious and time consuming.

In contrast, our approach derives the 3D positions of key facial feature points

in a sparse point cloud generated from ISL system, requiring no training process. The low computational complexity also ensures real-time performance.

3.2.2 Imperceptible Structured Light

A first proof of concept for embedding invisible structured light patterns into DLP projections was introduced in the "Office of the Future" project [136]. In this work, binary codes are embedded by projecting temporally alternating code images and their complements. Provided that the frequency of projection reaches the *flicker fusion threshold* ($\geq 75Hz$), the pattern and the inverse pattern are visually integrated over time in human perception, and the illumination has the appearance of a flat field ("white" light") to humans. However, the concept of embedding structured light into DLP projections was achieved with significant modification effort on the projection hardware and firmware, including removal of the color wheel and reprogramming of the controller. The resulting images were also in greyscale only. The implementation of such a setting was impossible without mastering and full access to the projection hardware.

Cotting et. al. introduced a coding scheme [34] that synchronizes a camera to a specific time slot of a DLP micro-mirror flipping sequence in which imperceptible binary patterns are embedded. However, not all mirror states are available for all possible intensities, and the additional hardware, DVI repeater with tapped vertical sync signal, is not an off-the-shelf instrument.

However, with the development of digital projection technology, some so-called 3D compatible DLP projectors with fresh rate of $120Hz$ or higher emerged recently. This makes it possible to implement imperceptible structured light without any hardware modification or extra assisting hardware.

3.3 Method

Our ILS has a capability to capture the pattern-illuminated image and scene-texture image simultaneously by specially designed pattern projection strategy. Below, we first outline the design of the pattern projection strategy for imperceptible structured light (ISL) sensing. Then, the method for deriving 3D positions of the facial landmarks will be described. Finally, the method of head pose estimation that makes use of the corresponding point sets derived for different image frames will be given.

3.3.1 Pattern Projection Strategy for Imperceptible Structured Light Sensing

The vital aspect of imperceptible structured light sensing is the synchronization between camera's image capture and projector's illumination projection. Here we take one capture-projection cycle as an example to explain the strategy of pattern projection, which is illustrated in Fig. 3.1. In order to achieve imperceptible structured light projection, the frequency of projection must exceed the flicker fusion threshold, which is $75Hz$ for most of the people. First of all, we ensure that the projector projects an image every $10ms$, i.e., at $100Hz$. As shown in Fig. 3.1, along the time axis, the colored pattern illumination, the inverse colored pattern illumination, and entirely white illumination are projected at the time instants $0ms$, $10ms$, $20ms$ respectively. The former two images are projected for ISL sensing, while the latter one is projected for capturing the scene-texture image at almost the same time instant. On the camera side, the camera captures the pattern-illuminated image at $5ms$. With a refresh rate of the camera at about 30 frames per second (which is similar to that of most of the CCD cameras), the camera captures

the scene-texture image at $40ms$, shortly after the projector projects the entirely white illumination on the object. At $50ms$ a new capture-projection cycle will resume. With the aforementioned capture-projection strategy, the system could capture 20 image pairs (pattern-illuminated image and scene-texture image) per second.

The colored pattern illumination in our system is designed after the principle of pseudorandom array [153]. The grid points, which are the intersection of neighboring rhombic pattern elements, are chosen as feature points. We employed an encoding mechanism described in [153] to assure the uniqueness of each grid point. The 2D pseudorandom color pattern of 65×63 elements that have red, green, blue, or black colors for the pattern elements (the foreground), and white color for the background, together with the pattern's inverse, are shown in Fig. 3.2. The sum of the two images is an entirely white image, i.e., for every pixel, $RGB[I(x, y)] + RGB[INV(x, y)] = (255, 255, 255)$. To human's sensing the pattern and the inverse pattern are visually integrated over time. Over time the illumination appears like fluorescent light to humans.

Next, the 3D positions of the key facial landmarks are located by a combined use of the the pattern-illuminated image and the scene-texture image.

3.3.2 Facial Feature Localization

An image pair composed of a pattern-illuminated image and the corresponding scene-texture image will be derived in each projection-capture cycle. From the pattern-illuminated image, the 3D positions of the grid points can be determined from the inter-geometry of the projector and camera and the intrinsic parameters of the two instruments, through triangulation; from the scene-texture image, some salient facial landmarks can be located with ease. How to locate the 3D positions of the facial features from the two modalities is described below.

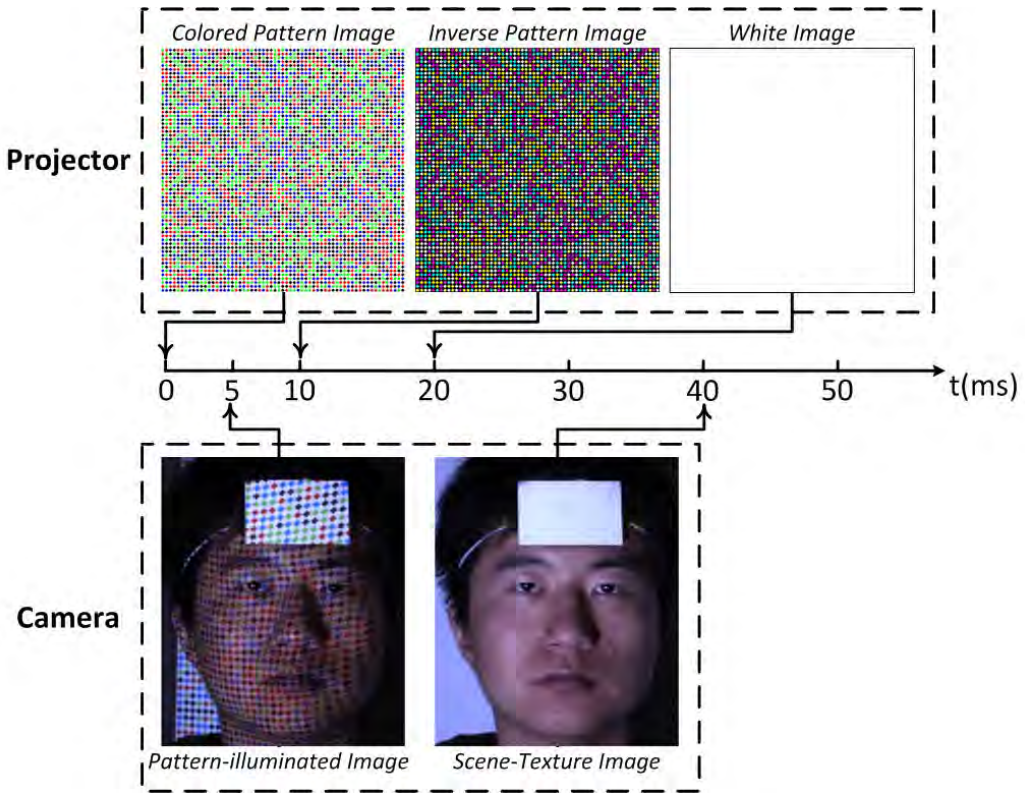
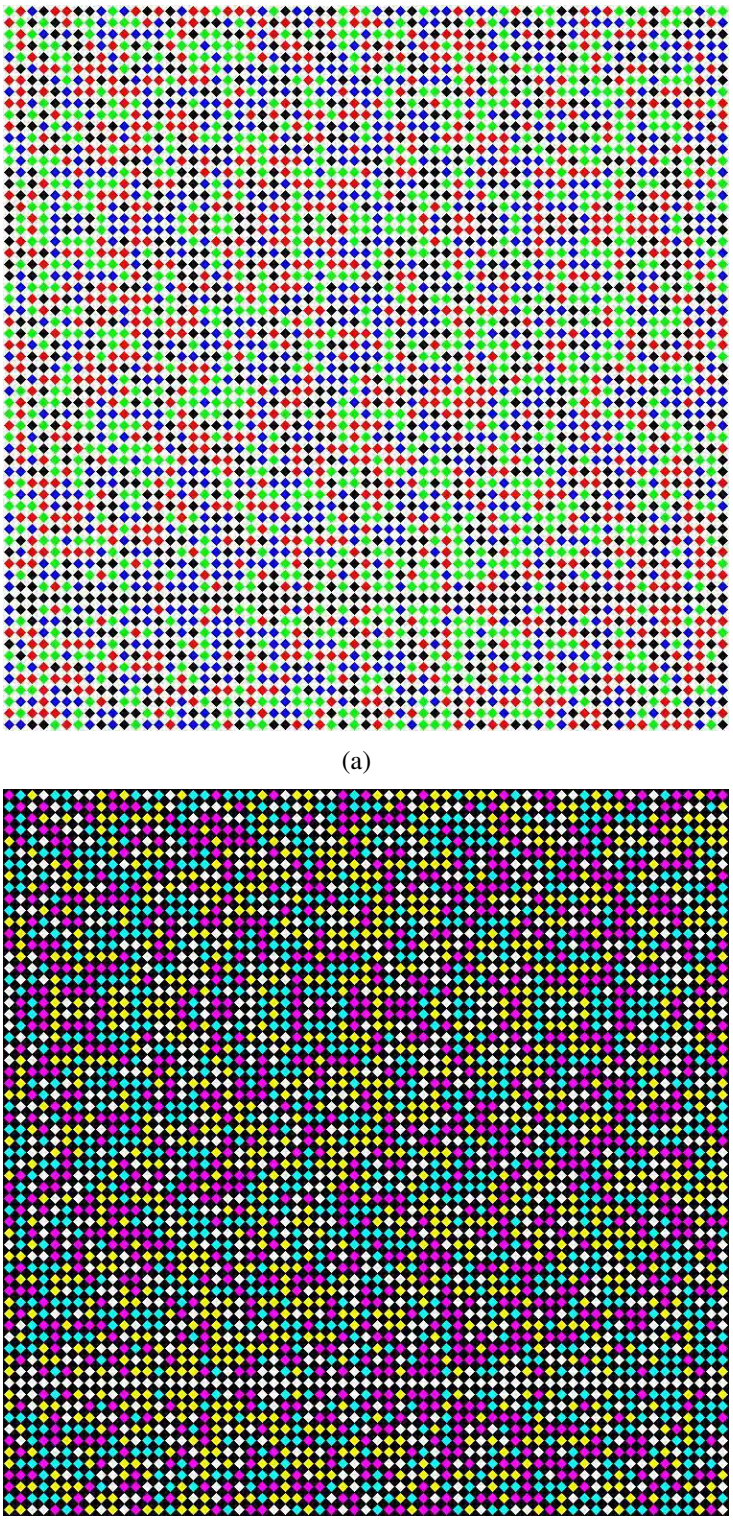


Figure 3.1: Capture-Projection Synchronization Strategy.



(a)

(b)

Figure 3.2: Pattern-illuminated images: (a) image under the original illumination; (b) image under the inverse illumination.

Localizing 2D Positions of Key Facial Feature Points in Scene-texture Image

Automatic face detection and facial feature localization in 2D image has been an actively researched subject for several years, and many effective methods have been proposed in the literature.

For the sake of accuracy and efficiency of 2D facial feature localization in scene-texture image, firstly, we employ the Adaboost [168] face detection method to extract the position of the face in the image. We then apply the AAM [113] method to localize the facial features in the segmented face image.

In Fig. 3.3, 25 feature points were defined from AAM localization. They lie on or around the salient features in the face, such as the inner corner and outer corner of the eyes, the corner of the eyebrows, the tip of the nose, and the corner of the mouth etc., which are relatively less affected by expression variation. In addition, all the feature points were distributed symmetrically in the frontal face, allowing at least half of them to be located accurately when there are extreme pose variations.

Determining 3D Positions of Grid Points in Pattern-illuminated Image

How unique code can be attributed to each position of the illuminated pattern is an essential question in SLS. On this, there are the temporal and spatial coding schemes. The spatial coding scheme has the advantage that 3D determination can be achieved with a single illumination and a single image capture. It is therefore particularly suitable for use in dynamic applications, such as the head pose estimation in this case. In our work, we employed the color coding scheme described in [153] to determine the 3D position of grid points in the pattern-illuminated image. In the illuminated pattern, each grid point is encoded by the color profile of the 2×3 rhombic elements surrounding it, and the code is preserved in the image data. Each of such grid points, once its position in the pattern-illuminated im-

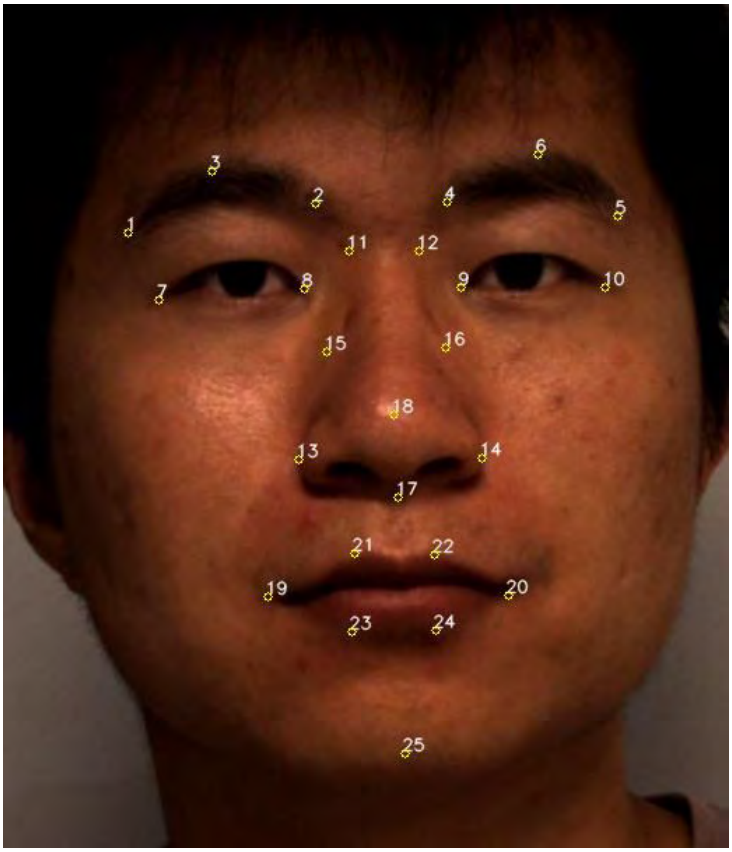


Figure 3.3: 2D facial features located by AAM.

age is located, can thus have its corresponding position in the illuminated pattern (on the projector side) identified from the unique code. With the knowledge of inter-geometry of the projector and camera and the intrinsic parameters of the two instruments acquired from an off-line calibration process, the 3D position of the grid point could be calculated by a simple triangulation step.

Inferring 3D Positions of Key Facial Features

Since the interval between the capture of the pattern-illuminated image and the scene-texture image is rather small (relative to the motion of the head), in this work we make the simplifying assumption that the head position is constant in the two images. With that, the grid point positions and the salient facial features in 3D can be related through the rigidity of the human face. More precisely, we infer the facial features from a combined use of the facial features' positions in the scene-texture image, the grid points' positions in the pattern-illuminated image, and the grid points' 3D positions estimated from the structured light sensing step. For each feature point in the scene-texture image, a mirror point could be found in the pattern-illuminated image, as illustrated in Fig. 3.4(a). It would be most desirable that the mirror point coincides with one of the grid points, as that way the 3D position of the feature point can be read as the depth of the grid point determined from structured light sensing. However, in practical condition, the coincidence would hardly occur, and the 3D positions of the facial feature points would need be interpolated from the 3D positions of the nearby grid points.

Consider a facial feature point and the image patch around it, which is illustrated by the yellow rectangle in Fig. 3.4(a). The window is magnified and shown in Fig. 3.4(b). Set an $n \times n$ window centered in the mirror point M. Assume that in this window, there are N grid points, denoted as $G_i, i = 1, \ldots, N$. Suppose the 3D position of G_i is X_i. Then the 3D position $\overline{X}$ of the feature point could be

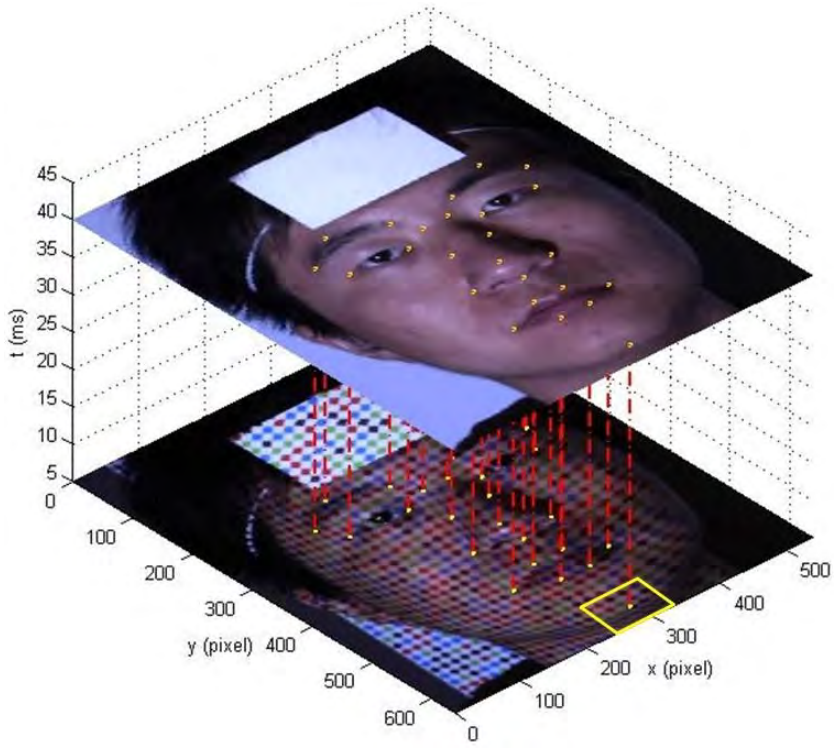
interpolated as the weighted average of the 3D positions of the nearby grid points in the selected window, which could be formulated by Eq. 3.1, where α_i is the weight, and d_i in Eq. 3.2 is the 2D Euclidean distance between the i-th grid point G_i and the mirror point M. For computational efficiency, here we only need to make use of 2D positions of the feature point and the nearby grid points, and in the structured light sensing step determine the 3D positions of not all grid points but only those that are in the immediate neighborhood of some key facial feature points. Despite that there should be certain discrepancy between the interpolated depth and the real depth of each facial feature point, the pose estimation algorithm described in the following sub-section could embrace such discrepancies and determine the head pose with the minimum influence.

$$\overline{X} = \sum_{i=1}^{N} \alpha_i X_i, \tag{3.1}$$

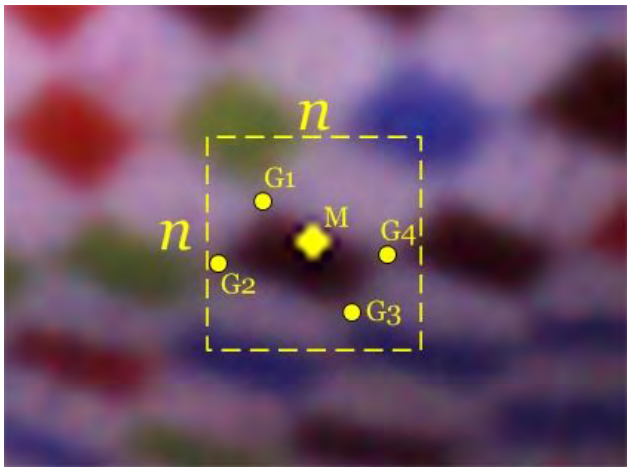
$$\alpha_i = \frac{d_i}{\sum_{j=1}^{N} d_j}. \tag{3.2}$$

3.3.3 6 DOF Head Pose Estimation

By the aforementioned method, the 3D positions of the predefined feature points could be determined in each frame. As a result, the correspondence between two sets of 3D points, each set from a consecutive image frame, can be established. Just like other computer vision tasks, notably those that require the estimation of the motion of a rigid object from 3D point correspondences, here we encounter the following mathematical problem. We have two 3D point sets $\{p_i\}$ and $\{p'_i\}, i = 1, 2, \ldots, N$(here, p_i and p'_i are considered as 3×1 column matrices), from which we need to determine the 3D rigid displacement (3×3 rotation matrix R, and



(a)



(b)

Figure 3.4: 3D facial feature landmarking by interpolation: (a) Feature points in the scene-texture image and the corresponding mirror points in the pattern-illuminated image. (b) One mirror point and its neighboring grid points in an $n \times n$ window.

3×1 translation vector T) between them:

$$p'_i = Rp_i + T + N_i, \tag{3.3}$$

where N_i is a noise vector. We want to estimate R and T to minimize

$$\Sigma^2 = \sum_{i=1}^{N} \|p'_i - (Rp_i + T)\|^2. \tag{3.4}$$

This problem is known as the *absolute orientation problem*, and there are a number of methods in the literature available to solve this problem. The solution methods can be categorized into two classes: iterative form, and closed form [43]. Closed form solutions are generally superior to iterative methods in terms of efficiency and robustness, because the latter suffers from the problems of not guaranteeing convergence, becoming trapped in local minima of the error function, and requiring good starting estimate. For these reasons, we chose the closed form solution to solve this problem. With comprehensive consideration of accuracy, robustness, stability, and efficiency of a number of methods, we employed the method proposed by Umeyama [162], which is based on computing the singular value decomposition (SVD) of a correlation matrix defined by:

$$H = \sum_{i=1}^{N} p'_{c_i} {p_{c_i}}^T = U\Lambda V^T, \tag{3.5}$$

where $p_{c_i} = p_i - \bar{p}$, $p'_{c_i} = p'_i - \bar{p}'$, $\bar{p} = \frac{1}{N}\sum_{i=1}^{N} p_i$, $\bar{p}' = \frac{1}{N}\sum_{i=1}^{N} p'_i$.

Then the optimal rotation matrix and translation vector could be calculated as

$$\hat{R} = UV^T, \tag{3.6}$$

$$\hat{T} = \bar{p}' - \hat{R}\bar{p}. \tag{3.7}$$

As long as more than three non-colinear corresponding point pairs are given, the method can determine the transformation parameters uniquely.

3.4 Experiments

3.4.1 Overview of Experiment Setup

In order to assess the feasibility of the proposed head pose estimation method using imperceptible structured light sensing, we conducted an accuracy evaluation experiment.

The projector-camera system we used in our experiment consisted of a DLP projector with a native resolution of 1024×768 and a refresh rate of $120Hz$ (Mitsubishi EX240U projector), and a camera of 1288×964 resolution at $30fps$ (Point Grey FL2G-13S2C-C CCD camera with Myutron FV1520 f15mm lens), both being off-the-shelf equipments. The focal length of the camera was fixed in $15mm$, while that of the projector was in the range of $25 - 31mm$. The ILS was configured for a working distance (the distance from the camera to the mean position of the human face) of about $800mm$.

We first fixed the camera and projector rigidly as shown in Fig. 3.5, and the projector and camera were connected to a desktop computer through VGA and IEEE-1394b interfaces respectively. Then the projector-camera system was calibrated using an LCD monitor as the calibration object; the calibration method,

detailed in [154], can derive the intrinsic and extrinsic parameters of the two instruments. Once the experimental system was set up, we could collect data for further experiments.

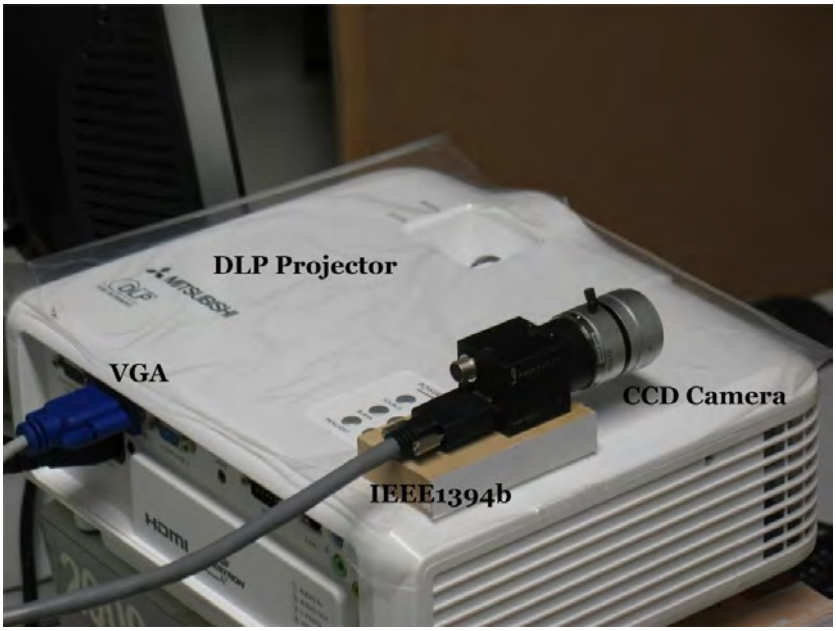


Figure 3.5: Prototype of ISL system.

3.4.2 Test Dataset Collection

Because of the differences in the various sensing methods used (such as monocular vision, stereo vision, infrared vision etc.), there is no standard benchmark for evaluating the performance of head pose estimation, and the researchers generally tested their algorithms on the databases collected by themselves. Through reviewing the literature, we found that the subjects in their databases range from one to less than 10, and for every subject, the video length is about several minutes. Because of the speciality of the proposed special sensing method, we have to collect experimental data by ourselves. The volume of our database is 15 persons, of which nine are male and six female, and 6 of them wearing glasses. The length of

each video sequence is about 1 minute, i.e., 1200 frames. The sequences started with the objects facing head-on to the cameras. Several sequences were recorded for each participant. The sequences were collected in the laboratory environment with some global illumination changes.

Performance assessment requires ground-truth of the orientation of the head in each frame. It is not possible and practical to acquire the ground truth manually. In order to make the ground truth more accessible, we asked the subjects to wear a headband to which a credit card sized white planar board has been attached, as seen in Fig. 3.7. The white board was adjusted to be parallel with the face, implying that the orientation of the face was equal to that of the white board. With color coded illumination, the 3D position of any three non-collinear grid points (named by P_1, P_2 and P_3) on the white board could be derived by the aforementioned approach, as shown in Fig. 3.6. Let $\mathbf{X_i}$ be the 3D positions of $P_i, i = 1, 2, 3$, the surface normal of the white board could be formulated as

$$\mathbf{n} = \frac{(\mathbf{X_1} - \mathbf{X_2}) \times (\mathbf{X_3} - \mathbf{X_2})}{\|(\mathbf{X_1} - \mathbf{X_2}) \times (\mathbf{X_3} - \mathbf{X_2})\|}. \tag{3.8}$$

Eventually, the accurate ground-truth of face orientation was made directly accessible. As for the position of the head, it could be interpreted through the centroid of the white board.

3.4.3 Results

Experimental results at some frames of a subject are illustrated in Fig. 3.7. In each sub-figure the AAM located feature points are indicated by yellow circles in the corresponding scene-texture image. The inset image in the bottom-right corner of each sub-figure shows the corresponding pattern-illuminated image, while the

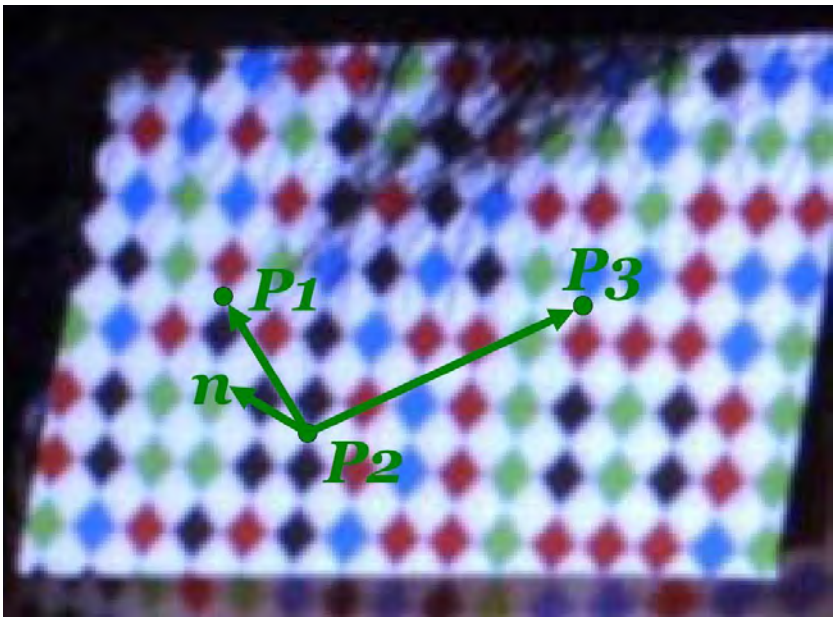


Figure 3.6: Ground truth on the surface orientation of human face: it was made the same as that of a white board attached to the face, and the latter could be computed directly for each image.

inset image in the top-right represents a qualitative description of the estimated head pose, in which the ground-truth and the estimated head pose are implied by a blue circle (or ellipse) and a red arrow respectively.

At the first frame, the subject was required to have his face in head-on orientation with respect to the camera, so that the orientation vector of the face was parallel with the optical axis of the camera. This is shown in the top-left sub-figure of Fig. 3.7. The 3D positions of all 25 facial feature points were derived for the first frame and the following frames, allowing the head poses relative to to the camera to be estimated on the basis of the corresponding 3D point pairs.

The mean absolute estimation error of the proposed method, along with those of three related approaches, are shown in Table 3.1. The comparison should be considered as a reference only, since the evaluation data-sets and the systems used to obtain the ground-truth are all different.

It should be noticed that the mean absolute error of yaw in the proposed method was generally larger than those of pitch and roll. We believe the reason for it lies in the asymmetric inaccuracy in localizing the 2D feature points by the AAM method, due to the illumination shadow around the eyes and nose caused at extreme pose variations.

Table 3.1: Comparison of pose estimation errors.

Method	Sensing	Mean Absolute Error (°)		
		Yaw	Pitch	Roll
Murphy-Chutorian [119]	Monocular	3.39	4.67	2.38
Morency [116]	Stereo	3.50	2.40	2.60
Jimene [90]	Stereo	1.85	1.61	1.20
Our method	ILS	2.02	1.18	0.76

For real-time applications, efficiency is of great importance, hence we implemented the proposed method in C++ using the Intel OpenCV Library to evaluate its processing time. Through multi-thread programming, the projection-capture process and calculation process were executed in two different threads respectively, each of which was able to run in real time in a desktop with Intel Pentium D 3.0GHz CPU. Table 3.2 shows the average processing times for AAM facial features localization, 3D depth calculation, and head pose estimation, for the given system. Facial features localization with AAM is the most time-consuming process. Processing times vary slightly according to the number of iterations in the AAM algorithm. However, they all satisfy the requirement of real-time application.

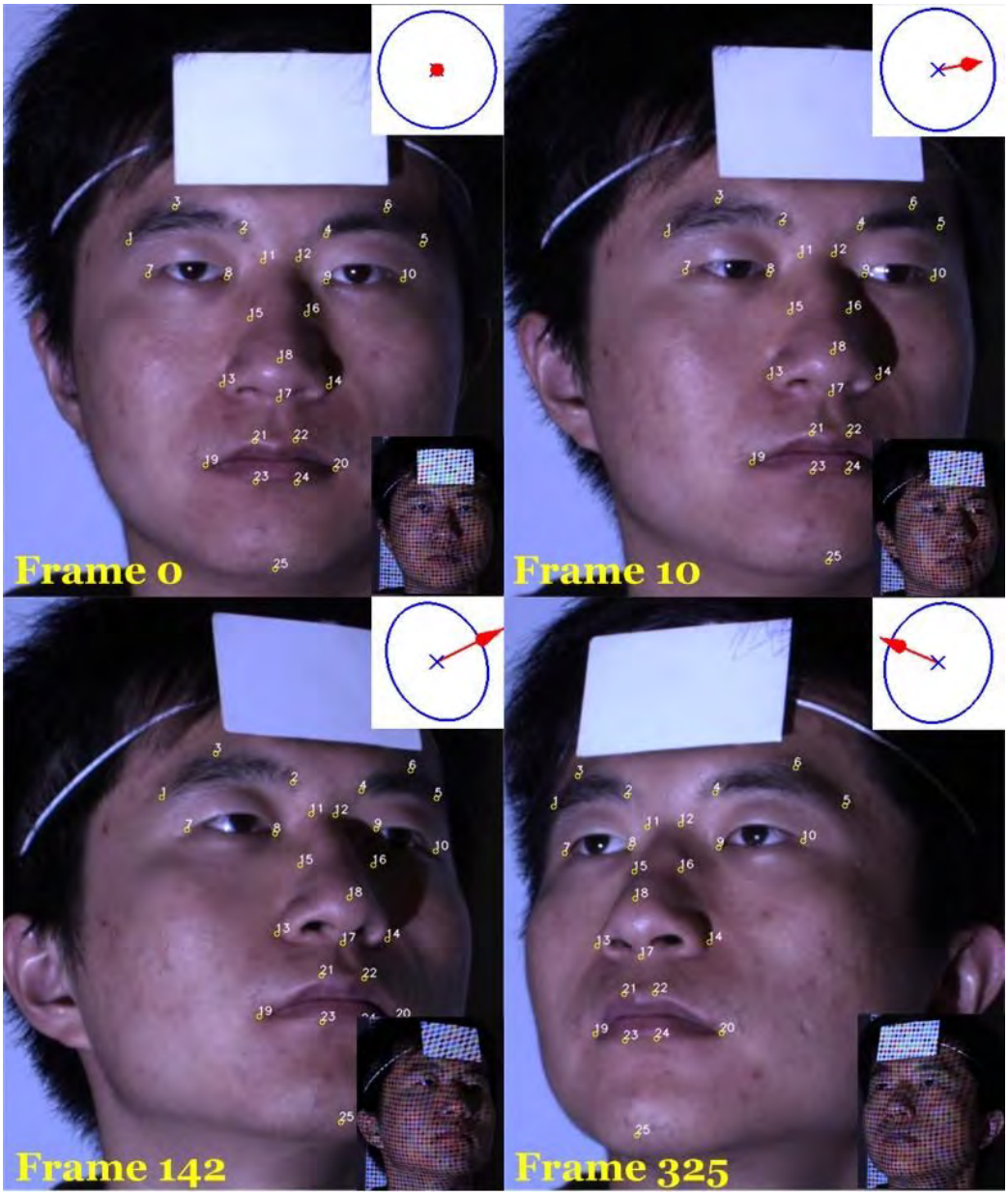


Figure 3.7: Experimental results.

Table 3.2: Average processing time.

Subroutine	AAM	3D Depth Calc.	Pose Est.	Total
Time (ms)	17.43	1.82	2.56	21.81

3.5 Summary

In this chapter, we proposed a novel method of estimating head pose using imperceptible structured light sensing. Through elaborate pattern projection strategy and camera-projector synchronization, pattern-illuminated images and the corresponding scene-texture images can be captured under imperceptible patterned illumination. Then, the 3D positions of the facial feature points are determined by putting together the 2D locations of the facial feature points in the scene-texture image (that are localized by AAM), and the point cloud generated by structured light sensing. Finally, the 6-DOF head motion is estimated from the 3D corresponding feature point pairs over different frames through SVD of a correlation matrix.

The proposed method has been tested on video sequences captured by a prototype of the described imperceptible structured light sensing system. Experimental results show that the proposed method is effective, accurate, and rapid for 6-DOF head pose estimation. The processing time is also fast enough for real-time application.

However, there are two assumptions made in our method. One is that the position of the human head is constant between the captures of the pattern-illuminated image and the subsequent scene-texture image. The other is that the human head is modeled as a rigid object. The former one will be violated when user has quick motion, while the latter will be destroyed when there are extreme expression variation. Our future work will lie on introduction of motion compensation between the pattern-illuminated image and the subsequent scene-texture image, and of the use of 3D deformable model that embraces facial expression variation, so that the two assumptions could be relaxed.

The proposed head estimation method could be adopted in many applications. The driver-assistance system is an example, in which the driver's mental state

reflecting from the head pose could be monitored, once the driver is tired and sleepy, the system can give a warning to wake up the driver. Our future work will try to integrate the proposed method to practical applications.

Chapter 4

Embedding Imperceptible Patterns into Regular Projection

In this chapter, we describe an approach of embedding codes into projection display for structured light based sensing, with the purpose of letting projector serve as both a display device and a 3D sensor. The challenge is to make the codes imperceptible to human eyes so as not to disrupt the content of the original projection. There is the temporal resolution limit of human vision that one can exploit, by having a higher than necessary frame rate in the projection and stealing some of frames for code projection. Yet there is still the conflict between imperceptibility of the embedded codes and the robustness of code retrieval that has to be addressed. We introduce noise-tolerant schemes to both the coding and decoding stages. At the coding end, specifically designed primitive shapes and large Hamming distance are employed to enhance tolerance toward noise. At the decoding end, pre-trained primitive shape detectors are used to detect and identify the embedded codes – a task difficult to achieve by segmentation that is used in

general structured light methods, for the weakly embedded information is generally interfered by substantial noise. Extensive experiments including evaluations of code imperceptibility, decoding accuracy and sensitivity analysis show that the proposed system is effective, even with the prerequisite of incurring minimum disturbance to the original projection.

4.1 Introduction

The improving performance and declining price of digital video projectors make it possible to use them prevalently. Being able to generate arbitrarily large display is a feature of projectors that makes them exceedingly attractive, especially in applications that demand portability. On the other hand, the adoption of structured light illumination has been proven to be an effective and accurate means for 3D information perception [82]. Recently, the availability of pico projectors with average dimensions of $4\times2\times1$ inches has widely extended the application domain of structured light system. There are already pocket DCs, DVs and cellular phones (as shown in Fig. 4.1) in the consumable market that have both projector and camera built-in, making it possible to implement structured light system in handheld consumer electronic products.

In other words, projector accompanied by camera has the potential of being a device for both display and sensing, i.e., for both input and output in human-computer interface, making it a possible device to replace traditional LCD panel, keyboard, and touch-sensitive screen altogether in computing, at the cost of only diminished size and weight. Projector has the potential of making a breakthrough of dramatically downsizing portable computing without sacrificing display size.

For these reasons projector-camera (ProCam) system has been actively researched in the last few years. Many research groups apply projectors in un-

Figure 4.1: Mobile devices with pico projector.

conventional ways to develop new and innovative information displays that go beyond simple screen presentations [123].

Some researchers designed structured light system in the non-visible spectrum [48]. That way the media for regular projection and structure light sensing can be made separate. However, additional hardware could be reduced and device size could be diminished if structured light and regular projection can be achieved through the same projector. This leads to the concept of Imperceptible Structured Light (ISL). ISL modulates the projected display either spatially or temporally to embed code patterns for structured light sensing. In principle, due to limitation of human visual perception, the embedded code patterns can be made undetectable to the user, but cameras synchronized to the modulation are able to reconstruct the embedded codes for structured light sensing. The embedding of code patterns into regular projection can be used for a variety of applications including projector calibration, camera tracking, and 3D scanning.

There is however challenge in embedding codes into regular projection. While

the codes should be made as undetectable as possible to the user, they have to be decodable to the camera for the purpose of structured light sensing. On top of the dilemma, there is the inevitable fact that the displayed signals are generally corrupted by substantial noise that arises from the nonlinearity of the projector, the sensing defects of the camera, and the variation of the ambient illumination. The objective of this work is to deal with the dilemma.

This chapter describes a novel method of embedding imperceptible structured codes into arbitrarily intended projection. Through precise projector-camera synchronization, structured codes consisting of three primitive shapes are embedded into the projection, in a way that is imperceptible to viewers but extractable from the ”difference image” between successive images captured by a camera. To make the decoding process more robust against noise, we do not extract the codes by region segmentation in the image domain. Instead we employ specially trained classifiers to detect and identify the codes. To enhance the error tolerance further, specially designed primitive shapes and large Hamming distance are adopted in the spatial coding. Even with some bits of the codewords missed or wrongly coded, the correct correspondence could still be derived correctly.

The remainder of this chapter is structured as follows. In Section 4.2, related works on imperceptible structured light sensing are briefly reviewed. The principle of embedding imperceptible codes along with robust coding and a noise-tolerant decoding mechanism are described in Section 4.3. In Section 4.4, system setup and experimental results are shown. Summary is offered in Section 4.7.

4.2 Previous Works

A proof of concept for embedding invisible structured light patterns into DLP (Digital Light Processing) projections first appeared in the ”Office of the Future”

project [136]. In this work, binary codes are embedded by projecting temporally alternating code images and their complements. Provided that the frequency of projection reaches the *flicker fusion threshold* ($\geq 75Hz$), the pattern and the inverse pattern are visually integrated over time in human perception, and the illumination has the appearance of a flat field ("white" light) to humans. However, the demonstration required significant modification effort on the projection hardware and firmware, including removal of the color wheel and reprogramming of the controller. The resulting images were also in greyscale only. The implementation of such a setting was impossible without mastering and full access to the projection hardware.

Cotting et. al. introduced a coding scheme [34] that synchronizes a camera to a specific time slot of a DLP micro-mirror flipping sequence in which imperceptible binary patterns are embedded. However, not all mirror states are available for all possible intensities, and the additional hardware, DVI repeater with tapped vertical sync signal, is not an off-the-shelf instrument.

However, with the development of digital projection technology, some so-called 3D compatible DLP projectors with fresh rate of $120Hz$ or higher emerged recently. This makes it possible to implement imperceptible structured light without any hardware modification or extra assisting hardware. Many researcher began to study how to determine the embedded intensity properly to guarantee the code imperceptibility.

In [61], subjective evaluation results and their statistical analysis on the visual perceptibility of embedded codes in different ways were reported. The factors affecting code visibility are also concluded. Park et al. [127] presented a technology for adaptively adjusting the intensity of the embedded code with the goal of minimizing its visibility. It was regionally adapted depending on the spatial variation of neighboring pixels and their color distribution in the YIQ color space. The final

code intensity was then weighted by the estimated local spatial variation. Since two manually defined parameters adjusted the overall strength of the integrated code, the system was not able to automatically calculate an optimized intensity. Grundhofer et al. [9] proposed a method considering the capabilities and limitations of human visual perception for embedding codes. It estimated the Just Noticeable Differences (JND) based on the human contrast sensitivity function and adapted the code intensity on the fly through regional properties of the projected image and code, such as luminance and spatial frequencies. The shortcoming of this method was that some parameters need be pre-measured using some optical devices (e.g. photometer), which were not accessible to nonprofessional users.

To the best of our knowledge, up to now, few works focus on the decoding method in imperceptible code embedding configuration, especially when huge external noises exist.

4.3 Method

4.3.1 Principle of Embedding Imperceptible Codes

The fundamental principle behind imperceptible structured code embedding is the temporal integration achieved by projecting each image twice at high frequency: a first image containing actual code information (e.g., by adding or subtracting a certain amount (Δ) to or from the pixels of the original image, depending upon the code) and a second image that compensates for the distortion in the first image. The vital aspects of ISL sensing are code embedding and projector-camera synchronization.

Since general projection is in color, it is possible to embed color code through three different channels theoretically. However, to enhance code robustness to-

ward noise, we use binary code and embed it into all three color channels simultaneously. Let B, O, I and I' be the binary code image, the original image, the projected image, and the complementary image respectively. Then the projected image and complementary image could be formulated as

$$I_i(x,y) = O_i(x,y) + P(x,y), \tag{4.1}$$

$$I'_i(x,y) = O_i(x,y) - P(x,y), \tag{4.2}$$

$$P(x,y) = \begin{cases} \Delta, & when \quad B(x,y) = 1; \\ 0, & when \quad B(x,y) = 0. \end{cases} \tag{4.3}$$

where i = {R,G,B} indicates red, green and blue channels, Δ is the embedded intensity.

To avoid intensity saturation at lower and higher intensity levels when adding or subtracting Δ, the original image needs to have the intensity range in each color channel compressed to between Δ to $255 - \Delta$. Since the embedded intensity required in the coding is small enough, the visual degradation due to contrast reduction is negligible.

The degree of imperceptibility thus depends upon the embedded intensity. A larger intensity ensures that the code be more tolerant toward noise and more readable in the image of the projection, whilst a smaller intensity makes the embedded codes more invisible. In our design, code imperceptibility has higher priority, and thus embedded intensity is set to a very small value.

In order to achieve imperceptible structured light projection, the frequency of projection must exceed the flicker fusion threshold, which is $75Hz$ for most of the people. Here we take one projection-capture cycle as an example to explain the strategy of projector-camera synchronization, which is illustrated in Fig. 4.2. Firstly, we ensure that the projector projects an image every $10ms$, i.e., at $100Hz$.

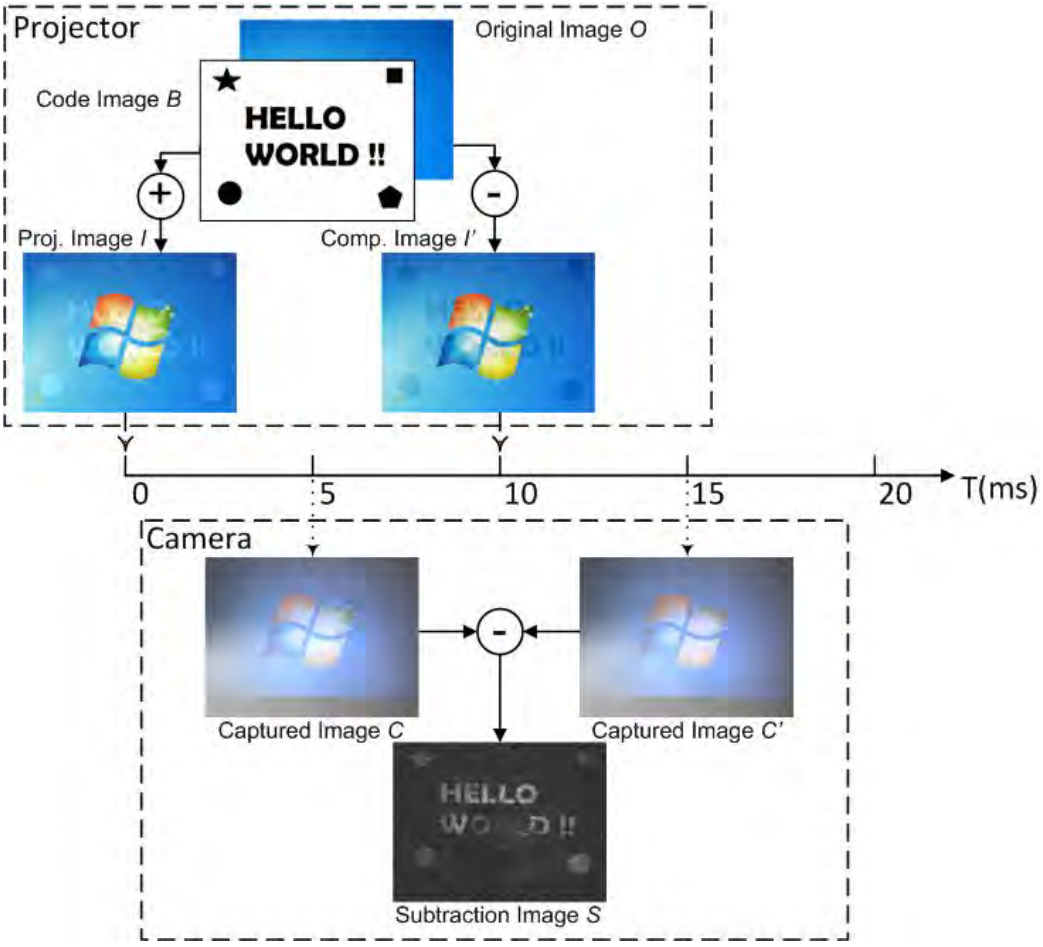


Figure 4.2: Projector-camera synchronization and basic principle for embedding and extracting imperceptible codes.

As shown in Fig. 4.2, along the time axis, the projected image I and the complementary image I' are projected at the time instants $0ms$, $10ms$ respectively. With a refresh rate of the camera at about 100 frames per second, the camera captures the image C and C' at $5ms$ and $15ms$, shortly after the projector projects the projected image and complementary image to the scene. At $20ms$ a new projection-capture cycle will resume. With the aforementioned projection-capture strategy, the system could capture 50 image pairs per second.

The embedded codes could be internally and simply extracted from the "subtraction image" [1] between consecutively captured images as

$$S(x, y) = \max_i [C_i(x, y) - C'_i(x, y)], \quad i = \{R, G, B\}. \tag{4.4}$$

Ideally, the subtraction image should be a binary image that has maximum value of 2Δ and minimum value of 0. However, the subtraction image in reality is generally disturbed by large external noises. Since the embedded intensity is always small, the subtraction image has low signal-to-noise ratio. It is generally nontrivial to retrieve the embedded codes. In the rest of this section, we describe how robust coding and noise-tolerant decoding approaches can help tackle the issue.

4.3.2 Design of Embedded Pattern

The strategy of encoding in general structured light methods could be classified into two categories [82]: time multiplexing and spatial multiplexing. The former one can achieve denser data samples with higher accuracy, but at the expense of requiring multiple illuminations and image captures over time, which is not suitable for imperceptible code embedding [61] and dynamic scenes. In contrast,

[1] All the subtraction images in this chapter are scaled to [0, 255] for illustration purpose.

Figure 4.3: The primitive shapes: (a) cross, (b) sandglass, (c) rhombus.

the latter one labels each pattern position by the appearance profile (color, shape or their combination) of the neighboring positions. The appearance profile can be about various gray levels, colors, or geometric primitives, and the coding methods include De-Bruijn sequences [25, 145, 126], M-arrays [44, 117, 115, 125], and non-formal coding [110, 45, 95, 66]. The spacial coding scheme has the advantage that 3D determination could be achieved with a single pattern.

Considering the constraints of imperceptible code embedding, we employ the spacial multiplex scheme to design our pattern. Due to the choice of using binary code for robust code embedding, the symbols cannot be coded with different colors, so we use an alphabet set comprising three different geometrical primitives: cross, sandglass, and rhombus, as shown in Fig. 4.3. There are three advantages of this configuration. First, all the shapes own a natural center point, which simplifies the shape identification process in the decoding stage. Then, there are sufficient variations between different shapes; even with large disturbance from noise on the shapes, the decoding method could discriminate them. Moreover, the directional information carried by the cross shape could rectify the observation window during the step of neighborhood detection without enforcing any other constraints.

In the decoding stage, the centroid of each detected primitive would be considered as the feature point position, and the 9-bit codeword associated to each feature point is composed of the elements in the 3×3 window centered on it. In traditional structured light methods, the uniqueness of the codeword is usually assured by M-arrays (perfect maps), which are random arrays of dimensions $r \times v$ in which a sub-matrix of dimensions $n \times m$ appears only once in the whole pattern

[44]. The M-arrays give a total of $rv = 2^{nm} - 1$ unique sub-matrices in the pattern and a window property of $n \times m$. However, the Hamming distance between the codewords is 1, which is generally too small for our code embedding scenario in which the codeword retrieval errors could be large due to noise. In our system, we generate a matrix of dimensions 27×29 using the method proposed by Albitar [13], in which 95.97% of the codewords have a Hamming distance higher than 3 and the average Hamming distance is $\bar{H} = 6.0084$, so that even some bits in the codeword are missed or incorrectly coded, the codeword is still distinguishable. On the basis of this matrix, the binary code image composed from the primitive shapes appears like the one illustrated in Fig. 4.4, in which the size of each primitive shape is a collection of 11×11 pixels while the interval between each shape is 11 pixels. The total number of feature points is 783.

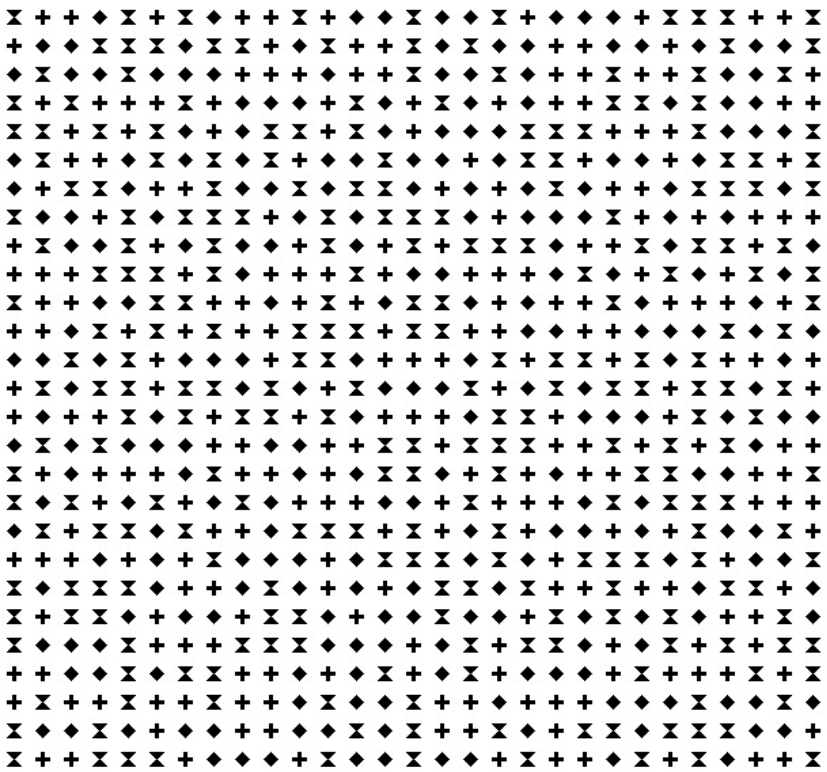


Figure 4.4: The embedded binary code image.

4.3.3 Primitive Shape Identification and Decoding

In the decoding stage, the existence of intense noises (from projector projection, camera sensing, ambient illumination and object surface reflection influence) makes it impossible to segment the primitive shape by the integrated use of region segmentation and edge or contour detection as in ordinary structured light methods. Here, we regard the primitive shapes as objects to "identify" and "detect" rather than "segment".

Compared with other object identification or recognition methods, the machine learning approach proposed by P. Viola [167] has been shown to be capable of processing images rapidly with high detection rates for visual object detection. The approach is adopted here for training detector to identify the three primitive shapes. Below we use cross shape as an example to describe the procedure of detector training.

The performance of training-based detector has a great deal to do with the availability of training samples. Unlike generic objects like human face, body or vehicle, which have a large number of samples in a great many of public databases, we have to collect the specific training samples ourselves in the required configuration. 500 color images with different contents were collected from Google Image [2], and 40 cross shapes were embedded in those images at different positions to generate 500 pairs of projected images and complementary images.

A white planar projection screen was placed in front of the projector-camera system with the distance of $800mm$, the orientation of the screen was adjusted to make the projection area appear as a rectangle, i.e. the projection screen was parallel to the projection plane of the projector. By projecting the images, 500 subtraction images could be derived from image capture exercises. The sub-images containing cross shapes were then segmented by manual labeling, which were considered as positive training samples. The background images with holes filled

by random noise were divided into small patches to generate negative training samples. The training sample preparation process is shown in Fig. 4.5.

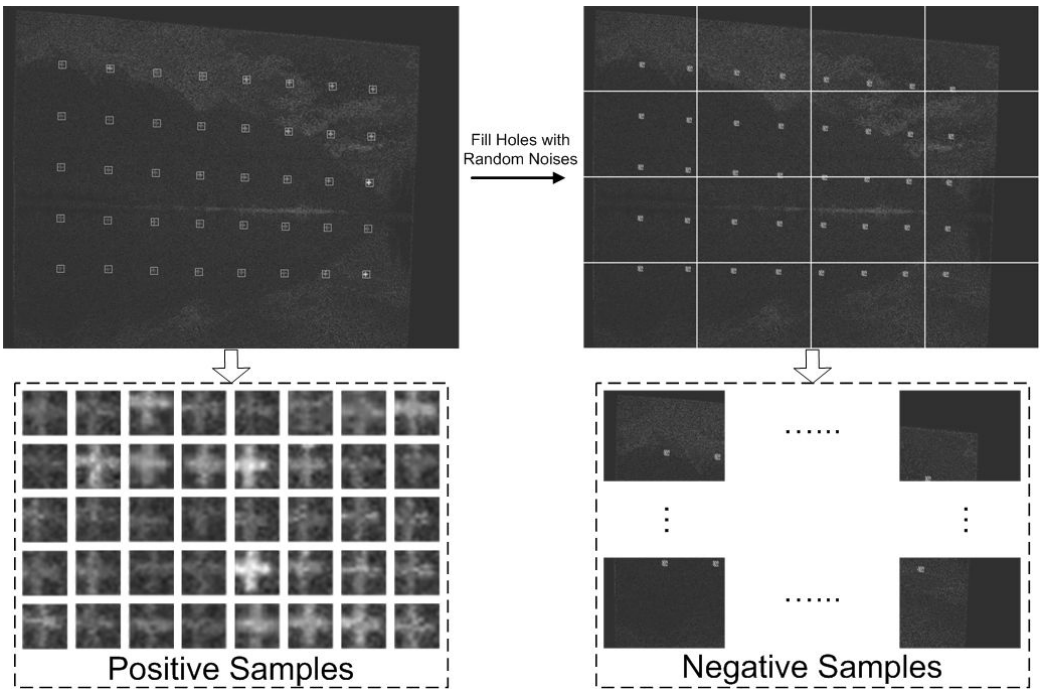


Figure 4.5: Training sample preparation.

To obtain the optimal performance, the positive samples were resized to 20×20, the extended haar-like features and Gentle Adaboost algorithm were employed, following the suggestion in [132]. Eventually, from over 7000 positive samples and 3000 negative samples, a 16-stage cascade classifier for cross detection was trained. Following the same procedure, the detectors for sandglass and rhombus shapes could be derived as well.

4.3.4 Codeword Retrieval

By using the pre-trained primitive shape detectors, the centroid of each primitive, i.e., the position of each feature point, can be determined. Once a feature point is extracted from the image, its codeword can be produced from the associated

3×3 intensity window centered on the feature point. As shown in Fig. 4.6, the codeword of P_0 is calculated as $CW = \sum_{i=0}^{8} 10^i \times C_i$, where C_i is the code of point P_i. It is time-consuming and inefficient for searching the primitive shapes in the whole image, the directional information embraced in the cross shape could rectify the search window around it to find the other two shapes. As illustrated in Fig. 4.6, the cross shapes are detected first, then two directions are fitted through the intensity distributions in the detected rectangle, and in the end, rhombus and sandglass shapes are detected in the nearby area along the two directions. The corresponding point on the projector image plane is known a priori. This way 3D position on the object surface can be determined via triangulation. The above is the 3D sensing step we use in the system.

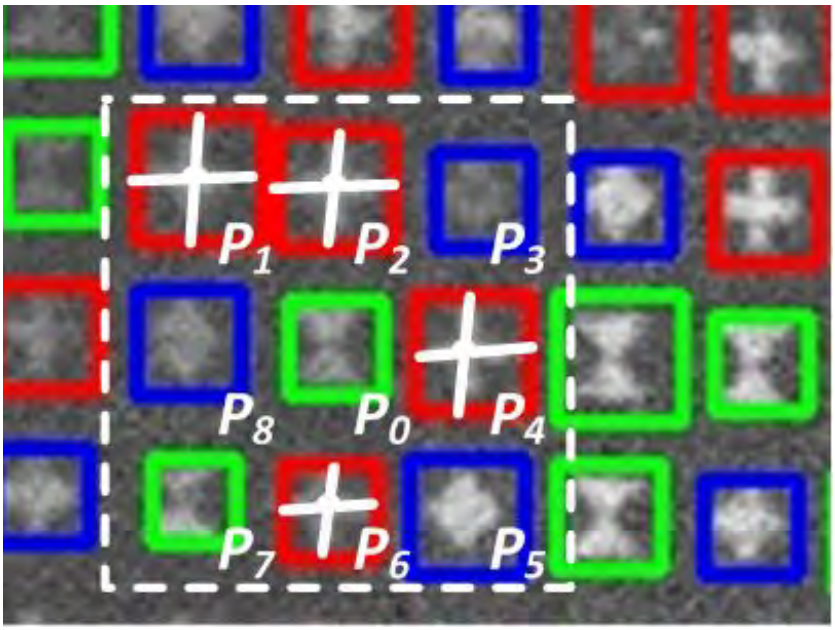


Figure 4.6: An example of codeword retrieval.

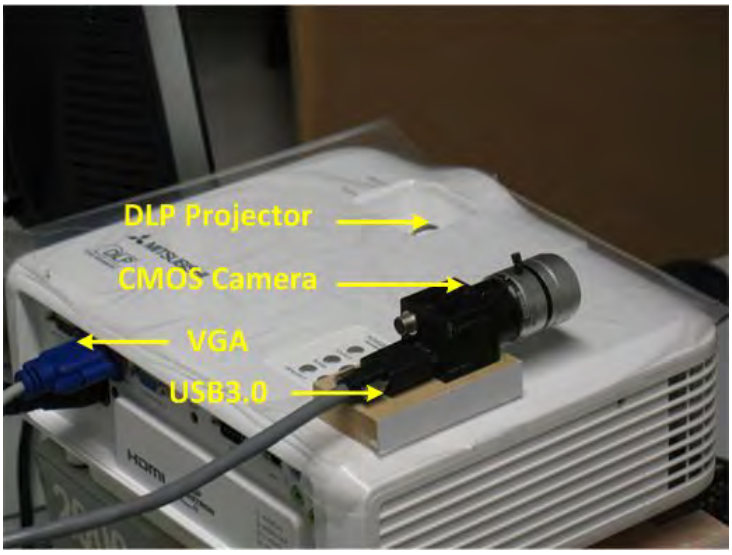
4.4 Experiments

4.4.1 Overview of Experiment Setup

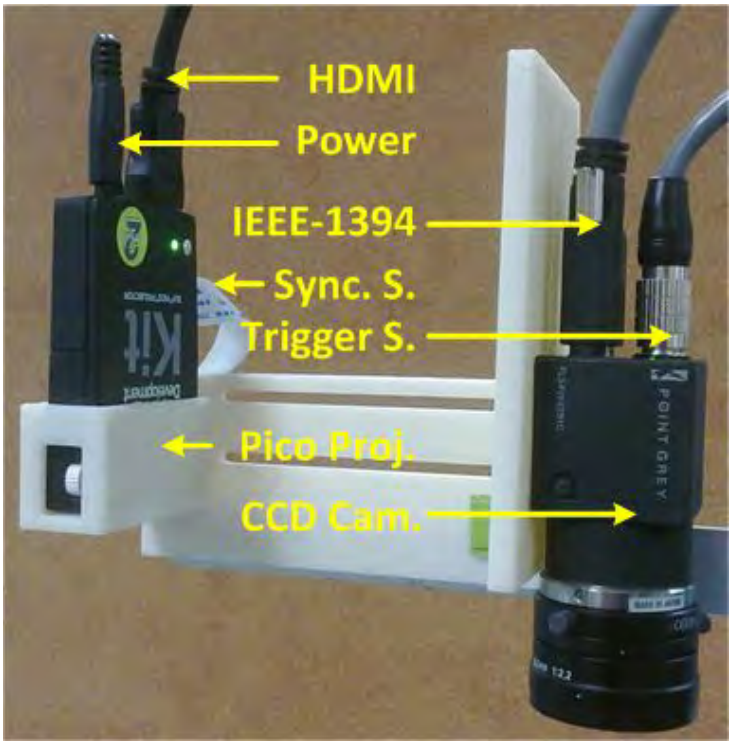
To assess the feasibility of the proposed method for embedding imperceptible codes in regular projection, we conducted experiments on embedded code imperceptibility evaluation, primitive shape detector accuracy evaluation and primitive shape detector sensitivity evaluation.

In order to evaluate the performance of our method in different platforms, we set up two projector-camera systems using different equipment. The first one (*PROCAMS-A*) consisted of a consumer-level DLP projector (Mitsubishi EX240U projector) of 1024×768 resolution and $120Hz$ refresh rate, and a CMOS camera (Point Grey Flea 3 FL3-U3-13S2C with Myutron FV1520 f15mm lens) of 1328×1048 resolution and $120fps$. While the second one (*PROCAMS-B*) consisted of a Pico DLP projector with a native resolution of 640×480 and an interface for firmware configuration (TI DLP Pico Projector Development Kit 2), plus a CCD camera of 648×488 resolution at $120fps$ (Point Grey FL3-FW-03S1C camera with Myutron FV0622 f6mm lens).

For *PROCAMS-A*, we first fixed the camera and projector rigidly, and the projector and camera were connected to a desktop computer through VGA and USB3.0 interfaces respectively. Since there was no synchronization signal output in the consumer-level projector, the synchronization between projectors and cameras was implemented through software delay. The hardware configuration is shown in Fig. 4.7(a). For *PROCAMS-B*, the projector and camera were mounted on a special designed framework rigidly, and were connected to a laptop computer through HDMI and IEEE-1394 interfaces respectively, and the hardware trigger signal of the camera was connected to the sync. output of the projector for synchronization between them, which are illustrated in Fig. 4.7(b).



(a) PROCAMS-A



(b) PROCAMS-B

Figure 4.7: Hardware configuration of two projector-camera systems.

Moreover, the projector-camera systems were calibrated using an LCD monitor as the calibration object; the calibration method, detailed in [154], could derive the intrinsic and extrinsic parameters of the two instruments. Once the experimental system was set up and calibrated, we could conduct further experiments.

4.4.2 Embedded Code Imperceptibility Evaluation

Embedded code imperceptibility and user satisfaction are of the first priority in the system design. The imperceptibility depends on the embedded intensity. We conducted a subjective evaluation using *PROCAMS-A* based on a questionnaire. Ten persons were invited to participate in this experiment, of which six were male and four were female, and seven wearing glasses. Another 500 images were collected from Google Image [2] randomly, the content of the images included natural scene, portrait, architecture, animals and so on. Our proposed pattern was embedded into all the collected images with different intensities. The viewers were seated in front of a white planar screen at a distance of about $1m$, and asked to comment on the images projected to the screen. The questions asked were simplified from the questionnaire in [61], focusing on the feeling of flickering, the recognition of image deterioration, and the overall satisfaction for projection quality. The score for each question was divided into 10 levels.

The average scores of the subjective evaluation are illustrated in Fig. 4.8. When the embedded intensity is small, i.e., $\Delta = 5, 10$, the viewer could rarely notice the embedded codes and were satisfied with the projection quality. With the increase of the embedded intensity, the viewers' sense of flickering and image degradation became stronger. When $\Delta = 25$, almost every viewer was not satisfied with the projection quality.

In practice, because it was difficult to retrieve weakly embedded codes with the standard commercial cameras, we choose $\Delta = 10$ in our configuration, striking a

compromise between user satisfaction and code imperceptibility.

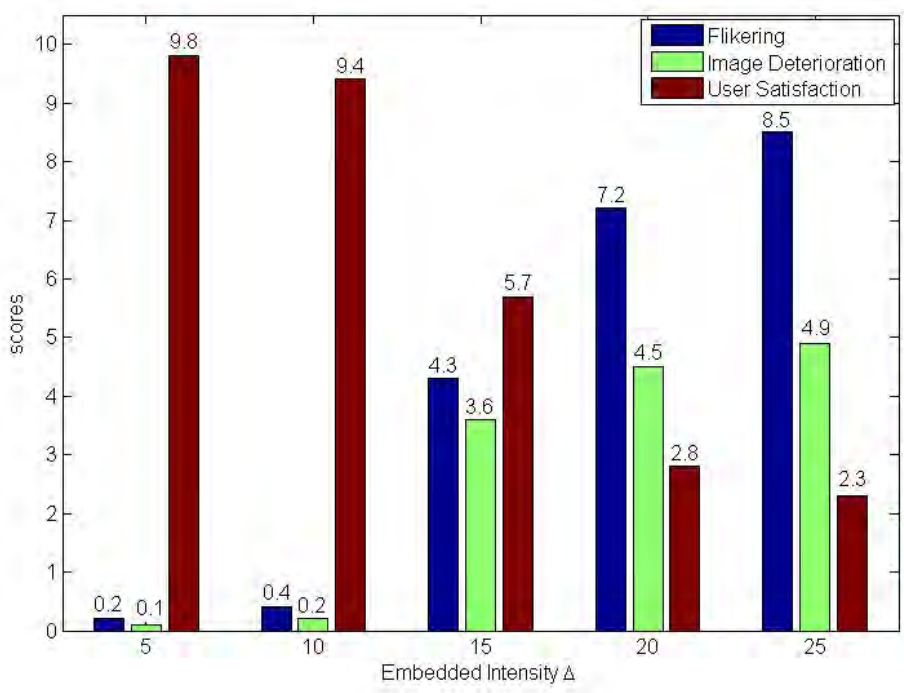


Figure 4.8: Subjective evaluation results for code imperceptibility.

4.4.3 Primitive Shape Detection Accuracy Evaluation

After embedded code imperceptibility evaluation, the experiments for primitive shape detection accuracy were carried out. Considering the training data for primitive shape detector training was collected by *PROCAMS-A*, we first evaluated the primitive shape detection accuracy on *PROCAMS-A*.

To assess accuracy, the experimental data with ground-truth were required. Three different primitives and the spatially coded pattern image were embedded into 500 images used for imperceptibility evaluation respectively, with intensity $\Delta = 10$. Then the projected and complementary images were projected successively to a projection surface, while the camera conducted synchronized capture.

The projection surface is the same as the one used for training data collection. Then the subtraction images embracing embedded codes information were derived for accuracy evaluation. The ground-truth was obtained by manual labeling in the image data captured under binary pattern illumination.

Experimental results in some subtraction images are presented in Fig. 4.9. The four sub-figures display the cross (top-left), sandglass (top-right), rhombus (bottom-left) shapes, and the spatially coded pattern (bottom-right) respectively. For qualitative evaluation, the detected features are indicated by rectangles, and in bottom-right sub-figure, the cross,sandglass and rhombus shapes are separately marked by red, green and blue rectangles. The accuracy of primitive detector are evaluated by hit rate (H), missing rate (M), false rate (F) and position error (E_d), which are formulated as

$$H = \frac{N_h}{N_t}, \tag{4.5}$$

$$M = \frac{N_m}{N_t}, \tag{4.6}$$

$$F = \frac{N_f}{N_t}, \tag{4.7}$$

$$E_d = \sqrt{\epsilon_X^2 + \epsilon_Y^2}, \tag{4.8}$$

$$\epsilon_X = \frac{1}{N_h} \sum_{i=1}^{N} |X_d - X_g|_i, \tag{4.9}$$

$$\epsilon_Y = \frac{1}{N_h} \sum_{i=1}^{N} |Y_d - Y_g|_i, \tag{4.10}$$

where N_t is the total embedded primitive shape number, N_h, N_m and N_f are the number of correct detections, missed detections and false detections respectively. ϵ_X and ϵ_Y are the average feature point detection errors along the x-axis and y-axis, (X_d, Y_d) and (X_g, Y_g) are the detected coordinate and ground-truth respectively.

The more detailed quantitative testing results are listed in Table 4.1. Through the proposed method, 95.74% of the embedded feature points could their correspondences found correctly. By analyzing the missed and false detection cases, we find that the mistakes were mainly caused by large noise that occludes the embedded codes, implying that external noise has the greatest influence on the decoding process.

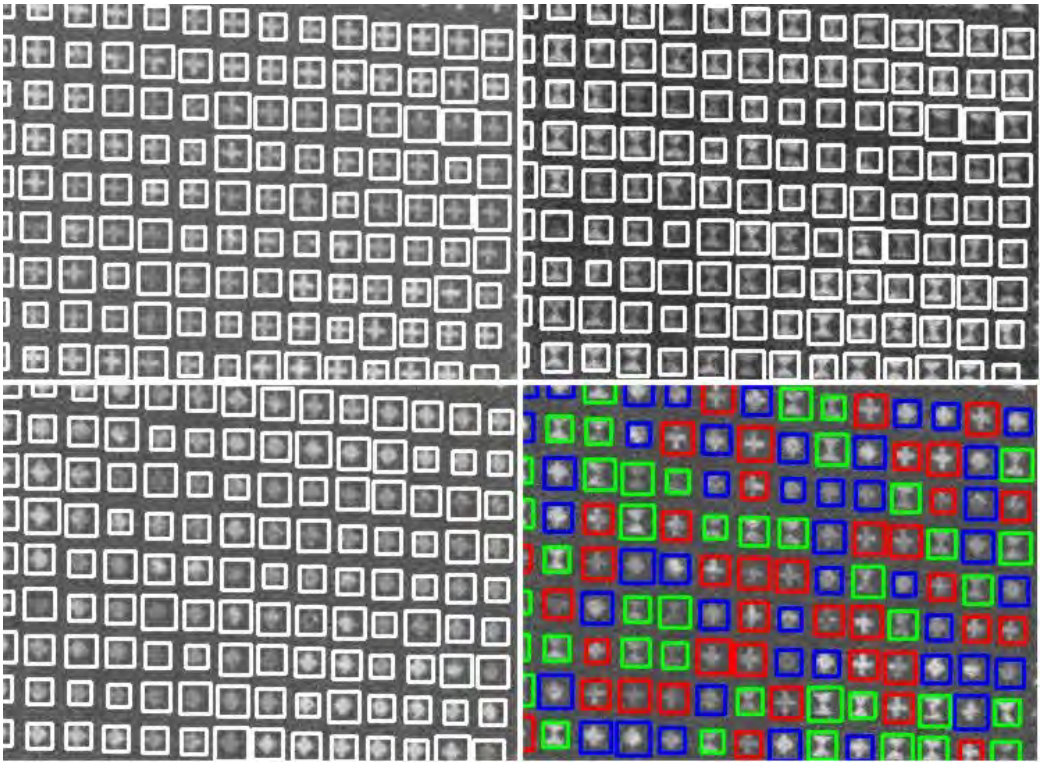


Figure 4.9: Some qualitative experiment results for accuracy evaluation.

4.5 Sensitivity Evaluation

It is obvious that the performance of our method depends on the performance of pre-trained primitive shape detectors, which is determined by the training process to a great extent. Generally, for the training based methods, generalization of the

Table 4.1: Benchmark for sensitivity evaluation.

	H(%)	M(%)	F(%)	E_d(pixel)	Corr. Acc.(%)
Cross	94.53	3.95	1.52	1.632	—
Rhombus	95.21	3.59	1.20	1.833	—
Sandglass	95.50	3.63	0.87	1.542	—
Whole Pattern	92.11	11.06	5.28	2.013	95.74

training results is an issue, especially, when the scenarios between training stage and operation stage are quite different.

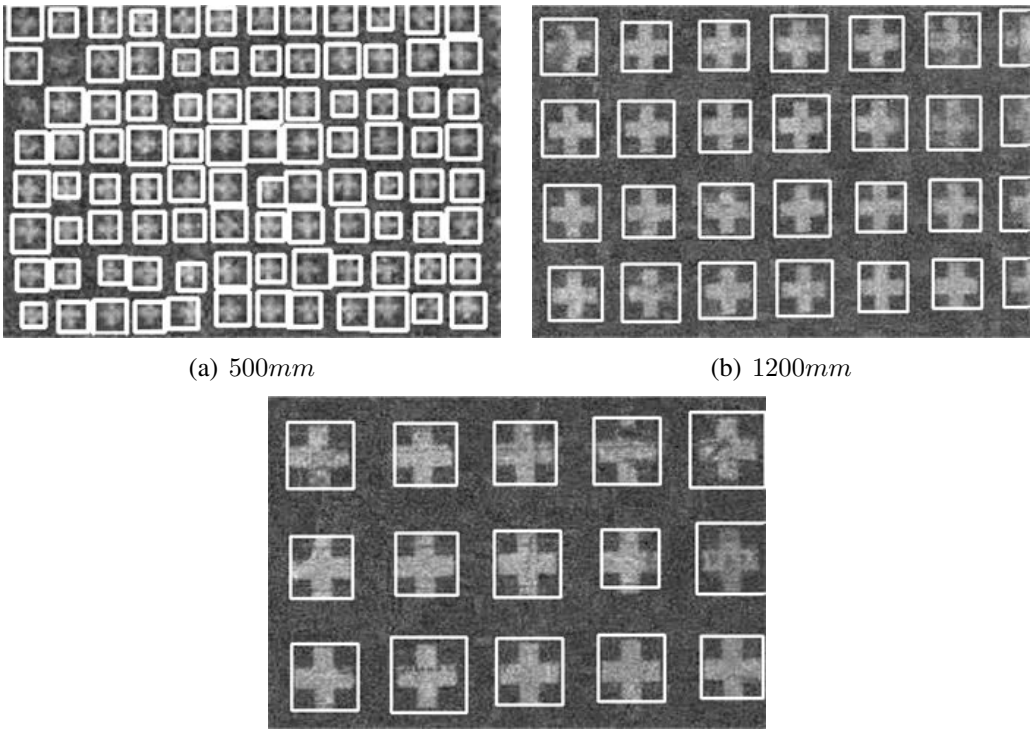
In the framework of our method, due to the different sensor-object localization, different projection surfaces, different surrounding environment and different hardware platforms, the generalization of the pre-trained detector is of great importance, since it is impractical even impossible to re-train the detector for different scenarios. It is necessary to certify the validity of our method in different application scenarios.

In this section, we will evaluate the the sensitivity of primitive detectors under different circumstances, including variations on working distance, projection surface orientation, projection surface shape, projection surface texture and hardware configuration. Since the settings of accuracy evaluation in Section 4.4.3 are the same as training sample collection stage, the results are considered as the benchmark for sensitivity evaluation.

4.5.1 Working Distance

The working distance is the average distance from the projector-camera system to the object surface. When the intrinsic parameters of the projector and camera (focal length and resolution) are fixed, the size of the primitive shapes in subtraction image data is determined by the working distance directly. In the configuration of

training stage, the working distance is set as $800mm$, the size of primitive shapes in image data is about 20 pixels. In the operation stage, the working distance is changed to $500mm$, $1200mm$ and $1600mm$, the focal length of procams is slightly adjusted to get sharp projection and clear capture. Some subtraction images with detection results are shown in Fig. 4.10, the size of the primitive shapes are around 15, 35 and 45 pixels respectively.



(a) $500mm$ (b) $1200mm$

(c) $1600mm$

Figure 4.10: Cross shape detection in different working distances.

The detailed quantitative results are listed in Table 4.2. It is clear that when the working distance decreased to $500mm$, the hit rates dropped, because it is difficult for primitive shape detectors to find small size shapes in image data. For the enlarged shapes in larger working distance, the performance of detectors are almost the same as the benchmark.

Table 4.2: Primitive shape detection accuracy in different working distances.

Distance	Primitive	Hits(%)	Missed(%)	False(%)	E_d(pixel)
$500mm$	Cross	86.21	11.63	2.16	1.814
	Rhombus	85.83	12.57	1.60	1.836
	Sandglass	87.49	11.64	0.87	1.712
$1200mm$	Cross	94.44	4.32	1.24	1.728
	Rhombus	94.86	4.23	0.91	1.904
	Sandglass	94.49	4.62	0.89	1.572
$1600mm$	Cross	94.52	4.11	1.37	1.731
	Rhombus	95.06	3.92	1.02	1.910
	Sandglass	95.39	3.68	0.93	1.591

4.5.2 Projection Surface Orientation

Besides the size of the primitive shapes in image data, the distortions will also influence the performance of the pre-trained detectors. The distortions mainly come from the variations on the orientation of the projection surface w.r.t. the sensing system and the variations on the shape of the projection surface. First, the detector accuracy will be evaluated under different projection surface orientations.

In training data collection stage, the images are projected to a planar surface that is almost parallel to the image plane of the camera. Now in operation stage, the orientation of the surface is adjusted to $10°$, $20°$, $30°$, $40°$, $50°$ in yaw direction, as shown in Fig. 4.11. In each sub-image, the upper part is the captured image to show the extent of distortion, while the lower part is the magnified subtraction image of the subregion indicated by the rectangle in captured image. The detection results are also shown in the subtraction images. More detailed quantitative results are listed in Table 4.3.

In the testing results, when the rotation degree θ is small, i.e.,$\theta = 10°, 20°$, the performance is almost the same as benchmark. With the increase of the rotation degree, the hit rates decrease slightly. When $\theta = 50°$, more than 85% primitive

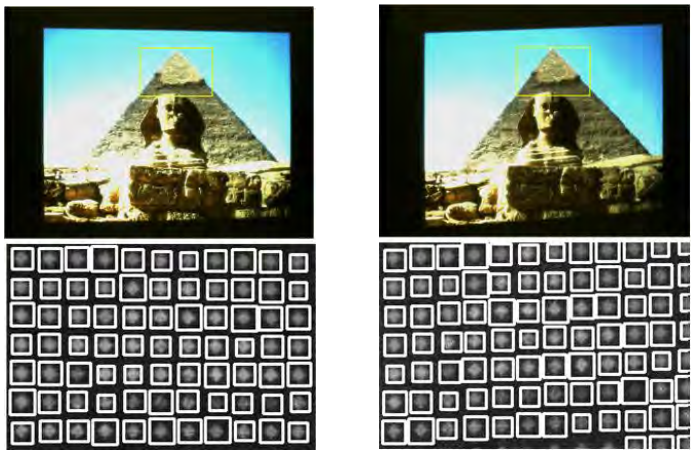
Table 4.3: Primitive shape detection accuracy in different surface orientations.

Orientation	Shape	Hits(%)	Missed(%)	False(%)	E_d(pixel)
10°	Cross	94.51	3.96	1.53	1.635
	Rhombus	95.08	3.60	1.22	1.845
	Sandglass	95.46	3.74	0.80	1.544
20°	Cross	94.50	3.96	1.54	1.634
	Rhombus	95.08	3.64	1.08	1.848
	Sandglass	95.43	3.77	0.80	1.564
30°	Cross	93.47	4.50	2.03	1.938
	Rhombus	92.15	6.37	1.48	2.141
	Sandglass	92.43	6.78	0.79	2.011
40°	Cross	90.19	7.70	2.11	2.414
	Rhombus	89.42	9.50	1.08	2.809
	Sandglass	91.23	7.87	0.90	2.374
50°	Cross	85.91	12.03	2.06	2.728
	Rhombus	85.48	12.81	1.71	2.904
	Sandglass	86.87	12.27	0.86	2.572

shapes are still detected correctly, which satisfies the application requirements.

4.5.3 Projection Surface Shape

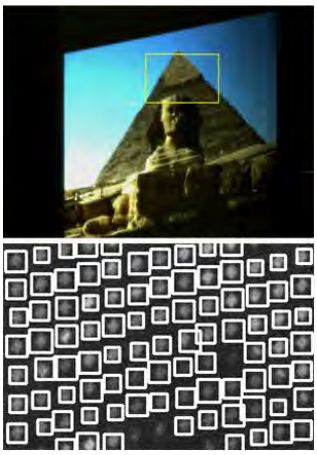
The alteration of projection surface shape will also result in the distortion of primitive shapes in image data. In training stage, the negative and positive sample were collected from the images projected to a planar surface. In this test, the projection surface are three different non-planar surfaces (convex paper, concave paper and plaster statue). Some test images and the statistical results are shown in Fig. 4.12 and Table 4.4 respectively. In all three surfaces, although the hit rates have small decrease, it is still sufficient to derive correct correspondences for triangulation. In the plaster statue case, the missing detections are mainly found in the regions where the surface has sudden change.



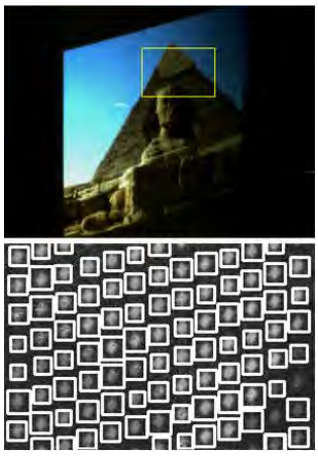
(a) 10°

(b) 20°

(c) 30°

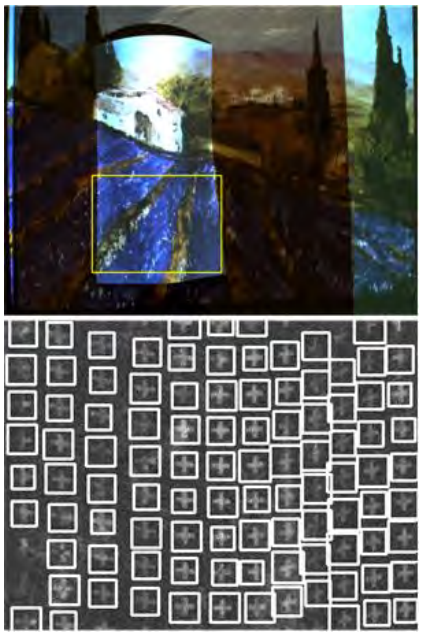


(d) 40°

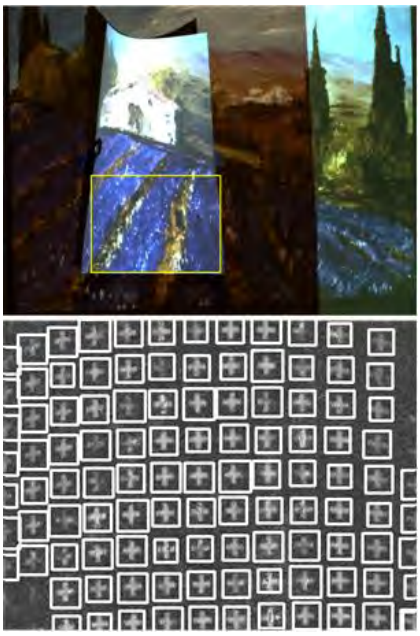


(e) 50°

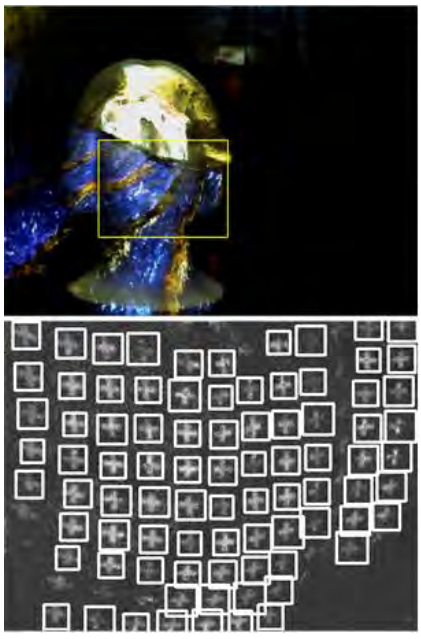
Figure 4.11: Rhombus shape detection in the projection surface with different orientations.



(a) Convex Paper



(b) Concave Paper



(c) Plaster Statue

Figure 4.12: Cross shape detection in different projection surfaces.

Table 4.4: Primitive shape detection accuracy in the projection surface with different shapes.

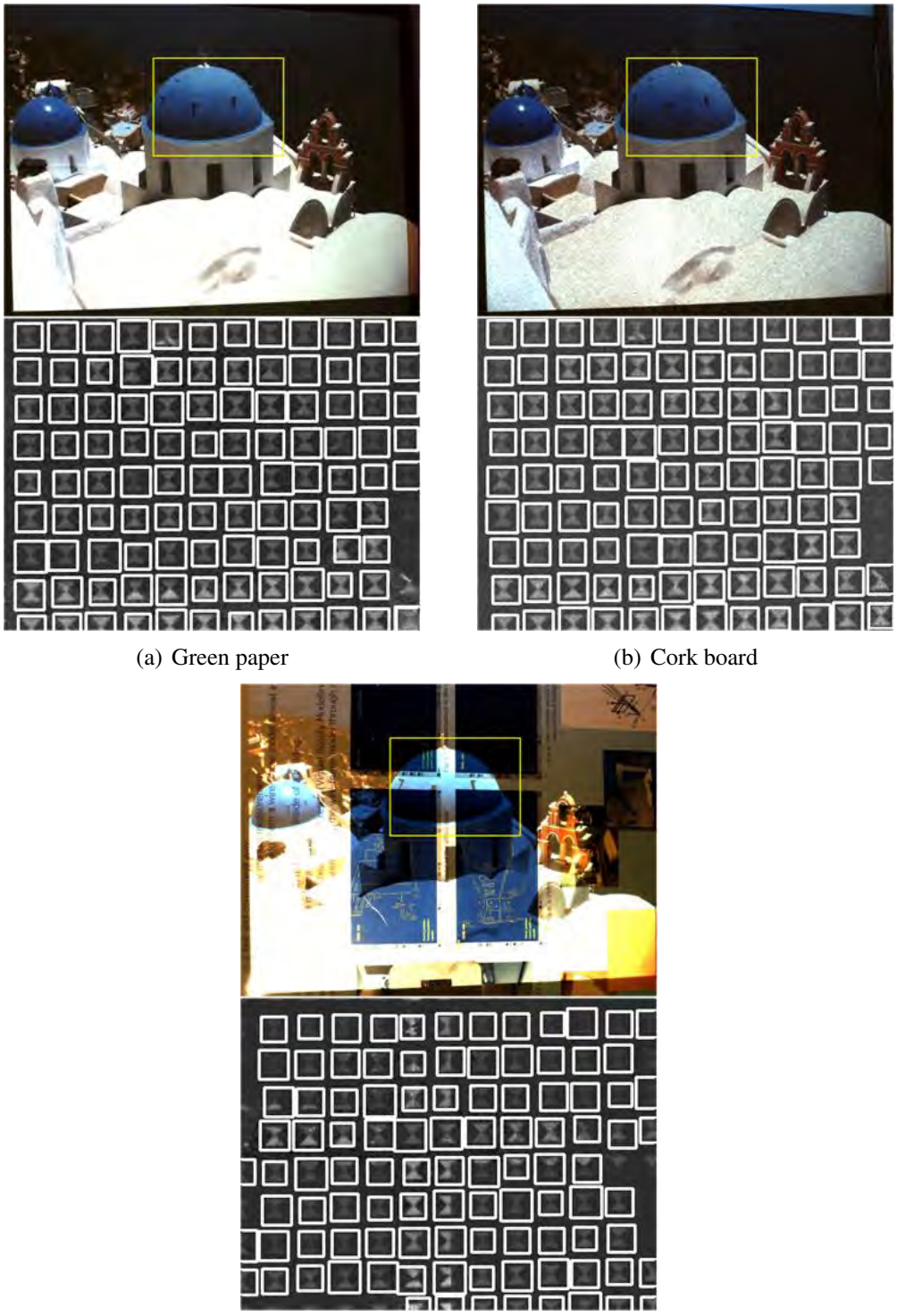
Surface	Shape	Hits(%)	Missed(%)	False(%)	E_d(pixel)
Convex Paper	Cross	93.53	4.86	1.61	1.756
	Rhombus	93.25	5.29	1.46	2.043
	Sandglass	94.14	4.85	1.01	2.122
Concave Paper	Cross	93.64	4.84	1.52	1.762
	Rhombus	93.82	4.70	1.48	2.108
	Sandglass	93.76	5.41	0.83	2.135
Plaster Statue	Cross	84.81	13.33	1.86	2.028
	Rhombus	85.73	13.06	1.21	1.904
	Sandglass	86.09	13.03	0.88	2.075

4.5.4 Projection Surface Texture

The texture on the projection surface will affect the quality of captured images. In the benchmark training stage, the projection surface is textless and in white color. In the operation stage for test, the images are projected to a planar surface in green color, a cork board and a poster with text and images, as illustrated in Fig. 4.13. The quantitative results are listed in Table 4.5. The results indicate that the texture variation on the projection surface has little influence on the performance of primitive shape detectors, since in our method the decoding process was conducted in subtraction image, which would weaken the texture influence to a certain extent.

4.5.5 Projector-Camera System

If the pre-trained detectors are used in another applications with different hardware configuration, the performance of the detectors would be affected, since the differences in the resolution of projector and camera (high vs. low), the camera sensor (CCD vs. CMOS) and the optical parameters (different lens) will change the appearance of the primitive shape in image data. In this test, the primitive de-



(a) Green paper

(b) Cork board

(c) Poster

Figure 4.13: Sandglass shape detection in different projection surface textures.

Table 4.5: Primitive shape detection accuracy in different projection surface texture.

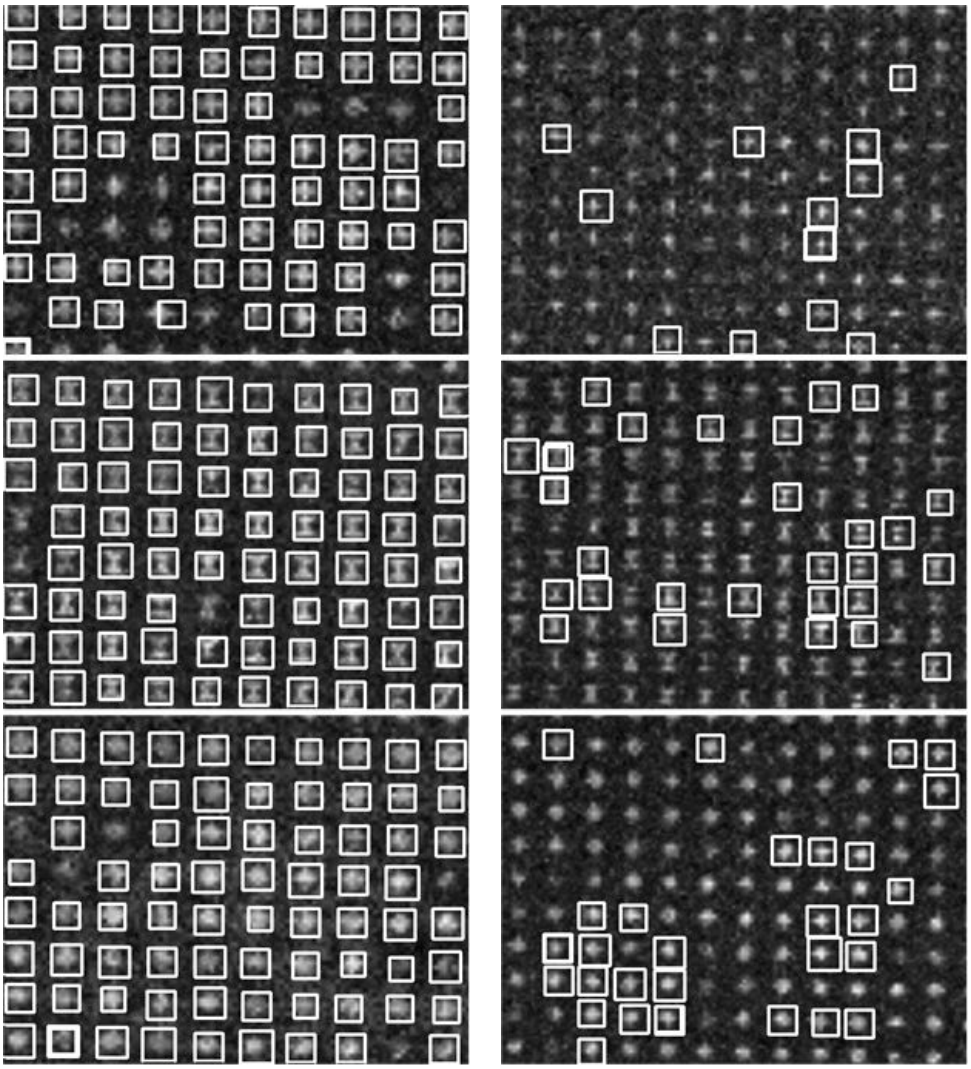
Texture	Shape	Hits(%)	Missed(%)	False(%)	E_d(pixel)
Green Paper	Cross	94.41	4.17	1.42	1.634
	Rhombus	95.19	3.66	1.15	1.836
	Sandglass	95.49	3.63	0.88	1.558
Cork Board	Cross	93.41	5.07	1.52	1.641
	Rhombus	94.25	4.43	1.32	1.850
	Sandglass	94.92	4.16	0.92	1.623
Poster	Cross	91.74	6.63	1.63	2.024
	Rhombus	90.28	8.25	1.47	1.996
	Sandglass	92.19	6.76	1.05	1.762

tectors trained by the data collected from *PROCAMS-A* are applied in *PROCAMS-B* during the operation stage.

Due to the low projector resolution in *PROCAMS-B*, the dimension of the original pattern image is too large for embedding, so we employ two method to solve this issue, the first one is to select a sub-region of the original pattern image as a new pattern image and the second one is to resize the original pattern image to coincide the projector resolution. Some detection results in the subtraction images derived from two different embedding methods are illustrated in Fig. 4.14(b) and 4.14(c). The quantitative results are also shown in Table 4.6.

Compared with the benchmark, it is obvious that the performance in *PROCAMS-B* degrades intensively, especially in the resized pattern case. By analyzing the missed and false detection cases, we find that the mistakes were mainly caused by large noise from the low luminance of the pico projector and the extremely small primitive shapes in image data.

(a) Captured Image



(b) Cropped Pattern (c) Resized Pattern

Figure 4.14: Primitive shape detection in PROCAMS-B with different embedding approaches.

Table 4.6: Primitive shape detection accuracy in PROCAMS-B with different embedding approaches.

	Shape	Hits(%)	Missed(%)	False(%)	E_d(pixel)
Cropped Pattern	Cross	80.23	14.43	5.34	3.028
	Rhombus	79.93	14.17	5.92	2.981
	Sandglass	81.09	13.28	5.63	2.812
Resized Pattern	Cross	30.52	59.23	10.25	2.628
	Rhombus	30.63	58.03	11.34	2.913
	Sandglass	30.80	57.93	11.27	2.874

4.6 Applications

The proposed method enables a common projector to serve the dual role of a display device as well as a 3D sensor, which can be extended or integrated to many applications. In this section, we will show three cases to demonstrate the feasibility of our method.

4.6.1 3D Reconstruction with Normal Video Projection

3D reconstruction is the most straightforward application for structured light sensing, for the sake of showing the effectiveness of our method in 3D reconstruction task, we compared our method with general structured light method using visible patterns.

As shown in Fig. 4.15-(a1)(b1)(c1) and Fig. 4.15-(a2)(b2)(c2), three objects (sphere, cone and cylinder) with known dimensions were illuminated by visible binary pattern image (the same as Fig. 4.4) and code embedded normal projection respectively.

In the general structured light scenario, some feature points were extracted by segmentation and shape identification using the method proposed in [13]; whilst

in our code embedded normal projection scenario, the feature points were detected and classified through the pre-trained primitive shape detectors. The depth value of each feature point was calculated through triangulation using the intrinsic and extrinsic parameters of projector and camera. Then on the basis of point clouds calculated through our method, surfaces were rendered as illustrated in Fig. 4.15-(a3)(b3)(c3). Since the dimensions of the objects are known, we can conducted quantitative accuracy assessment. The residual mean error E_μ and standard deviation E_σ of the calculated 3D points with respect to ground-truth were listed in Table 4.7. It is evident that our method almost has the same performance as general structured light method in 3D reconstruction. By reason that the textures on cylindrical object obstruct the code retrieval, of which the reconstruction error is greater than that of another two objects. It is worth pointing out that in our method the decoding process was conducted in subtraction image, which would weaken the texture influence to a certain extent.

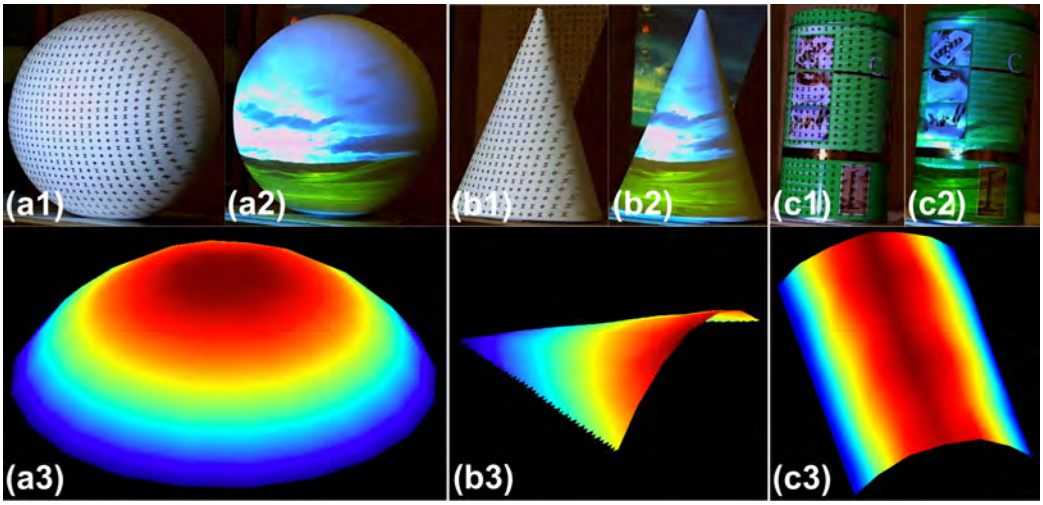


Figure 4.15: Some results of 3D reconstruction.

Table 4.7: The comparison of 3D reconstruction accuracy.

Object	General SL [13]		Our Method	
	$E_\mu(mm)$	$E_\sigma(mm)$	$E_\mu(mm)$	$E_\sigma(mm)$
Sphere	1.502	0.576	1.410	0.587
Cylinder	2.054	0.824	1.939	0.762
Cone	1.383	0.557	1.391	0.564

4.6.2 Sensing Surrounding Environment on Mobile Robot Platform

For the purpose of illustrating the proposed method's potential applications in robotic system working in varied environment, we mounted a projector and a camera rigidly on special designed frame, and then fixed the frame on a tripod affixed on a mobile robot manufactured by ARRICK Robotics [1], as shown in Fig. 4.16.

For a mobile robot, one of the essential capabilities is to sense the surrounding environment for navigation, obstacle avoidance, object recognition and some other purposes. We assist the visual sensing through a normal grey illumination with invisible codes embedded. By retrieving the embedded codes, correspondences between projection plane and image plane could be established accurately and efficiently. In Fig. 4.17 (a) and (c), a green tea can and toy bricks were located in the illumination area of the projector, 3D depth information of certain points on the objects was acquired through simple triangulation in real-time. The surfaces of the objects were rendered in 3D as shown in Fig. 4.17 (b) and (d). Although the ground truth of the objects was not available, such qualitative examinations showed that the reconstructed surfaces were of reasonable quality.

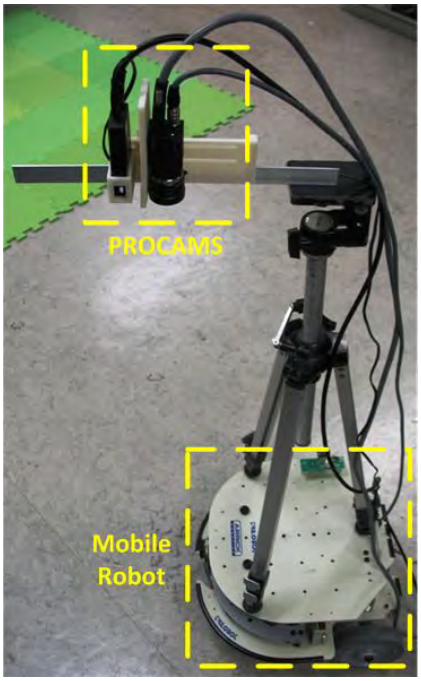


Figure 4.16: Integration with mobile robot system.

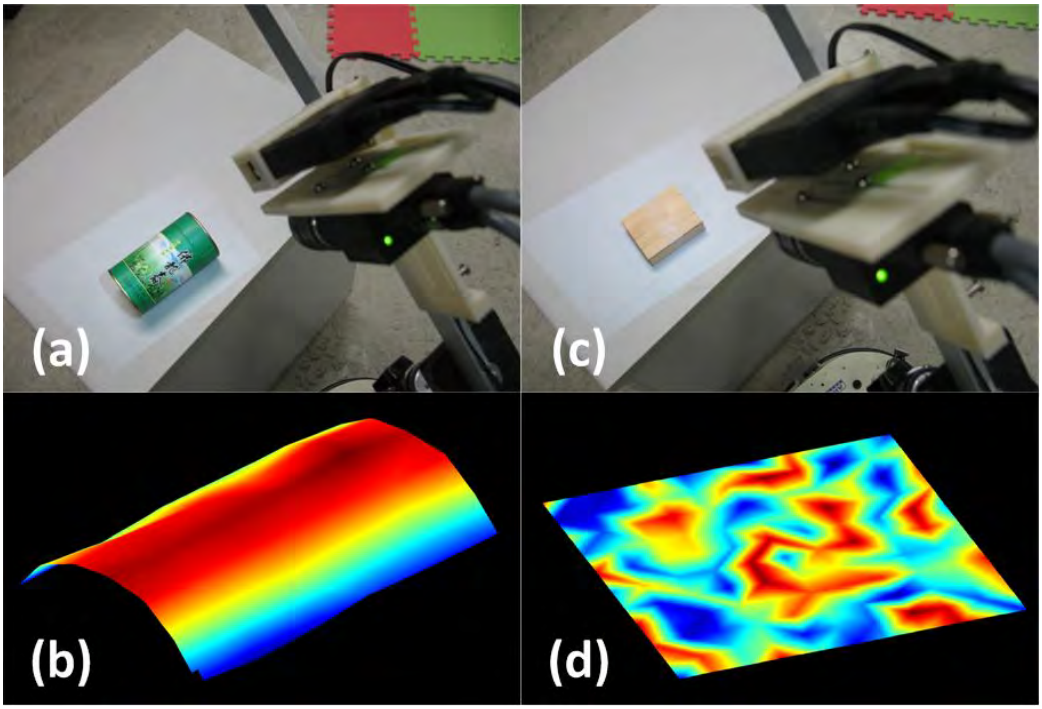


Figure 4.17: Some 3D sensing results.

4.6.3 Natural Human-Computer Interaction

Besides sensing capabilities, the mobile robot should also provide an effective channel for the interaction between users, such as an interface for system configuration or a display panel to show prompt information. In traditional way, an LCD monitor plus mouse-and-keyboard or an LCD touch-screen are attached to the robot, inevitably increasing the weight and size of mobile robot, let alone more energy consumption. Our method enables a common projector to serve the dual role of a display device as well as a 3D sensor with the assistance of camera, providing a platform for more natural user interface schemes. As shown in Fig. 4.18 (a), a system configuration interface (Fig. 4.18 (b)) was projected onto a desk surface, a user was operating on the projected desk surface with bare-hand (Fig. 4.18 (c)). From an image alone, say of a finger on top of a table surface, one cannot tell whether the finger is actually touching the table surface or not. The case of a finger hanging in air, and the case of a finger touching the table surface, could both produce the same image to the camera. By incorporating the structured light invisible embedded into the projection, 3D acquisition can be made possible, and contact identification and finger movement recognition should be more readily tackled. It is possible to convert any textureless light color plane (table-surfaces, whiteboards or walls) to be a touching sensitive screen, providing more natural and flexible interface for bare-hand human-robot interaction.

4.7 Summary

We have described a novel system of embedding imperceptible structured codes into normal projection that strikes the balance between imperceptibility and detectability of the codes. Through precise projector-camera synchronization, structured codes consisting of three primitive shapes are embedded into normal projec-

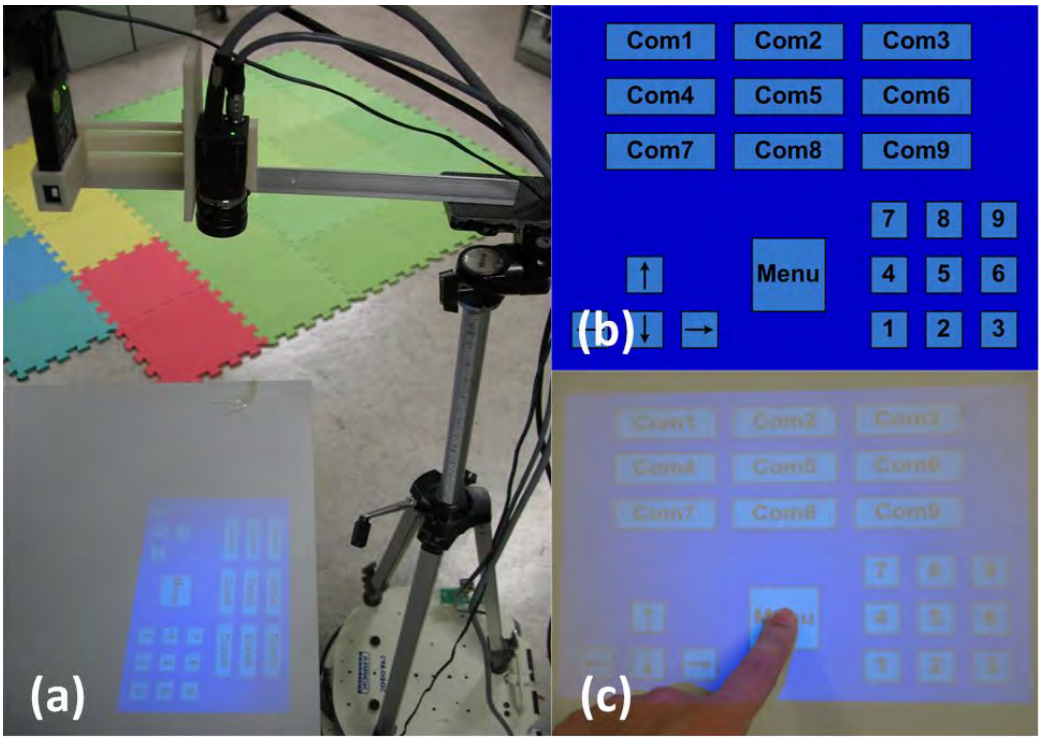


Figure 4.18: Touch sensitive user interface.

tion, in a way that is imperceptible to the user but extractable by a camera (through the ”difference image” between successive images). The disturbances caused by external noise make it difficult to retrieve the codes by the region segmentation approaches adopted in general structured light systems. Instead of segmenting the codes, specially trained classifiers are employed to detect and identify them. To increase the robustness of code extraction, large Hamming distance are adopted in spatial coding. Even if some bits are missed or wrongly decoded, the correct correspondence between the projection panel and the image plane could still be arrived at correctly for structured light sensing. Extensive evaluations shows that the method is a promising one.

In the current system, the image capture interval is $10ms$. In sensing object that moves fast, the substantial displacement between successive images will result in blur or destruction of the embedded codes in the difference image. Some compensation methods need be in place to deal with the problem. In addition, the embedded code could be denser for more precise 3D sensing. New coding scheme capable of generating denser patterns should be used. The proposed method enables a common projector to serve the dual role of a display device as well as a 3D sensor. That provides a platform for more natural user interface schemes. Our future work will lie on these directions.

Chapter 5

Hand Segmentation in Projector-Camera Systems

One goal of projector-camera system is let human finger be used like a mouse to click and drag objects in the projected content. It requires segmentation of the human palm and fingers in the image data captured by the camera, which is a challenging task in the presence of the incessant variation of the projected video content and the shadow cast by the palm and fingers. In this chapter, we describe a coarse-to-fine hand segmentation method for projector-camera system. After rough segmentation by contrast saliency detection and mean shift-based discontinuity-preserved smoothing, the refined result is confirmed through confidence evaluation. Extensive experimental results are shown to illustrate the accuracy and efficiency of the approach.

5.1 Previous Works

Hand segmentation, as the first step for most barehand-based applications, plays an important role in the robustness, accuracy and efficiency of a HCI system. The approaches for hand segmentation have been studied extensively in computer vision society.

Among them, skin color detection [91, 40] is very common for its simpleness and easy implementation, and is very efficient against simple background or in the scene of hand being the only skin-colored object.

For histogram-based skin color detection approach as described in [91], a set of captured images with associated skin masks is used as the training set to train the classifier. Using a bin size of 32 for each color channel, each of the RGB pixels in the training set are assigned to either the 3D skin histogram H_s or the non-skin histogram H_n. Given these histograms we can then compute the probability that a given RGB color belongs to the skin and non-skin classes as follows

$$P(rgb|skin) = \frac{s[rgb]}{T_s}, \tag{5.1}$$

$$P(rgb|\neg skin) = \frac{n[rgb]}{T_n}, \tag{5.2}$$

where $s[rgb]$ is the pixel count in bin rgb of H_s, $n[rgb]$ is the pixel count in bin rgb of H_n, T_s and T_n are the total counts contained in H_s and H_n respectively.

At operation stage, the probability that any given rgb pixel is skin or non-skin can be determined using Bayes rule as

$$P(skin|rgb) = \frac{P(rgb|skin)P(skin)}{P(rgb|skin)P(skin) + P(rgb|\neg skin)P(\neg skin)}, \tag{5.3}$$

$$P(skin) = \frac{T_s}{T_s + T_n}, \tag{5.4}$$

$$P(\neg skin) = 1 - P(skin), \tag{5.5}$$

where $P(skin)$ and $P(\neg skin)$ are the prior probabilities for skin and non-skin respectively.

Therefore, hand region can be segmented with this skin classifier to only keep pixels with a high skin probability:

$$P(skin|rgb) \geq \sigma_s, \tag{5.6}$$

where $\sigma_s \in [0, 1]$ is the threshold value.

However, in projector-camera scenario, diverse video contents are projected continuously, when some skin-colored objects are projected on the background (Region A in Fig. 5.1) or non-skin-colored objects are projected on the hand (Region B in Fig. 5.1), the skin color based methods will be influenced severely.

Since the geometrically and radiometrically calibrated projector-camera system can predict where the video contents are projected and how they should appear in the image data, background subtraction [99] is adopted to segment the hand as the set of pixels that are out of expectation on the projection surface, but suffers from separating hand region from the hand-cast shadows (Region C in Fig. 5.1), let alone calibration procedures and constraints of constant ambient illuminations and fixed projection surface.

The graph-based [46, 138] approaches are able to generate good segmentations. However, the time consuming of these approaches and the requirement of user's interaction would weaken their advantage for the HCI application where the speed is an important factor for realtime interaction.

Rather than monocular camera, some researchers use additional instruments, such as infrared camera [148], stereo camera [170], depth sensor [166], to distinguish hand region from background, that inevitably increasing the complexity of projector-camera system configuration.

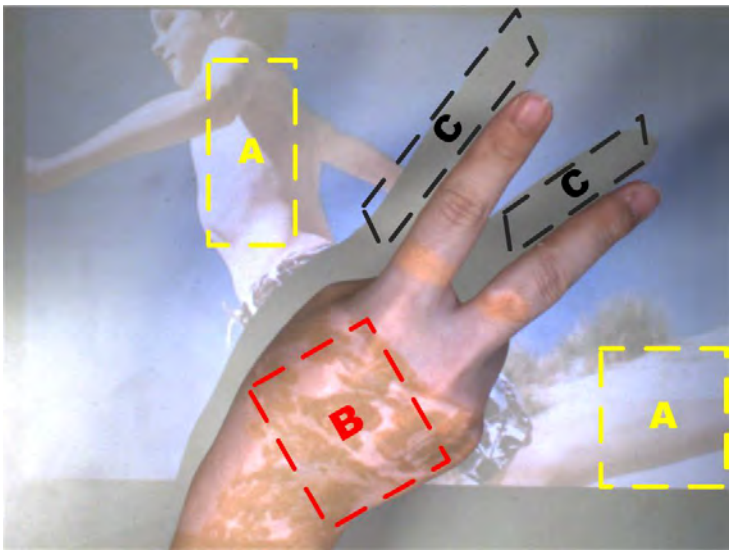


Figure 5.1: A sample hand image captured by projector-camera system.

In this chapter, we introduce a coarse-to-fine approach to solve the aforementioned problems. The main idea of the method is to combine contrast saliency map with mean-shift based smoothing and segmentation by a confidence function. Low-level contrast saliency detection enables the hand region to be highlighted roughly, and mean-shift based smoothing method removes the noises induced by projection contents without demolishing discontinuity information. Moreover, without any pre-training and pre-calibration procedures, the robust, precise and also rapid hand segmentation can be derived.

The rest of this chapter is organized as follows. Section 5.2 presents the proposed method. Experimental results in section 5.3 demonstrates the accuracy and efficiency of the proposed method. and Section 5.4 concludes this chapter.

5.2 Method

5.2.1 Rough Segmentation by Contrast Saliency

Although incessant varied video contents are projected to the projection surface and the hand operating above, it is obvious that the hand is always the most noticeable object from the human vision system's perspective. Motivated by this biological vision cue, firstly we employed a saliency detector to derive a rough hand region segmentation. Salient region detection as a typical low-level vision approach has been widely studied in computer vision society. According to our special projector-camera scenario, the saliency detector must satisfy the following requirements:

- Emphasizing the largest salient objects
- Uniformly highlighting whole salient regions
- Disregarding artifacts arising from projection content and ambient illumination
- Accomplishing detection less than $15ms$ for real-time requirement

After comparing different saliency detection methods [70, 105, 62, 67, 12, 54, 184], we chose the histogram-based contrast [31], which best fulfills the aforementioned criterions , to define the saliency values for image pixels.

The saliency of a pixel is defined using its color contrast to all other pixels in the image, i.e., the saliency value of a pixel I_k in image I is defined as

$$S(I_k) = \sum_{i=1}^{N} D(I_k, I_i), \tag{5.7}$$

where $D(I_k, I_i)$ is the color distance metric between pixels I_k and I_i in the HSV color space. It is clear that pixels with the same color value have the same saliency value under the definition, since the measure is oblivious to spatial relations. Hence, rearranging Eq. 5.7 such that the terms with the same color value c_j are grouped together, we get the saliency value for each color as,

$$S(I_k) = S(c_l) = \sum_{j=1}^{n} f_j D(c_l, c_j), \tag{5.8}$$

where c_l is the color value of pixel I_k, n is the number of distinct pixel colors, and f_j is the probability of pixel color c_j in image I.

In order to reduce the high dimension of 256^3 true-color space, more frequently emerging 85 colors were selected by building a compact color histogram using color quantization. At the same time, artifacts would be introduced. A smoothing procedure is used to refine the saliency value for each color, which replacing the saliency value of each color by the weighted average of the saliency value of similar colors. Typically, $m = n/4$ nearest color are chosen to refine the saliency value of color c by

$$S'(c) = \frac{1}{(m-1)T} \sum_{i=1}^{m} [T - D(c, c_i)] S(c_i), \tag{5.9}$$

where $T = \sum_{i=1}^{m} D(c, c_i)$ is the sum of distances between color c and its m nearest neighbors c_i, and the normalization factor comes from

$$\sum_{i=1}^{m} [T - D(c, c_i)] = (m-1)T. \tag{5.10}$$

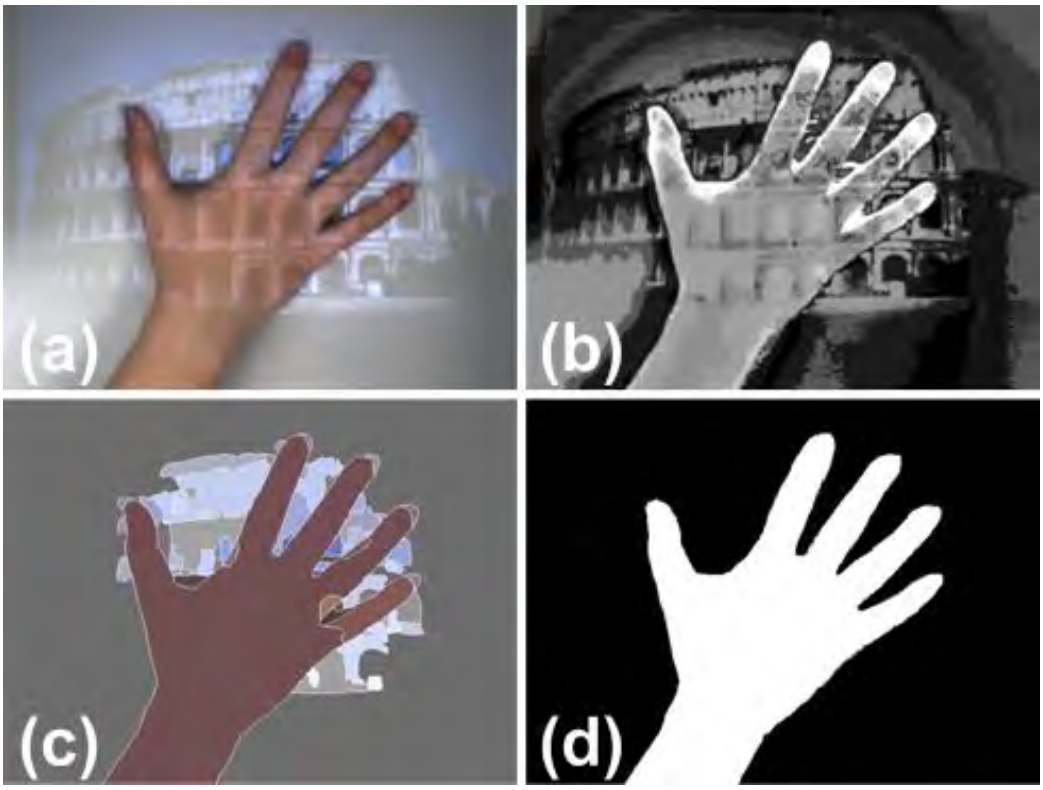


Figure 5.2: (a) Origin image; (b) histogram contrast salient map; (c) segments derived through mean-shift; (d) refined segmentation result.

More implementation issues are detailed in [31]. The saliency map $S(x, y)$ of image I (Fig. 5.2(a)) is derived as shown in Fig. 5.2(b).

5.2.2 Mean-Shift Region Smoothing

Even though the hand region has been highlighted through saliency detection, as illustrated in Fig. 5.2(b), it is not uniformly emphasized due to the influence of the projection content on the hand and projection surface. Hereby, it is impossible to have precise hand segmentation through traditional threshold methods. We employed mean-shift based smoothing and segmentation approach [32] in the salient regions, which not only eliminates the noises but also preserves the discontinu-

ity by adaptively reduce the mount of smoothing near abrupt changes in the local structure, i.e. boundaries.

Mean shift is a procedure for locating the maxima of a density function given discrete date sampled from that function. It is useful for detecting the modes of this density. This is an iterative method, and we start with an initial estimate x. Let a kernel function $K(x_i - x)$ be given. This function determines the weight of nearby points for re-estimation of the mean. Typically we use the Gaussian Kernel on the distance to the current estimate, $K(x_i - x) = e^{c||x_i - x||^2}$. The weighted mean of the density in the window determined by K is

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x) x_i}{\sum_{x_i \in N(x)} K(x_i - x)}, \tag{5.11}$$

where $N(x)$ is the neighborhood of x, a set of points for which $K(x) \neq 0$.

One important advantage of mean shift-based segmentation is its capability to resolve over-segment issue. The joint domain mean shift-based segmentation succeeds in over-coming the inherent limitations of methods based only on gray-level or color clustering which typically over-segment small gradient regions, which are common in projector illuminated area, due to the projector's nonlinearity and the variations of ambient illuminations.

Another important advantage of mean shift-based segmentation [32] is its modularity which makes the control of segmentation output very simple, just through three parameters: (h_s, h_r, M). The range parameter h_r and the smallest significant feature size M control the number of regions in the segmented piecewise constant model, larger values have to be used for h_r and M to discard the effect of small local variation. The spatial parameter h_s determines the size of spatial window. In our case, (h_s, h_r, M) is set to $(7, 10, 20)$.

It is worth mentioning that the inherent iterative property of mean-shift based

method usually invokes the efficiency problem. The rough salient region detection decreases the mean-shift search space which accelerates the convergence speed dramatically.

After mean-shift smoothing and segmentation, the image is divided into L candidate partitions P_k, $i = 1, \ldots, L$, as shown in Fig. 5.2(c). The contour of the hand is preserved well.

5.2.3 Precise Segmentation by Fusing

For the sake of acquiring precise hand region segmentation, we proposed a confidence function combining contrast saliency and region discontinuity to evaluate the probability of a candidate partition to be a part of hand region. The value of confidence function for each candidate partition is determined by several terms listed as follows:

- The average salient value of the pixels in the partition;
- The number of the neighbor partitions and the average salient value of neighbor partitions;
- The area of the partition;
- Whether the partition is on the image boundaries.

Hence, the value of confidence function $C_F(k)$ for partition P_k is formulated as

$$C_F(k) = \frac{1}{e^{(L-1)}}[\alpha\bar{S}(k) + \beta\bar{S}_N(k) + \gamma A(k)], \tag{5.12}$$

where $\bar{S}(k)$ is the average saliency value of the pixels in P_k, $\bar{S}_N(k)$ is average saliency value of its N neighbor partitions, and $A(k)$ is the partition's area. The three terms above are all scaled to $[0, 1]$. L is the number of image boundary to

which the partition attached, when $L \geq 2$, it is indicated that the partition belongs to background that should have low confidence value. The average weights are α, β, γ, when the number of the neighbor partitions N is equal to 1, $\beta = 1/2, \alpha = \gamma = 1/4$, which means that the confidence value is mostly depends on its surround neighborhood, if the partition is an isolated area in hand region or background region; Otherwise, $\alpha = 1/2, \beta = \gamma = 1/4$.

If $C_F(k)$ is greater than a pre-defined threshold Δ, the partition is considered as a part of hand region. Since not all skin pixels will be categorized correctly at all times, a morphological closing operation is employed in order to remove small noisy holes in the skin pixel areas. Hence, the refined binary segmentation is derived, as shown in Fig. 5.2 (d).

5.3 Experiments

The projector-camera system we used in our experiment consisted of a Pico DLP projector of resolution 640×480 and a CCD camera of resolution 648×488. The system was calibrated geometrically and radiometrically by method detailed in [29] for background subtraction method.

We collected a great diversity of images (e.g. flowers, buildings, celebrities, animals etc.) from Google Image [2] and projected them to a desk surface. An experimental dataset of 500 images was captured under different projection contents and different hand shapes. The ground-truth is manually annotated with the assistance of GrabCut [138]. Several test images with their ground-truth are shown in Fig. 5.3 (a) and 5.3 (b).

In order to illustrate the merits of proposed method, we conducted comparison experiments with some related methods. The choice of these methods is motivated by the following reasons: citation in literature (the classic approach of statistical

color model-based (SCM) method is widely cited [91]), precision (the background subtraction method (BkSub) has higher precision, since it is on the basis of using pre-calibrated geometric and radiometric information to predict the background image [99]), and recency (the sophisticated graph based method (GB) [46]).

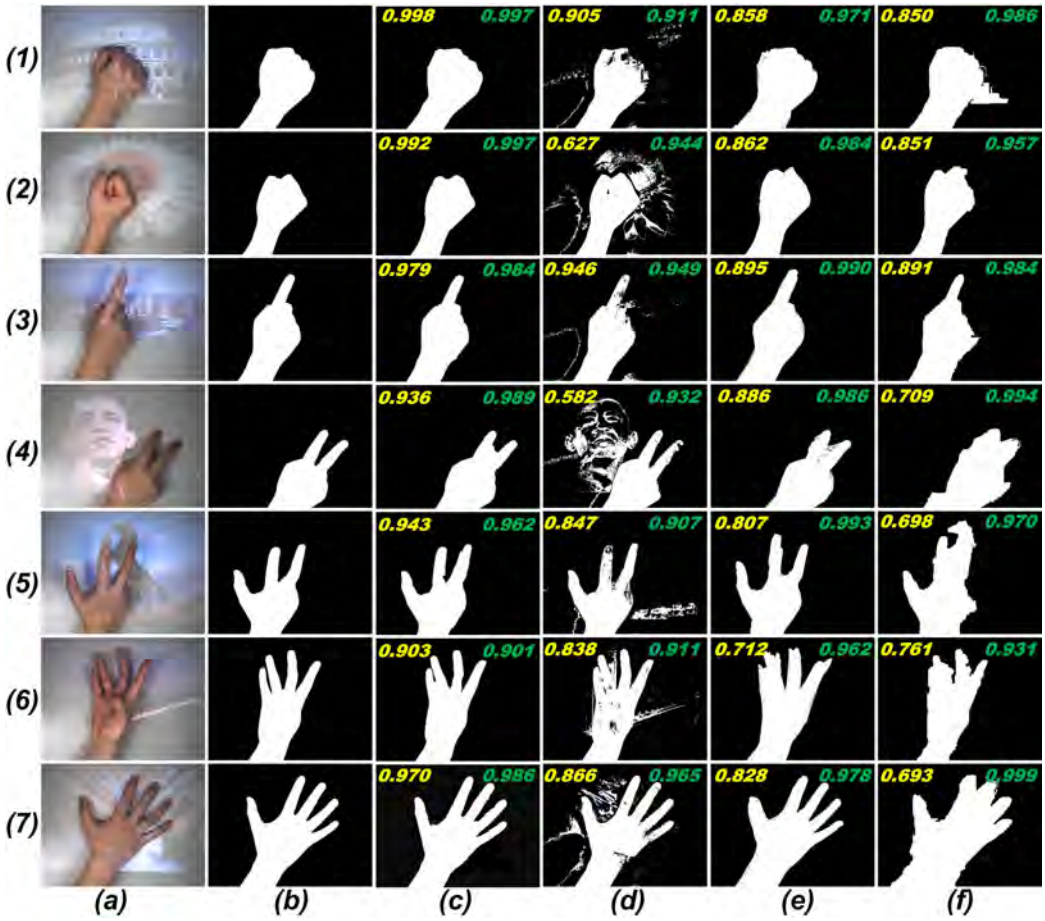


Figure 5.3: Visual comparison. (a) original image; (b) ground-truth; (c) our method; (d) SCM [91]; (e) BkSub [99]; (f) GB [46]. The yellow (top-left) and green (top-right) numbers in each result image are the corresponding precision p and recall r values, respectively.

As in [31], we adopted the F-beta score to evaluate the accuracy of segmenta-

tion, which considers both the precision p and the recall r to computer the score:

$$p = \frac{N_C}{N_R}, \tag{5.13}$$

$$r = \frac{N_C}{N_G}, \tag{5.14}$$

where N_C, N_R, N_G are the number of correct segmented pixels, all segmented pixels and ground-truth pixels respectively. The F-beta score is the harmonic mean of precision and recall, formulated as

$$F_\beta = (1 + \beta^2) \cdot \frac{\mathrm{p} \cdot \mathrm{r}}{(\beta^2 \cdot \mathrm{p}) + \mathrm{r}}, \tag{5.15}$$

where β is set to 0.3 to weight precision more than recall. The visual and quantitative comparisons are shown in Fig. 5.3 and 5.4 respectively. Among all the methods, our method shows the highest precision, recall and F_β values. It is evident that the skin color-based method (SCM) gets low precision when some projected objects have the color similar to skin, such as human face and yellow flower in the case of Fig. 5.3 (d2) and 5.3 (d4). The background subtraction method (BkSub) shows a high recall but poor precision, verifying that the shadow cast by video projection has great influence, as shown in Fig. 5.3 (e4) and 5.3 (e6). The graph-based method (GB) can not reserve smooth boundaries and confuse projected objects with hand region, which are the main reasons for low precision, as illustrated in Fig. 5.3 (f4-f7).

Table 5.1 compares the average processing time taken by each method. All the methods are implemented in C++ and executed on a desktop PC with Intel Core 2.8GHz CPU and 2GB RAM. Although our method is not the fastest one, it is sufficiently for real-time applications.

Table 5.1: Average processing time.

Method	Ours	SCM [91]	BkSub [99]	GB [46]
Time (ms/frame)	29.6	10.9	2.3	115.2

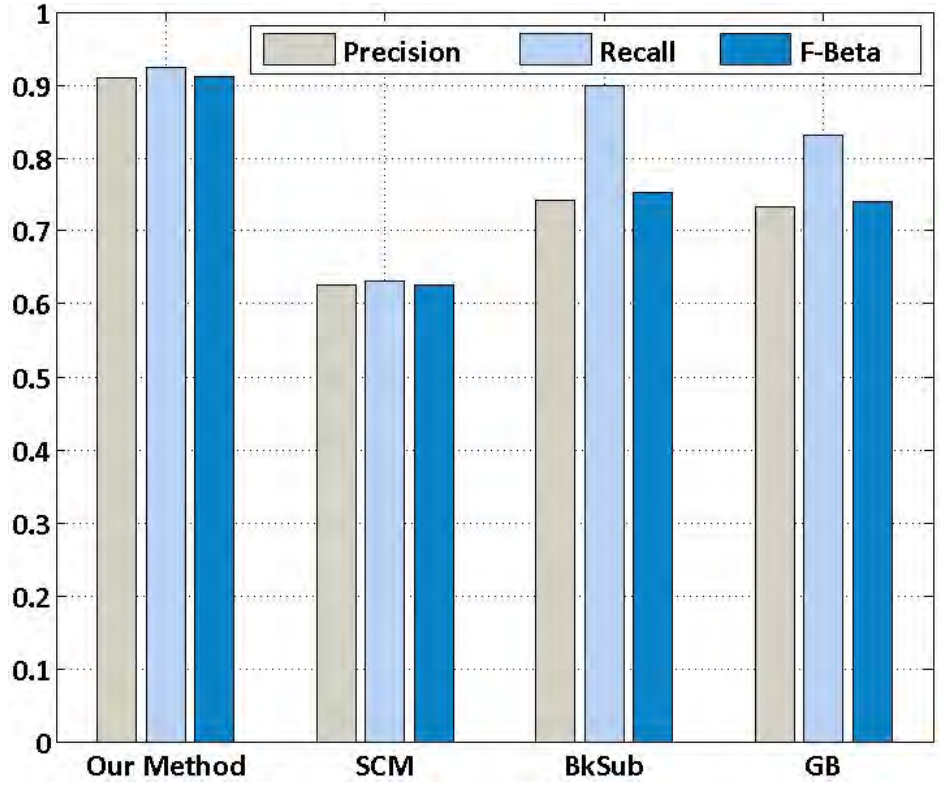


Figure 5.4: Precision-Recall bars for hand segmentation using different methods. Our method shows high precision, recall and F_β values.

5.4 Summary

In this chapter, we described a novel coarse-to-fine approach for hand segmentation in projector-camera system, which combining the contrast saliency and region discontinuity through a confidence function. The experimental results prove that the proposed method can segment hand region accurately and rapidly, even the captured images are interfered by successively projection contents and the shadow cast by moving hand.

Chapter 6

Touch-Sensitive Display on Arbitrary Planar Surface

In this chapter, we address how an HCI (Human-Computer Interface) with small device size, large display, and touch input facility can be made possible by a mere projector and camera. The realization is through the use of a properly embedded structured light sensing scheme that enables a regular light-colored table surface to serve the dual roles of both a projection screen and a touch-sensitive display surface. A random binary pattern is employed to code structured light in pixel accuracy, which is embedded into the regular projection display in a way that the user perceives only regular display but not the structured pattern hidden in the display. With the projection display on the table surface being imaged by a camera, the observed image data, plus the known projection content, can work together to probe the 3D world immediately above the table surface, like deciding if there is a finger present and if the finger touches the table surface, and if so at what position on the table surface the finger tip makes the contact. All the

decisions hinge upon a careful calibration of the projector-camera-table surface system, intelligent segmentation of the hand in the image data, and exploitation of the homography mapping existing between the projector's display panel and the camera's image plane. Extensive experimentation including evaluation of the display quality, touch detection accuracy, trajectory tracking accuracy, multi-touch capability and system efficiency are shown to illustrate the feasibility of the proposed realization.

6.1 Introduction

HCI (Human-Computer Interface) has been traversing from firstly punch card and LEDs, then paper tape and CRO display, more recently mouse-plus-keyboard and LCD panel, and now fingers and touch-sensitive display panel over the history of development. Technologies have been ever improving, with the data-input mechanism growing only more natural, and the display only more vivid. Indeed for the input-output interface of computers, scarcely anything could be more natural than using our fingers to drag items on the "virtual desktop" of the computer, to open (and move and copy) files and folders, and to scroll (and enlarge) pages.

In today's computers and other portable devices like cellular phones and P-DAs, a large display panel is desired not only for enhancing display quality and coping with say aged vision, it is also essential, for touch input interface, for allowing finger - a rather bulky pointing device - to specify position on the "virtual" desktop in adequate precision. On that there is the following dilemma. A bigger and higher-resolution display, and a bigger keyboard, are desired to incur less strain on eyes and fingers. Yet they also make the devices less portable. This article attempts to solve this dilemma by exploring the possibility of replacing the display panel and the mouse-and-keyboard by a mere projector and camera. Specifically,

it is to enable a light-colored table surface, to which the projection is illuminated, to serve as a touch-sensitive display panel for finger-based user input. The use of a projector in place of an LCD panel would dissociate display size from device size, making portability much less an issue. Touch-sensitive input facility on such a large display would also alleviate the need of a large keyboard.

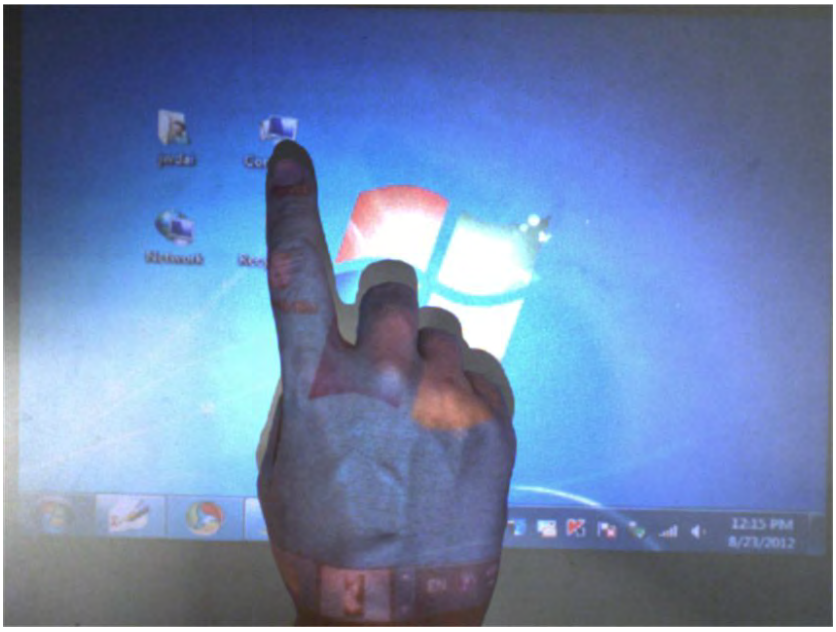


Figure 6.1: Single view of the projector illuminated table surface.

The challenge is, from a single image alone (as illustrated in Fig. 6.1) there is generally difficulty in even distinguishing whether there is a physical contact between the finger and the table surface. The facility of acquiring certain 3D information about the illuminated workspace would be of much aid. A desirable way of making that possible is to use no additional sensor or instrument beyond what are already there - the projector and camera - by embedding structured codes into the projection. This way, the projector serves two purposes: the display device, as well as the 3D acquisition channel.

This chapter aims at building the stated system, letting any tabletop surface

to which the projection is illuminated become a touch-sensitive computer screen, with the entire system requiring a mere video projector and camera.

6.2 Previous Works

Traditional human-computer interface is largely mouse-and-keyboard based, which is effective but not necessarily the most natural. Tangible interfaces have been used in some projected environments. By letting users hold some physical objects in hand and manipulate them, more comfortability could be induced in the interaction. Sensetable [128] uses a projected interface for visualization and design. Physical objects with embedded sensors can be held by users for movements to represent the corresponding interactions. The Flatland system [120] projects onto a whiteboard, and interactions are based on the interpretation of strokes by the stylus on the whiteboard. More recently, Escritoire [15] uses special pens with embedded sensors to enable interaction between a user and an illuminated table surface. These applications are all based on manipulating tangible objects like pens for interactions. The flexibility can however be further improved if even the intermediate objects can be waived, and hands and fingers are directly used. Barehand interface enjoys higher flexibility and more natural interaction than tangible interfaces.

Earlier researches on barehand interfaces demanded assistance from some additional sensors. The interfaces in DiamondTouch [38] and SmartSkin [137] both allow hand input on a table surface, but the table has to embed a grid of wired sensors in the first place. Another interface called Light Touch [3] also demands the use of special hardware. It requires the use of a special projector - laser projector - not LED projector that is the nominal one in the consumable market. Laser projector could have stronger luminance and shorter projection distance, but the

projection quality is not in the same par, and the projection technology is generally far less mature than that of the LED projectors. It is also costly. In addition, an infra-red sensor is needed in LightTouch to recognize finger's contact with the projection surface. In Skininput [64], the location of finger taps on the arm and hand was resolved by analyzing mechanical vibrations that propagate through the body. To collect these signals, an array of sensors worn as an armband was essential.

With the development of computer vision algorithms, some vision-based projected tabletop interfaces equipped with finger tracking began to emerge in the last few years. Letessier [98] employed a single camera to detect and track the 2D position of the tip of bare finger on a planar display surface, but neglected finger clicking detection. In [169, 94], the "click" event was determined through a delay-based scheme, which has limited usability in applications that require fast response and multiple same-button clicks. Moreover, such click events were not intuitive and were rather deliberate since the user had to hold his finger over the button for a stipulated period to register a button select. Marshall [109] detected touch from the change in color of the fingernail when the finger was pressed against a surface. Song [152] proposed a finger-based interface in a projector-camera setting by examining if the finger and its shadow in the image were separated or merged. Wilson's PlayAnywhere [174] adopted extra infrared illumination to enhance the contrast between the finger and non-finger regions of the image data. This scheme however demands a capability of distinguishing the finger from its shadow robustly in the image. There is also substantial challenge in extending the scheme to multi-touch interface. Fitriani [47] projected a button based interface onto the surface of a soft deformable object such as a sofa pillow. The appearance changes of the virtual button being pressed were observed by a camera, which was considered as a signal for the touch event. The error detection

rate was high due to complex and unpredictable deformations of the deformable surface.

After the release of PrimeSense's [6] depth-sensing camera-based Microsoft Kinect [4], depth-sensing cameras have been used in various interactive surface applications. LightSpace [176] used an array of depth-sensing cameras to track users's manipulations on multiple surfaces. In [175], the touch event was determined by using a per-pixel depth threshold derived from a histogram of the static scene. Omnitouch [63] detected surface touch by counting the pixel number in a flood filling operation in depth map. Yet depth-sensing camera is rather bulky, and is not a standard device as compared to pico-projector and CCD camera. All these hinders its applicability in hand-held consumer electronic products.

This chapter aims at making the following contributions in building a touch-sensitive device:

1. *Using only off-the-shelf devices*

   Pocket DCs and cellular phones with built-in projector and camera have already emerged in the consumable market. They form the necessary pro-cam foundation in building touch-sensitive interface in handheld devices.

2. *Achieving 3D sensing without explicit 3D reconstruction*

   Detecting if a finger has indeed touched a tabletop surface and deciding at which position of the surface the touch takes place is a 3D sensing problem. Yet our system achieve all these without the need of going through explicit 3D reconstruction. The system exploits merely the homography mapping (induced by the table surface) between the projector's display panel and the camera's image plane. Without going through explicit depth recovery, the complexity of the sensing task is much reduced.

3. *Use of prior knowledge to enhance robustness*

By exploiting prior knowledge say about the relative geometry of the projector, camera, and projection surface, the system is endowed with better adaptability to environmental variations.

The remainder of this chapter is structured as follows. In the next section, prior knowledge embraced in the pro-cam system is reviewed. In Section 6.4, the principle and strategy of embedding structured light codes in an invisible way into regular projection is described. The essential processes of the proposed method including hand segmentation, fingertip detection, and touch detection are detailed in Section 6.5. In Section 6.6, the system setup and experimental results are shown. Conclusion and possible future work are offered in Section 6.7.

6.3 Priors in Pro-Cam System

Consider a Pro-Cam system that has a projector illuminating certain display pattern to a planar projection surface (e.g. a tabletop surface) that is imaged by a camera. Once the two electronic instruments' intrinsic parameters and extrinsic relationship relative to the projection surface are fixed, the image data about the projection surface are predictable from the projection content. Specifically, which image position carries which part of the projection content that is reflected by the projection surface is governed by a particular homography mapping [131] existing between the projector's display panel Π_P and the camera's image plane Π_C, which is induced by the projection surface Π_T; and how close color or gray level in the image resembles that of the original projection content is governed by a radiometric process that can be calibrated. In this work, we make use of such priors for enhancing the efficiency and precision of the human-computer interface we aim at building.

6.3.1 Homography Estimation

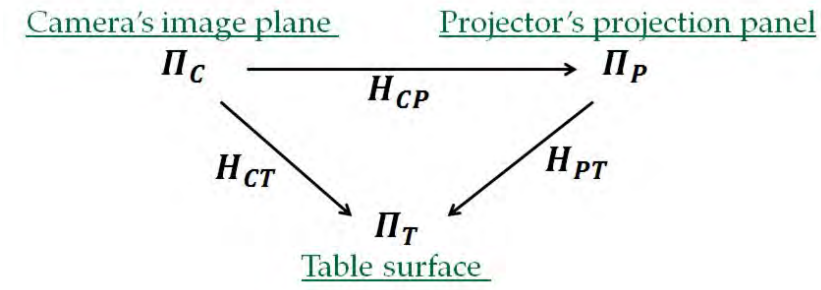


Figure 6.2: Homographies in projector-camera-surface system.

As illustrated in Fig. 6.2, there are altogether three homographies in our system: the homography H_{TC} between the camera's image plane Π_C and table surface Π_T, the homography H_{PT} between the projector's display panel Π_P and Π_T, and the homography H_{CP} between Π_C and Π_P that is induced by the table surface. Among them, H_{PT} is used for projector keystone correction, H_{CP} is for retrieval of the structured light code, and H_{CT} is for deriving H_{PT} which cannot be directly calibrated for the reason that projector does not have visual sensing capability.

Since homography can be expressed as a 3×3 matrix of arbitrary scale, i.e., a matrix with 8 degrees of freedoms (DOFs), it could be determined from as few as four pixel correspondences only across the input and output planes; when more than four correspondences are available, least-squares solution of the homography is to be obtained.

Firstly, the homography H_{TC} between the camera's image plane and the table surface is determined. On this, any rectangular object of known or standard dimension (e.g. credit card, plastic ruler) placed on the projection surface can be

used as the calibration object. The H_{TC} could be estimated as

$$X_T = H_{TC}X_C, \tag{6.1}$$

where X_T is any corner of the flat reference object in homogenous coordinates, and X_C is the corresponding point on the camera's image plane.

With H_{CT}, the homography H_{CP} between the camera and projector could be derived with ease. By instructing the projector to project some distinct markers (e.g. chessboard) to the table surface, the homography could be calculated in the same way as the above:

$$X_C = H_{CP}X_P, \tag{6.2}$$

where X_C is the position of projected marker in the observed image, and X_P is the marker position on the display panel of the projector, both in homogeneous coordinates.

Finally, the homography H_{TP} between projector and table surface is determined as

$$X_T = H_{TC}X_C = H_{TC}H_{CP}X_P = H_{TP}X_P. \tag{6.3}$$

6.3.2 Radiometric Prediction

Besides the geometric distortion, the photometric appearance (e.g. brightness, RGB color etc.) of the projection surface in the image data is another prior that has to be seized before it can be exploited in the construction of the touch interface system. The appearance is generally distorted from that of the projected pattern due to nonlinearity of the projection and imaging processes, the texture of the projection surface, and the influence of ambient illumination. To predict the projection appearance in the image data, radiometric calibration is necessary.

Here we employed the photometric model described in [30], which is formulated as

$$\mathbf{C}_{pre} = \mathbf{VP} + \mathbf{F}, \tag{6.4}$$

where $\mathbf{C}_{pre}$ and $\mathbf{P}$ are the RGB values of the predicted image and real image respectively, the 3×3 matrix $\mathbf{V}$ is the color mixing matrix that captures all the couplings between the projector and camera channels and their interaction with the spectral reflectance of the projection surface, and the vector $\mathbf{F}$ is the contribution of the environmental lighting relative to the black level of the projector.

To measure these parameters, five images are projected and captured, first a black image, then a red, a green, a blue and a chromatic in sequence. In addition to the one image projection and two image captures required in homography estimation, the process of deriving all the priors involves 7 projection-capture cycles, which can be accomplished in only a few seconds. Unless the system is moved to another working environment, or the environmental illumination is changed, the prior knowledge is approximately constant in the operation of the touch interface system.

In our current system, only the geometrical priors, the homographies, are employed for keystone correction and touch action recognition. So in the initialization stage, only one image projection and two image captures are required.

6.4 Embedding Codes into Video Projection

6.4.1 Imperceptible Structured Light

The basic principle of imperceptible structured light is described in Section 4.3.1. For detailed information, please refer to that section.

6.4.2 Embedded Pattern Design Strategy and Statistical Analysis

Structured light coding is about equipping each pattern position a unique code that can be distinguished in the image data. The coding can be realized over time or space (the 2D space of the code pattern). In the touch sensitive interface we are to build, the movement of hand and finger, the real-time operation requirement, and the constraints of imperceptible code embedding make the temporal coding scheme not applicable. We are thus left with the option of using spatial coding scheme, which has the advantage that 3D determination can be achieved with at few as one single image.

Since the resolution, optical parameters, and the position and orientation with respect to the target object are all different between the camera and projector, it is impossible to align the pixels on the camera's image plane and those on the projector's display panel for one-to-one pixel correspondence. To overcome the problem, binary spatial coding methods generally adopt some special shapes (such as stripe, square, circle etc.) as appearance profiles, which could be easily segmented in the decoding stage. A shortcoming of this design scheme is that the density of the effective feature points is sparse, and in our case is generally too sparse to ensure that the depth information of the fingertip can always be derived no matter where it is located. So here we proposed a new binary encoding scheme that has pixel precision.

Almost all the spatial coding methods were based on perfect map or M-array theory for its unique window property. MacWilliams [106] and Etzion [44] proposed methods to construct M-array mathematically. By folding pseudorandom array, the methods are effective and efficient to generate M-arrays. However, they could only generate the ones of $n_1 \times n_2$ size with the $k_1 \times k_2$ window property,

where $n = 2^{k_1 k_2} - 1$, $n_1 = 2^{k1} - 1$, $n_2 = n/n_1$. In our case, the resolution of pico projector is 640×480, in order to make sure that every pixel has a unique binary code, $2^k \geq 640 \times 480$, so that $k \geq \ln(640 \times 480)/\ln 2 \geq 18.23$. Thus the windows size should be set as 5×5. Therefore, the dimension of perfect map is $n_1 = 2^{k_1} - 1 = 31$, $n_2 = n/n_1 = 1082401$. However, this result is not applicable for our projector.

Some researchers employed other practical methods to generate the perfect map. Morano [115] proposed an algorithm for constructing an M-array, fixing the length of the alphabet, the window property size, the dimensions of the array and the Hamming distance between every window. The algorithm used to generate an array with fixed properties is based on a brute force approach. For our case, when constructing a binary M-array with window property of 5×5, the following steps are taken: first, a sub-array of 5×5 is chosen randomly and is placed in the top-left corner of the M-array that is being built. Then consecutive random columns of 5×1 are added to the right of this initial sub-array, maintaining the integrity of the window property of the array. Afterwards, rows of 1×5 are added beneath the initial sub-array in a similar way. Then, both horizontal and vertical processes are repeated by incrementing the starting coordinates by one, until the whole array is filled. When filling the array, the code uniqueness of the new added point will be checked. If not satisfied, the array is cleared and the algorithm starts again. Since the computational complexity is extremely high, the author only generated an array of 45×45. Besides the three aforementioned methods, some other typical methods in binary spatial coding are listed in Table 6.1. In the literature there is not an effective method to generate a binary array of 640×480 size that has the required unique window property. For this reason, in this work we seek to generate the pattern array by statistical analysis.

In our system, we use a pico projector that is of 640×480 resolution. To make

Table 6.1: Summary of typical spatial coding methods

Method	Array Size	Win. Size	Alph. Length
Morita [117]	24×24	3×4	2
Kiyasu [93]	18×18	4×2	2
Salvi [146]	29×29	3×3	3
Spoelder [59]	65×63	2×3	2
Albitar [13]	27×29	3×3	3
Desjardins [39]	53×38	3×3	3
Chen [28]	82×82	3×3	7

sure that every pixel has a unique binary code, it is required that $2^k \geq 640 \times 480$, which means $k \geq \ln(640 \times 480) \geq \ln 2 \geq 18.23$. In other words, the codeword at each pattern position must be at least 18 bits long. In accordance with the resolution of pico projector, a matrix of 640×480 is to be filled with pseudo-random generated sequence consisting of 0 and 1 in standard uniform distribution. If an $m \times n$ window is selected for coding each pixel, and if the window is picked to be the one with the pixel as its bottom-right corner, totally $(640 - m + 1) \times (480 - n + 1)$ pixels will be coded by a (mn)-bit binary string. The codeword of every effective pixel can be derived and some statistical analysis can be employed to evaluate the code uniqueness. For our pico projector, random generation of 6×6 arrays are generally sufficient to equip each pixel with a unique window label.

In our experimentation, after conducting 100 trials of pattern generation, the array with the largest average inter-codeword Hamming distance ($\bar{H} = 4.524$) was derived. The large inter-codeword Hamming distance corresponds to good noise-tolerance of the codewords on the imaging side. We chose this array (part of which is shown in Fig. 6.3) to embed into normal video projection.

In the decoding stage, the correspondences between the camera's image plane and the projector's display panel were established by the homography induced by

Figure 6.3: Magnified part of the binary pattern (dotted line grid is added for illustration)

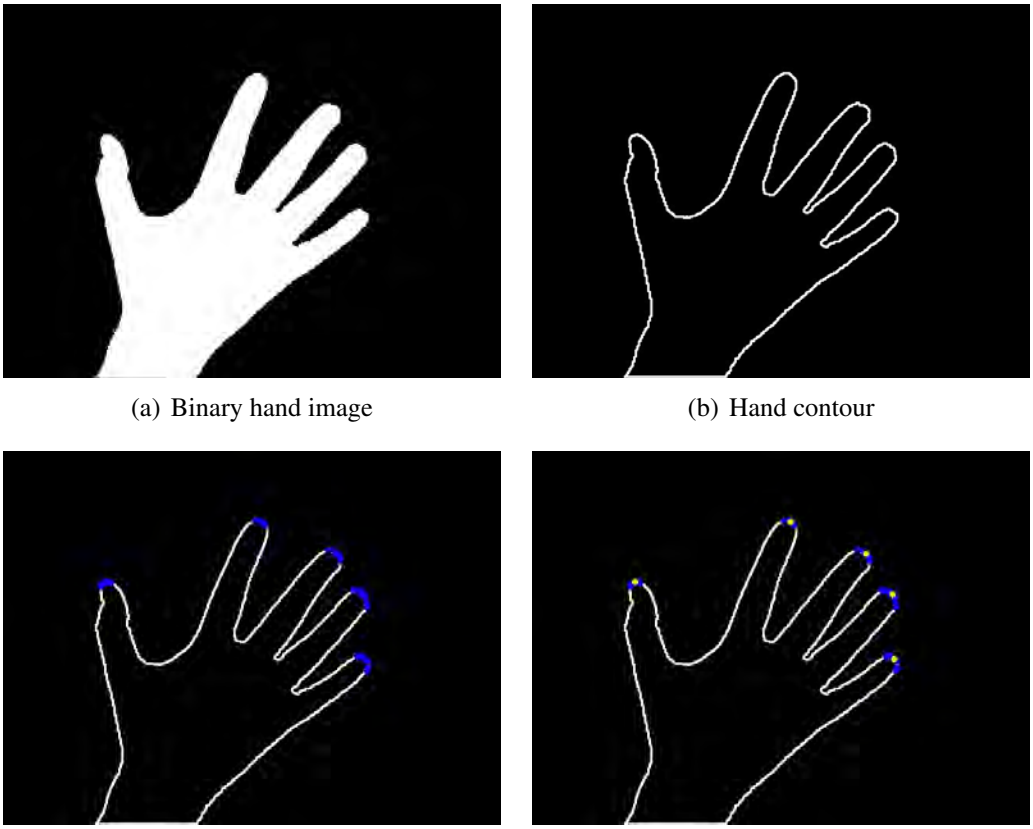
the projection surface. This will be discussed in the following section in depth.

6.5 Touch Detection using Homography and Embedded Code

For the purpose of locating the position of the fingertip and determining whether a physical touch takes place, some preliminary processes need be employed, such as hand segmentation and fingertip detection. In this section we discuss these processes in the circumstance of our particular pro-cam system.

6.5.1 Hand Segmentation

By making use of the hand segmentation method proposed in Chapter , the hand operating above the projection area is segmented. After filling up the isolated small cavities by the use of the morphological ”close” operation, the largest connected subregion is regarded as the hand, as shown in Fig. 6.4(a).



(a) Binary hand image (b) Hand contour

(c) Fingertip candidates (d) Detected fingertips

Figure 6.4: Hand segmentation and fingertip detection.

6.5.2 Fingertip Detection

Fingertip detection is conducted on the basis of segmented binary hand image. The hand contour is retrieved from the binary image using the algorithm detailed

in [155], as illustrated in Fig. 6.4(b). The extracted contour serves to offer fingertip candidates through a simple arc line analysis. Let $\mathbf{T}(x), x = 1, \ldots, N$ be the various points of the hand silhouette in clockwise order, where N is the total number of contour points. Whether a particular contour point $T(k)$ is a fingertip candidate is examined by the curvature of the contour there. We express the curvature approximately as the angle θ,

$$\theta = \arccos \frac{\mathbf{v_1} \cdot \mathbf{v_2}}{\|\mathbf{v_1}\|\|\mathbf{v_2}\|}, \tag{6.5}$$

$$\mathbf{v_1} = \mathbf{T}(k) - \mathbf{T}(k-t), \tag{6.6}$$

$$\mathbf{v_2} = \mathbf{T}(k) - \mathbf{T}(k+t), \tag{6.7}$$

where $\mathbf{T}(k-t)$ and $\mathbf{T}(k+t)$ are contour points in the vicinity of $\mathbf{T}(k)$, each on a different side of $\mathbf{T}(k)$ at an interval of t points from it.

If $\theta < \frac{\pi}{2}$ and $|\mathbf{v_1}, \mathbf{v_2}| > 0$, $\mathbf{T}(k)$ is regarded as a fingertip candidate. The second conditional term as a determinant is employed to distinguish fingertip peaks from valleys between two fingers. Some fingertip candidates, that indicated by blue points in Fig. 6.4(c) are determined. Finally, the candidates that are consecutive or nearly consecutive in the hand silhouette are clustered into the same group, and in each group only the candidate in the median position is confirmed as a fingertip (yellow points in Fig. 6.4(d)).

6.5.3 Touch Detection Through Homography

With the fingertips detected, the next task is to examine if any of the fingertips touches the display surface. In the coding design, we ensure that every pixel in the projected pattern is coded by a 36-bit binary codeword. However, as discussed above, it is impracticable to align the pixels on the camera's image plane and those

on the projector's display panel for one-to-one pixel correspondence between the two. Instead, we make use of the homography between the image plane and display panel that is induced by the table surface. Below we use the single-touch case as an example to illustrate how a mere touch is detected. Multi-touch is a simple extension of the single-touch.

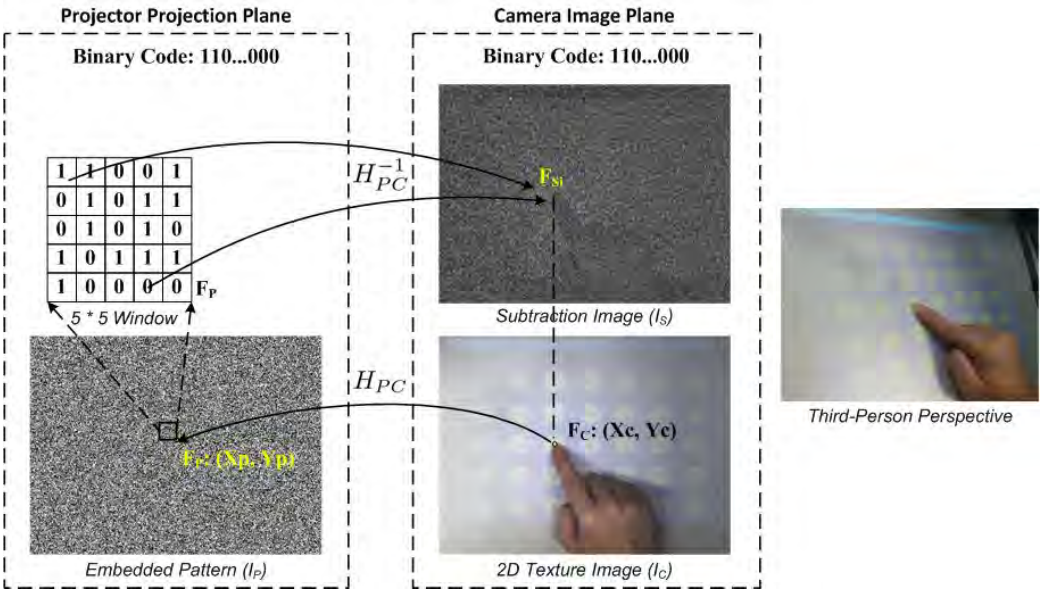


Figure 6.5: Touch detection via homography.

As illustrated in Fig. 6.5, suppose we have a finger touching the projection surface. The fingertip F_C lies on the plane of the projection surface, and thus would satisfy the associated homography. More precisely, a position F_P on the display panel of the projector Π_P can be derived in homogenous coordinates as $\tilde{F_P} = H_{PC}\tilde{F_C}$. The codeword at F_P is then determined by the code values of the pixels F_{P_i} in a 6×6 window that has F_P as its bottom-right corner. In other words, the binary codeword BC_P at F_P is regarded as

$$BC_P = \sum_{i=0}^{35} 2^i \cdot I_P(F_{P_i}), \tag{6.8}$$

where $F_{P_i} \in \{(X_{P_i}, Y_{P_i}) | X_P - 5 \leq X_{P_i} \leq X_P, Y_P - 5 \leq Y_{P_i} \leq Y_P\}$.

On the other hand, the binary code embedded in the image data at point F_C can be observed as

$$BC_S = \sum_{i=0}^{35} 2^i \cdot I_S(F_{S_i}), \tag{6.9}$$

$$\tilde{F_{S_i}} = H_{PC}^{-1} \tilde{F_{P_i}}, \tag{6.10}$$

where $\tilde{F_{S_i}}$ and $\tilde{F_{P_i}}$ are homogenous representations.

If the Hamming distance between BC_P and BC_S is less than a preset threshold λ_H, F_P and F_S are considered as sharing the same code, meaning that the touch has taken place. Otherwise, the finger is regarded as not having physical contact with the table surface. The threshold λ_H should be adjusted according to the ambient illuminations for suitable noise-tolerance.

The above allows touch to be determined without going through explicit 3D reconstruction, and can operate in real-time.

6.5.4 From Resistive Touching to Capacitive Touching

In the last section, we have emulated an "resistive" touch operation, which requires touching with a certain pressure on the projection surface. Below we show how to enhance the touch sensitivity and move the interface from a "resistive touch" to a "capacitive touch".

In fact we can generate from the table surface-induced homography to another homography that is induced by a plane parallel to but slightly elevated from the table surface, as indicated by any of the shown dashed lines in Fig. 6.6). The dash lines correspond to different levels of touch sensitivity demanded. If the homography so generated is satisfied by any detected finger tip in the image data,

a touch action can regarded as confirmed.

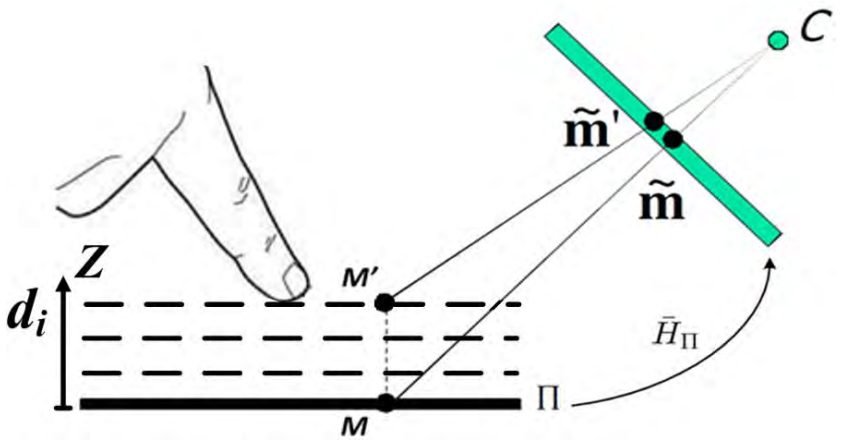


Figure 6.6: Homography transfer across parallel planes.

As shown in Fig. 6.6, given a plane Π, we can define a coordinate frame $W : X - Y - Z$ local to it, with X, Y axes within the plane Π and Z-axis perpendicular to Π. Suppose the plane Π is real table surface, and we know the homography $\bar{H}_\Pi$ from Π to the camera's image plane, that is induced by Π itself. Then let the pre-calibrated projection matrix of the camera be

$$P \cong [p_1, p_2, p_3, p_4] \cong K[r_{1\Pi}, r_{2\Pi}, r_{3\Pi}, t_\Pi], \tag{6.11}$$

where K is the 3×3 matrix containing all the intrinsic parameters of the camera. Notice that the homography $\bar{H}_\Pi$ that owns the property:

$$\tilde{m} \cong \bar{H}_\Pi [X, Y, 1]^T, \tag{6.12}$$

is related to the camera projection matrix by $\bar{H}_\Pi \cong [p_1, p_2, p_4] \cong K[r_{1\Pi}, r_{2\Pi}, t_\Pi]$.

Suppose we have a plane Π_{d_i} parallel to but elevated from Π by a perpendicular distance d_i. For the 3D position (X, Y, d_i) on Π_{d_i}, which is elevated from point

$(X, Y, 0)$ on Π perpendicularly by distance d_i, the image projection $\tilde{m}'$ can be expressed as

$$\begin{aligned}
\tilde{m}' &\cong K[R_\Pi, t_\Pi][X, Y, d_i, 1]^T \\
&\cong K(Xr_{1\Pi} + Yr_{2\Pi} + d_i r_{3\Pi} + t_\Pi) \\
&\cong K([r_{1\Pi}, r_{2\Pi}, t_\Pi] + d_i[0, 0, r_{3\Pi}])[X, Y, 1]^T \\
&\cong (\bar{H}_\Pi + d_i[0, 0, Kr_{3\Pi}])[X, Y, 1]^T, \qquad (6.13)
\end{aligned}$$

By substituting Eq. 6.12 into Eq. 6.13, we have

$$\tilde{m}' \cong (I + d_i[0, 0, p_3]\bar{H}_\Pi^{-1})\tilde{m} \cong H_{Cd_i}\tilde{m}. \qquad (6.14)$$

Hence, through the original homography and the third column of the camera projection matrix, we can derive the homography H_{Cd_i} between the camera's image plane and the elevated plane. In a similar way, the homography H_{Pd_i} between projector's display panel and the elevated plane can also be expressed. Finally, the new homography between the projector's display panel and the camera's image plane, that is induced by elevated plane, is obtained as $H_{CPd_i} = H_{Cd_i} H_{CP} H_{Pd_i}^{-1}$, which can be adopted for more sensitive touch sensing on the table surface.

6.6 Experiments

In order to assess the feasibility of the described system for barehand human-computer interface, we conducted experiments to evaluate display quality, touch detection accuracy, trajectory tracking accuracy, multi-touch capability, and system efficiency respectively.

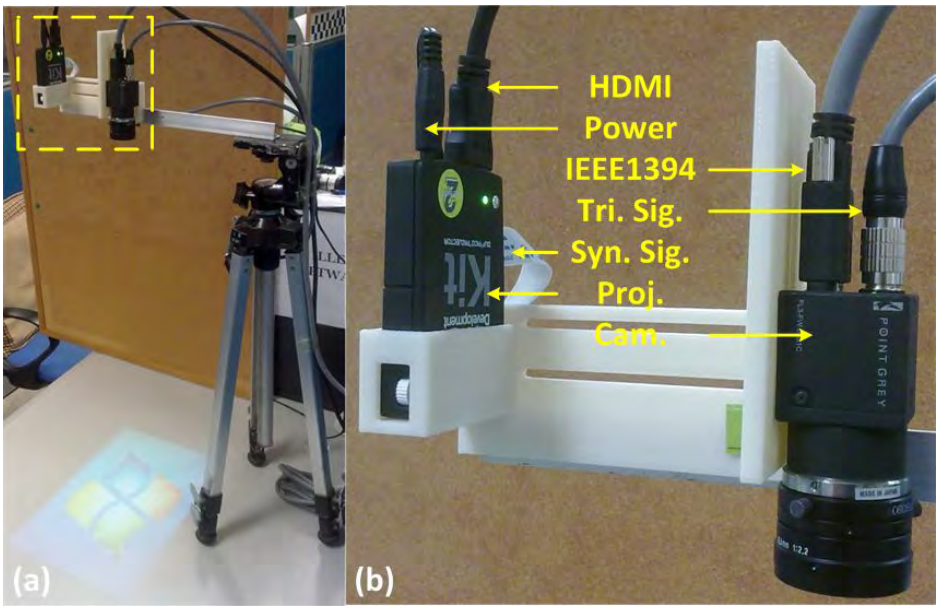


Figure 6.7: System prototype.

The projector-camera system we used in our experiment consisted of a DLP projector with a native resolution of 640×480 and an interface for firmware configuration (TI DLP Pico Projector Development Kit 2), plus a camera of 648×488 resolution at 120fps (Point Grey FL3-FW-03S1C camera with Myutron FV0622 f6mm lens), both being off-the-shelf equipments. The system was configured for a working distance of about $500mm$, making a 15-inch projection area. If short-throw projector and short focus lens are employed, a bigger projection area could be acquired with shorter distance.

We first mounted the projector and a camera rigidly and then fixed them on a tripod standing on a table surface, as shown in Fig. 6.7(a). The projector and camera were connected to a desktop computer through HDMI and IEEE1394 interfaces respectively, and the hardware trigger signal of the camera was connected to the sync. output of the projector for synchronization between them, which are illustrated in Fig. 6.7(b). Moreover, the projector-camera system was pre-calibrated using the method detailed in [154].

6.6.1 System Initialization

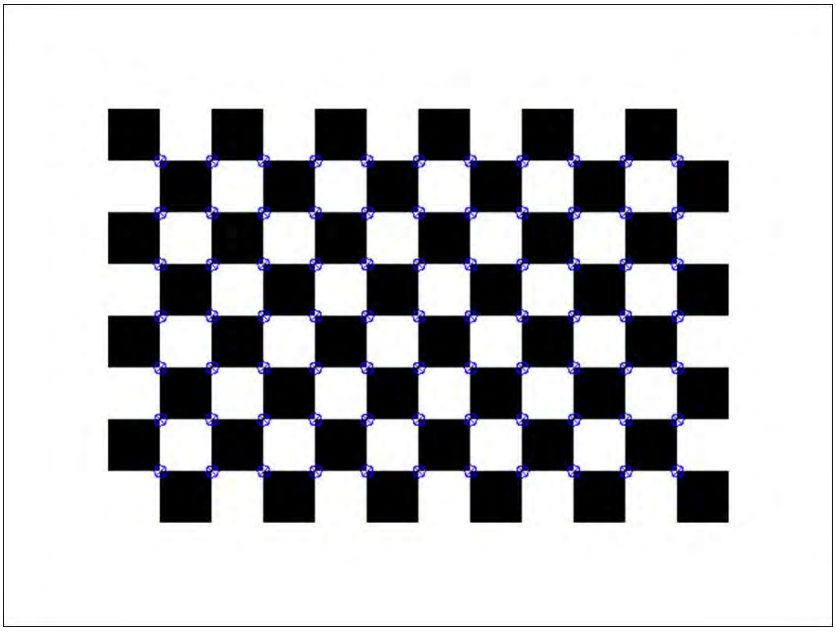
For the camera-projector-plane system, projection keystone correction is accomplished by the homography between projector projection plane and table surface; the finger touch action is determined through the homography between camera image plane and projector projection plane embraced by the planar table surface. Therefore, before the system operation stage, the initialization step to estimate the camera-plane and camera-projector homoraphies is necessary.

Camera-Projector Homography Estimation

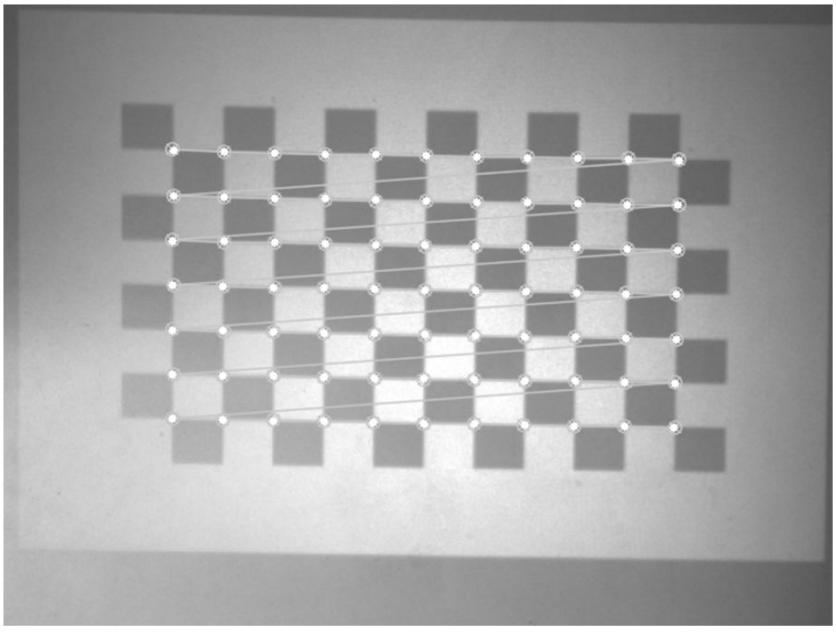
To estimate the camera-projector homography, one projection-capture cycle is needed. As shown in Fig. 6.8(a), a chessboard pattern was projected onto the table surface, the chessboard corners $CP_i(i = 0, \ldots, N)$ indicated by the blue circle were considered as the feature points, the coordinate of these points were known during the chessboard generation. After the camera's capture, an image of the table surface illuminated by the chessboard was acquired. Then through the automatic corner detection in the image data, the corresponding points $CC_i(i = 0, \ldots, N)$ were found, as indicated by the white dots in Fig. 6.8(a). Finally, by use of $CP_i \sim CC_i$ correspondences, the camera-projector homography H_{CP} can be calculated by least-square method.

Camera-Plane Homography Estimation

Since the lack of sensing capability of projector, it is impossible to estimate the projector-plane homography directly. Here, we have already obtained camera-projector homography, through estimating camera-plane homography, the projector-plane homography will be derived. For the sake of estimating camera-plane homography, an planar object with standard dimension is required. As illustrated in



(a) Projected chessboard



(b) Captured image

Figure 6.8: Images for camera-projector homography estimation.

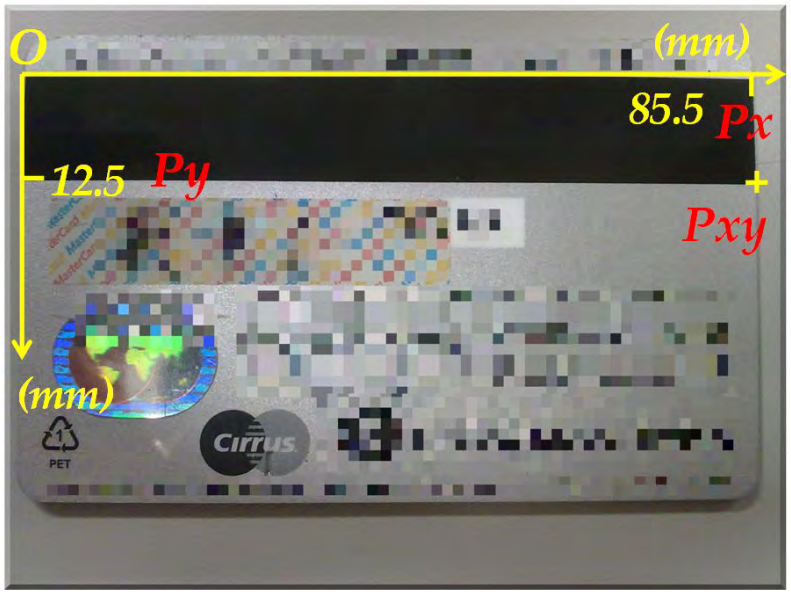
Fig. 6.9(a), a credit card was employed as calibration object, the black magnetic stripe on it is a rectangle with standard dimension of $85.5mm \times 12.5mm(W \times H)$. The top-left corner was chosen as the origin of the coordinate system of the card, so the coordinates of the four corners O, P_x, P_y and P_{xy} were $(0, 0)$, $(85.5, 0)$, $(0, 12.5)$ and $(85.5, 12.5)$ respectively. The credit card was put on the table surface, then an image was captured as shown in Fig. 6.9(b). After binary segmentation and corner detection, four corresponding points $C_i, i = 1, \ldots, 4$ were detected in the image, as indicated by yellow cross in Fig. 6.9(b). Then through the four correspondences ($C_1 \sim O$, $C_2 \sim P_x$, $C_3 \sim P_y$, $C_4 \sim P_{xy}$), the camera-plane homograpy was confirmed.

So in the initialization step, only one image was projected and two images were captured. With the addition of calculation time, the initialization can be accomplished within 20 seconds.

6.6.2 Display Quality Evaluation

Embedded code imperceptibility and user satisfaction is of the first priority in the system design. We conducted user studies based on a questionnaire. Twenty persons were invited to participate in this experiment. 500 images were collected from Google Image randomly, in which binary pattern was embedded with different intensities. The viewers were seated in front of a desk surface where the video contents were projected, and asked to comment on the quality of the image. The questions asked were simplified from the questionnaire in [61], focusing on the feeling of flickering, the recognition of image deterioration, and the overall satisfaction for projection quality. The score for each question ranged from 0 to 10. The questionnaire is enumerated as follows:

1. To what extent did you feel the flickering in the projected images (0-10)?



(a) Credit card

(b) Captured image

Figure 6.9: Images for camera-plane homography estimation.

2. To what extent did you recognize the image deterioration (such as discolorment and ghost image) (0-10)?

3. To what extent were you satisfied with the quality of projected images (0-10)?

The average scores of the subjective evaluation are illustrated in Fig. 6.10. When the embedded intensity was small, i.e., $\Delta = 5, 10$, the viewer could rarely notice the embedded codes and were satisfied with the projection quality. With the increase of the embedded intensity, the viewers' sense of flickering and image degradation became stronger. When $\Delta = 25$, almost every viewer was not satisfied with the projection quality.

In practice, because it was difficult to retrieve weakly embedded codes with the standard commercial cameras, we chose $\Delta = 10$ in our configuration, striking a compromise between user satisfaction and code imperceptibility.

6.6.3 Touch Accuracy Evaluation

Similar to [63], we specially designed an image, in which 35 circles were distributed uniformly. As shown in Fig. 6.11(a), the center of each circle, indicated by the cross symbol, was known. The testing pattern was projected to three table surfaces with different textures as shown in Fig. 6.11(b-d). In each round, the users clicked the virtual projected circles one by one as accurately as they could. If a touch contact was detected, a yellow circle was placed around the clicked circle (Fig. 6.11b & d). Five persons were invited to participate in the experiment, each of them conducted 6 rounds (on the three surfaces and under two ambient illuminations, and the two different environmental illuminations are shown in Fig. 6.12). Totally, 1050 touch trials were produced.

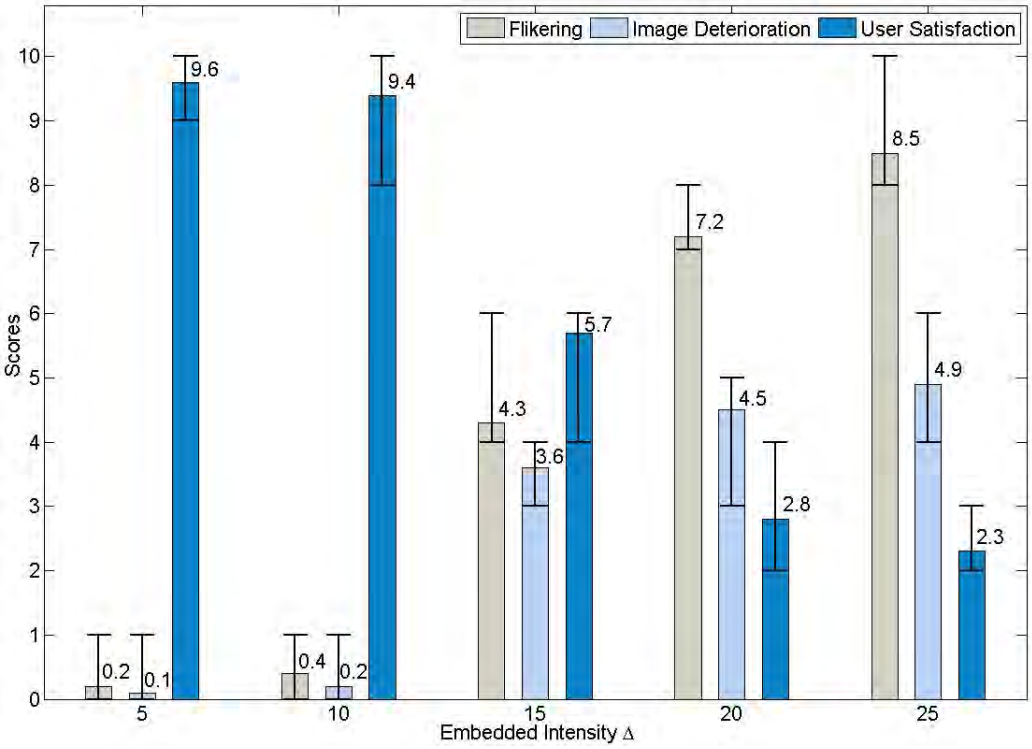


Figure 6.10: User studies results for code imperceptibility.

The precision of touch position localization is evaluated by the average distance between ground-truth and the detected position, which is formulated as

$$\epsilon = \frac{1}{N_t} \sum_{i=1}^{N_t} \sqrt{(X_{d_i} - X_{g_i})^2 + (Y_{d_i} - Y_{g_i})^2}, \tag{6.15}$$

where N_t is the total number of correctly detected touch contacts, and (X_{d_i}, Y_{d_i}) and (X_{g_i}, Y_{g_i}) are the detected position and ground-truth respectively.

The accuracy of touch detection is estimated by false reject rate (*FRR*), the probability that the system fails to detect an actual touch action, and false accept rate (*FAR*), the probability that the system incorrectly confirms a non-contact action as a touch contact. *FRR* and *FAR* are formulated as

$$FRR = \frac{N_{md}}{N}, \tag{6.16}$$

$$FAR = \frac{N_{fd}}{N}, \tag{6.17}$$

where N is the total trial number, N_{md} and N_{fd} are the number of missed detections and false detections respectively.

The detailed quantitative testing results, listed in Table 6.2, illustrate the performance and robustness of the described system against different projection surfaces and different surrounding illuminations. Here, we compared our method with some recent depth-camera sensing based methods. In [175], the informal observed spatial error of finger detection on planar surface was between 3-6 pixels, but the finger click detection error was not mentioned. As for OmniTouch [63], the *FRR* and *FAR* of finger click detection on four different surfaces were reported as 0.8% and 3.3%. Even though the evaluation data-sets, the sensing systems and working environments were not all exactly identical, the comparison results

Table 6.2: The quantitative experiment results.

Surface	Illumination			
	Dark		Normal	
	ϵ(px)	*FRR*/*FAR*(%)	ϵ(px)	*FRR*/*FAR*(%)
Gray	2.98	1.12/0.45	3.05	1.32/0.48
Yellow	3.04	1.23/0.57	3.12	1.54/0.61
Artifact	3.12	1.77/0.67	3.20	1.76/0.63

show that the described system has at least comparable performance even under less complicated devices. Some frames from one trial are shown in Fig. 6.13, camera view, third person view and fingertip trajectory are also demonstrated in each sub-figure.

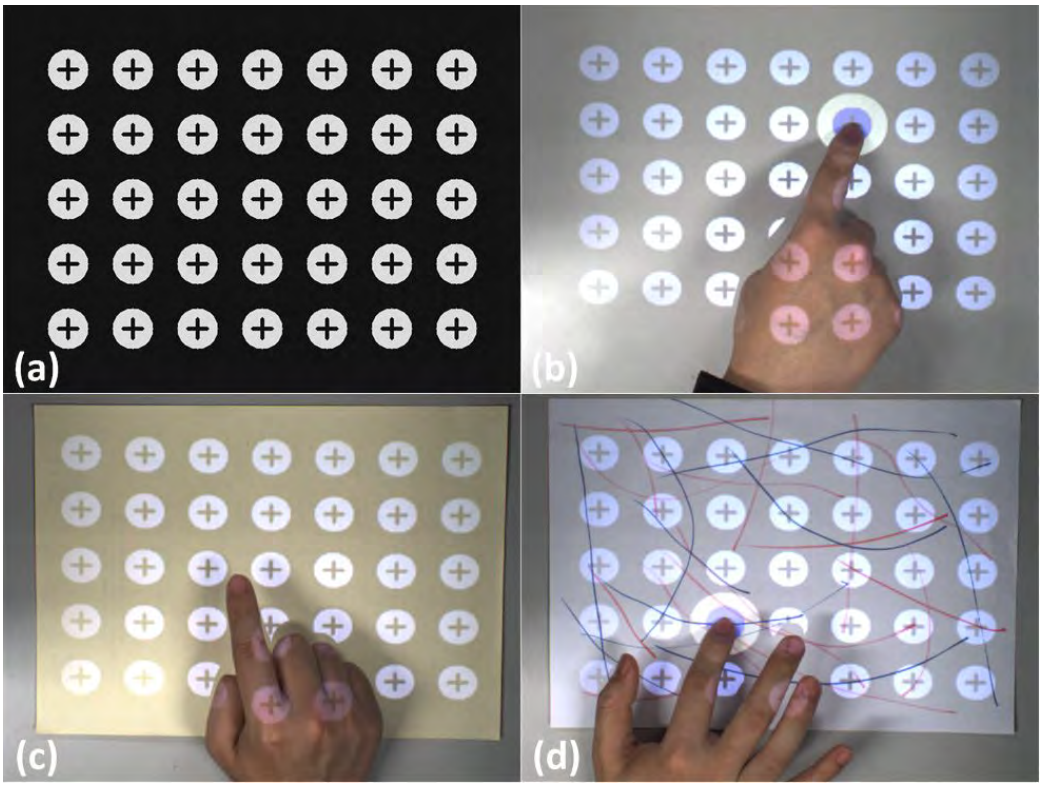
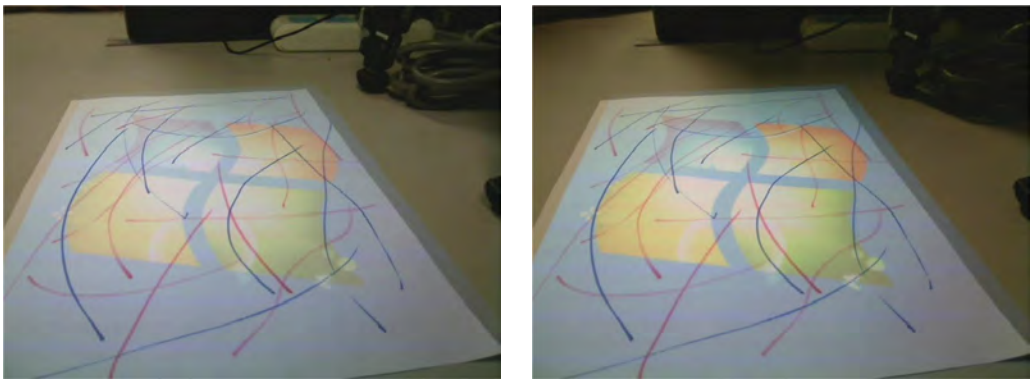


Figure 6.11: (a)Image projected for ground-truth collection, (b) gray surface, (c) yellow surface, (d) surface with artifacts



(a) Normal (b) Dark

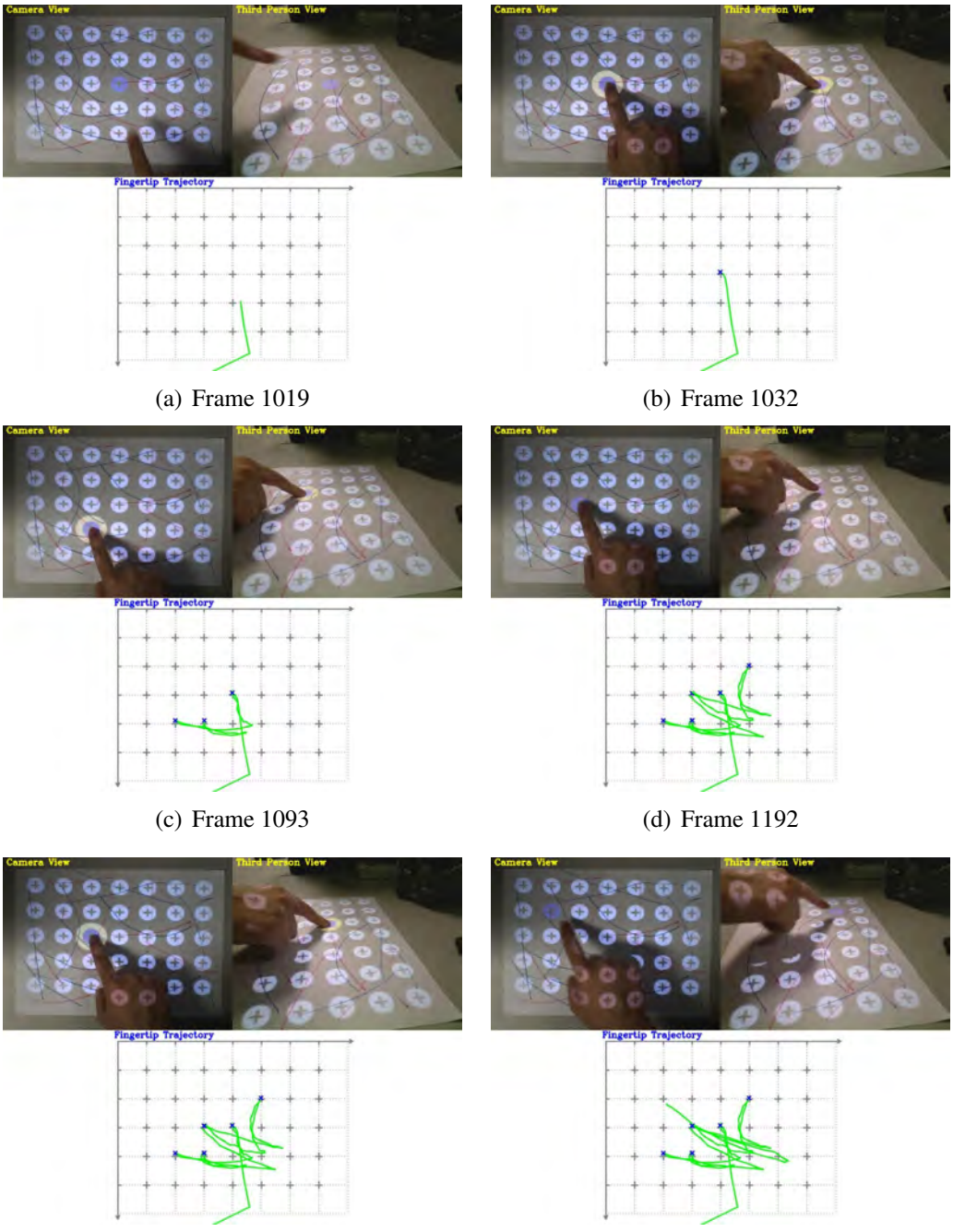
Figure 6.12: Two different environmental illuminations.

6.6.4 Trajectory Tracking Evaluation

Besides click, finger dragging is also an important action in typical touch screen operation. Here we conducted an evaluation for trajectory tracking when dragging fingers on the projection surface. As shown in Fig. 6.14(a), three different geometrical shapes (square, right triangle and circle) were projected onto the table surface. Five users were asked to drag their index finger along three boundaries one by one, the average trajectories indicated by blue curves (as shown in Fig. 6.14(b)) almost coincided with the ground-truth in grey. This experiment certifies our method can track the trajectory of dragged finger precisely.

6.6.5 Multiple-Touch Evaluation

Multi-touch refers to a touch sensing surface's ability to recognize the presence of two or more points of contact with the surface. This plural-point awareness is often used to implement advanced functionality such as pinch to zoom or activating predefined programs. In the aforementioned experiments, our method has been proved as an accurate and effective method for tracking the status of a single finger. It is straightforward to extend single-touch to multi-touch. Some key



(a) Frame 1019 (b) Frame 1032

(c) Frame 1093 (d) Frame 1192

(e) Frame 1198 (f) Frame 1224

Figure 6.13: Some frames from one trial for touch accuracy evaluation.

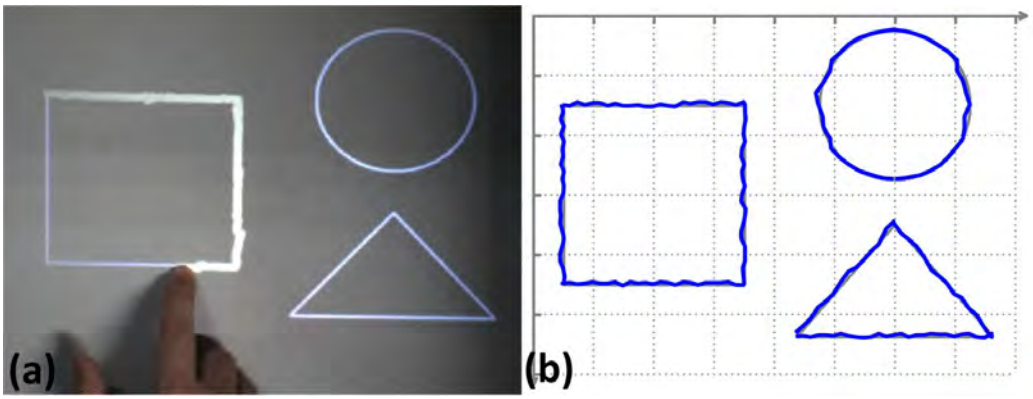


Figure 6.14: (a) Image projected for ground-truth collection, (b) fingertip dragging trajectories.

frames from one trial are demonstrated in Fig. 6.15, revealing the feasibility in multi-touch case.

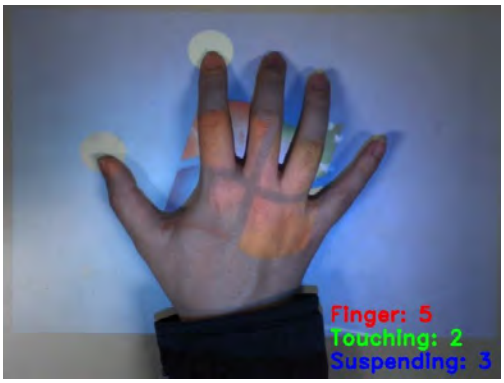
6.6.6 Efficiency Evaluation

For human-computer interface, real-time performance is of great importance. Hence we implemented the proposed system in C++ using the Intel OpenCV [5] Library to evaluate its processing time. Through multi-thread programming, the projection-capture process and calculation process were executed in two different threads respectively, each of which was able to run in real time in a desktop computer with Intel Core2 Duo 2.53GHz CPU. Table 6.3 shows the average processing times for hand segmentation in 2D image, fingertip localization, and touch detection. The total time consumption is less than $20ms$, indicating the system meets the requirement of real-time application.

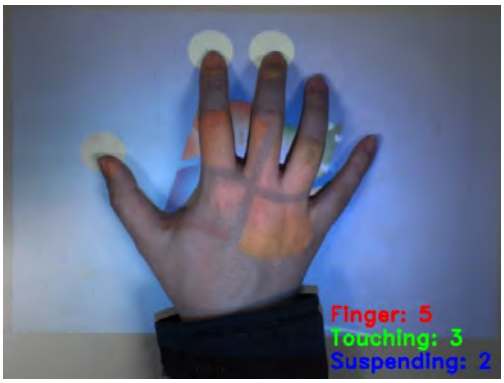
For the potential virtual keyboard application, if two hands with 10 fingers appear in the camera view, the processing time for hand segmentation and fingertip localization is almost the same as that in single hand case, since these two subroutines are global operations. To detect single finger touch action, the average

(a) No touch (Frame 10641)

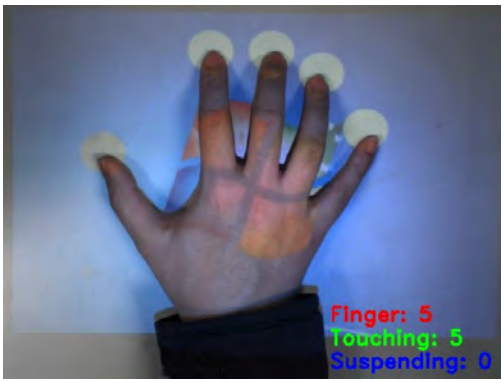
(b) One finger (Frame 10760)



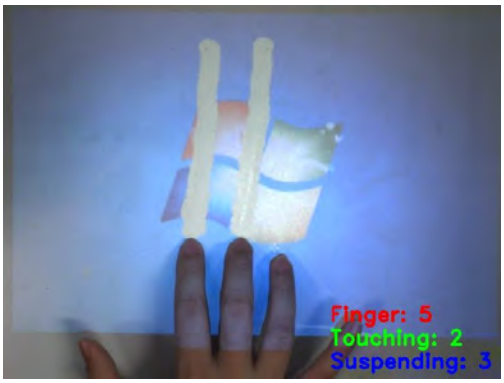
(c) Two fingers (Frame 10785)



(d) Three fingers (Frame 10809)



(e) Five fingers (Frame 10813)



(f) Two fingers dragging(Frame 11601)

Figure 6.15: Some frames from one trial for multi-touch capability evaluation.

Table 6.3: Average processing time.

Subroutine	Hand Seg.	FTip Loc.	Touch Det.	Total
Time (ms/frame)	14.63	1.32	1.74	17.69

processing time is $1.74ms$, so it will spend about $17.4ms$ on ten fingers touch detection in sequential execution. It will be faster through parallel processing. Eventually, the whole procedure can be accomplished within $30ms$, verifying our methodology is applicable in virtual keyboard application.

6.7 Summary

This chapter explores the possibility of replacing the display panel and the mouse-and-keyboard by a mere projector and camera. Specifically, it is to enable a light-colored table surface, to which the projection is illuminated, to serve as a touch-sensitive display panel for finger-based user input.

The described work lays down the setup and design of the pro-cam system for touch-sensitive interface. Single-touch, touch dragging tracking and multi-touch facilities are also constructed and thoroughly experimented with. All these form the basis of a more complete touch interface system. Future work includes more thorough experimentation with multi-hand interface using the system. Based upon the touch detection facility, advanced touch gestures (e.g. double clicking, scrolling, zoom-in, zoom-out) and even typing recognition on the described platform will also be studied.

Chapter 7

Conclusion and Future Work

This chapter presents the conclusion and some perspectives opened by this work. The contributions of the thesis are described firstly. The publications related to this work is then listed. Finally, future possible works beyond this research are discussed.

7.1 Conclusion and Contributions

This thesis has focused on developing a projector-camera system that can provide a platform for the user interacting with computer in a natural way. Three key issues, including 3D information interpretation, display and sensing, and human action recognition have been involved in this work. All the contributions in this work can be enumerated as follows.

The first contribution in this work is to determining the 6-DOF head pose by the use of an imperceptible structured light system to demonstrate the feasibility of combining 2D texture information with 3D depth information. The method is

able to track accurate 3D positions of salient facial landmarks without the need of going through any training process. Firstly, through elaborate pattern projection strategy and camera-projector synchronization, a pattern-illuminated image and the corresponding scene-texture image are captured under illumination that appears as white light yet embeds coded patterns. Then, in the point cloud generated by structured light sensing, the facial feature points in the scene-texture image localized by AAM will have their 3D positions interpolated. Correspondences between such facial features in 3D, with those associated with the previous or reference image frame, can then be constructed. Finally, the head orientation and translation are estimated by SVD of a correlation matrix that is generated from such point pairs in 3D. Experiment results show that mean absolute estimation errors of our method are 2.02°, 1.18° and 0.76°, in yaw, pitch and roll directions, respectively.

The second contribution is that we have proposed a novel method of embedding imperceptible structured codes into arbitrarily intended projection. Through precise projector-camera synchronization, structured codes consisting of three primitive shapes are embedded into the projection, in a way that is imperceptible to viewers but extractable from the ”difference image” between successive images captured by a camera. To make the decoding process more robust against noise, we do not extract the codes by region segmentation in the image domain. Instead we employ specially trained classifiers to detect and identify the codes. To enhance the error tolerance further, specially designed primitive shapes and large Hamming distance are adopted in the spatial coding. Even with some bits of the codewords missed or wrongly coded, the correct correspondence could still be derived correctly. Through the proposed method, more than 90% of the embedded feature points could their correspondences found correctly. Sensitivity analysis proved that our method was still effective even through the scenarios of

training stage and operation stage were different. Some application cases were also demonstrated.

The third contribution is that we have introduced a coarse-to-fine approach to solve the segmentation problem in projector-camera system. The main idea of the method is to combine contrast saliency map with mean-shift based smoothing and segmentation by a confidence function. Low-level contrast saliency detection enables the hand region to be highlighted roughly, and mean-shift based smoothing method removes the noises induced by projection contents without demolishing discontinuity information. Moreover, without any pre-training and pre-calibration procedures, the robust, precise and also rapid hand segmentation can be derived. Extensive experiments showed that our method archived the highest precision and recall rate among previous methods.

The last contribution is that we explored the possibility of replacing the display panel and the mouse-and-keyboard by a mere projector and camera. Specifically, it is to enable a light-colored table surface, to which the projection is illuminated, to serve as a touch-sensitive display panel for finger-based user input. Compared with the recent depth-camera sensing based methods, the experimental results showed that the described system has comparable performance even under less complicated devices.

7.2 Related Publications

[1] J. Dai and R. Chung, Making Any Planar Surface into a Touch-sensitive Display by a Mere Projector and Camera, *In Proc. of 25th IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12) - Workshop (PROCAMS'12)*, pages 35-42, 2012.

[2] J. Dai and R. Chung, On Making Projector both a Display Device and a

3D Sensor, *In Proc. of The 8th International Symposium on Visual Computing (ISVC'12)*, pages 654-664, 2012.

[3] J. Dai and R. Chung, Combining Contrast Saliency and Region Discontinuity for Precise Hand Segmentation in Projector-Camera System, *To Appear in Proc. of The 21st International Conference on Pattern Recognition (ICPR'12), November 2012.*

[4] J. Dai and R. Chung, Embedding Imperceptible Codes into Video Projection and Applications in Robotics, *To Appear in Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'12), October 2012.*

[5] J. Dai and R. Chung, Head Pose Estimation by Imperceptible Structured Light Sensing, *In Proc. of IEEE International Conference on Robotics and Automation (ICRA'11)*, pages 1646-1651, 2011.

[6] J. Dai and R. Chung, Sensitivity Evaluation of Embedded Code Detection in Imperceptible Structured Light Sensing, *Submitted to IEEE Workshop on Robot Vision (WoRV'13), January 2013*.

[7] J. Dai and R. Chung, Embedding Invisible Codes into Regular Video Projection: Principle, Evaluation and Applications, *Submitted to IEEE Transactions on Circuits and Systems for Video Technology*.

[8] J. Dai and R. Chung, Touch-sensitive Display on Arbitrary Planar Surface by a mere Projector and Camera, *Prepared to submit to IEEE Transactions on Pattern Analysis and Machine Intelligence*.

7.3 Future Work

The extension of this work lies on four directions: (1) Relaxation the two assumptions in head pose estimation method; and (2) increasing the signal-to-noise ratio of substraction image and the density of the embedded patterns in ISL sensing;

and (3) more elaborate fusing confidence function to refine hand segmentation accuracy; and (4) the extension to multi-hand supporting and advanced touch gestures recognition in the touch-sensitive interface.

There are two assumptions made in head pose estimation approach. One is that the position of the human head is constant between the captures of the pattern-illuminated image and the subsequent scene-texture image. The other is that the human head is modeled as a rigid object. The former one will be violated when user has quick motion, while the latter will be destroyed when there are extreme expression variation. The future work will lie on introduction of motion compensation between the pattern-illuminated image and the subsequent scene-texture image, and of the use of 3D deformable model that embraces facial expression variation, so that the two assumptions could be relaxed.

In the current ISL system, the image capture interval is $10ms$. In sensing object that moves fast, the substantial displacement between successive images will result in blur or destruction of the embedded codes in the difference image. Some compensation methods need be in place to deal with the problem. And some image enhancement technologies should be studied to increase the low signal-to-noise ratio of subtraction image. In addition, the embedded code could be denser for more precise 3D sensing.

For the touch-sensitive interface, single-touch, touch dragging tracking and multi-touch facilities have been already constructed and thoroughly experimented with. All these form the basis of a more complete touch interface system. Future work includes more thorough experimentation with multi-hand interface using the system. Based upon the touch detection facility, advanced touch gestures (e.g. double clicking, scrolling, zoom-in, zoom-out) and even typing recognition on the described platform will also be studied.

Bibliography

[1] ARRICK robotics. http://www.arrickrobotics.com/. Accessed: 30/08/2012.

[2] Google image. http://images.google.com/. Accessed: 30/08/2012.

[3] Light Touch. http://lightblueoptics.com. Accessed: 30/08/2012.

[4] Microsoft Kinect. http://www.xbox.com/kinect. Accessed: 30/08/2012.

[5] Open Source Computer Vision. http://opencv.org/. Accessed: 30/08/2012.

[6] PrimeSence. http://www.primesense.com. Accessed: 30/08/2012.

[7] A. Bobick and J. Davis. The representation and recognition of action using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(3):257–267, 2001.

[8] A. Gee and R. Cipolla. Determining the gaze of faces in images. *Image and Vision Computing*, 12(10):639–647, 1994.

[9] A. Grundhofer, M. Seeger, F. Hantsch and O. Bimber. Dynamic adaptation of projected imperceptible codes. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–10, 2007.

[10] A. Ramamoorthy, N. Vaswani, S. Chaudhury, and S. Banerjee. Recognition of dynamic hand gestures. *Pattern Recognition*, 36:2069–2081, 2003.

[11] A. Shamaie and A. Sutherland. Hand tracking in bimanual movements. *Image and Vision Computing*, 23(13):1131–1149, 2005.

[12] R. Achanta, S. S. Hemami, F. J. Estrada, and S. Süsstrunk. Frequency-tuned salient region detection. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '09, pages 1597–1604, 2009.

[13] I. Albitar, P. Graebling, and C. Doignon. Robust structured light coding for 3d reconstruction. In *Proceedings of IEEE 11th International Conference on Computer Vision*, pages 1–6, 2007.

[14] A. A. Argyros and M. I. A. Lourakis. Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *Proceedings of 8th European Conference on Computer Vision*, pages 368–379, 2004.

[15] M. Ashdown and P. Robinson. Escritoire: A personal projected display. *IEEE MultiMedia Magazine*, 12(1):34–42, 2005.

[16] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1372–1384, 2006.

[17] B. Stenger, R. Mendonca, and R. Cippola. Model-based 3d tracking of an articulated hand. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 126–133, 2002.

[18] S. Ba and J. Odobez. A probabilistic framework for joint head tracking and pose estimation. In *Proceedings of the 17th International Conference on Pattern Recognition*, pages 264–267 Vol.4, 2004.

[19] H. Benko, R. Jota, and A. Wilson. Miragetable: freehand interaction on a projected augmented reality tabletop. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pages 199–208, 2012.

[20] H. Benko, A. D. Wilson, and R. Balakrishnan. Sphere: multi-touch interactions on a spherical display. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST '08, pages 77–86, 2008.

[21] F. Berard. The magic table: Computer vision based augmentation of a whiteboard for creative meetings. In *Proceedings of IEEE International Workshop on Projector-Camera Systems*, 2003.

[22] D. Beymer. Face recognition under varying pose. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 756–761, 1994.

[23] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[24] K. W. Bowyer, K. Chang, and P. Flynn. A survey of approaches and challenges in 3d and multi-modal 3d + 2d face recognition. *Computer Vision and Image Understanding*, 101(1):1–15, 2006.

[25] K. L. Boyer and A. C. Kak. Color-encoded structured light for rapid active ranging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):14–28, 1987.

[26] M. Breitenstein, D. Kuettel, T. Weise, L. van Gool, and H. Pfister. Real-time face pose estimation from single range images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[27] D. Chai and K. Ngan. Locating facial region of a head-and-shoulders color image. In *Proceedings. Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 124–129, 1998.

[28] S. Chen, Y. Li, and J. Zhang. Vision processing for realtime 3-d data acquisition based on coded structured light. *IEEE Transactions on Image Processing*, 17(2):167–176, 2008.

[29] X. Chen, X. Yang, S. Xiao, and M. Li. Color mixing property of a projector-camera system. In *Proceedings of the 5th ACM/IEEE International Workshop on Projector camera systems*, PROCAMS '08, pages 1–6, 2008.

[30] X. Chen, X. Yang, S. Xiao, and M. Li. Color mixing property of a projector-camera system. In *Proceedings of the 5th ACM/IEEE International Workshop on Projector Camera Systems*, PROCAMS '08, pages 1–6, 2008.

[31] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu. Global contrast based salient region detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 409–416, 2011.

[32] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[33] J. Corso, D. Burschka, and G. Hager. The 4d touchpad: Unencumbered hci with vics. In *Proceedings of the CVPR-HCI Workshop*, 2003.

[34] D. Cotting, M. Naef, M. Gross, and H. Fuchs. Embedding imperceptible patterns into projected images for simultaneous acquisition and display. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '04, pages 100–109, 2004.

[35] D. Cotting, R. Ziegler, M. Gross, and H. Fuchs. Adaptive instant displays: Continuously calibrated projections using per-pixel light control. *Computer Graphics Forum*, 24(3):705–714, 2005.

[36] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 142–149, 2000.

[37] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[38] D. Leigh and P. Dietz. Diamondtouch characteristics and capabilities. In *Proceeding of UbiComp 2002 Workshop on Collaboration with Interactive Tables and Walls*, 2002.

[39] D. Desjardins and P. Payeur. Dense stereo range sensing with marching pseudo-random patterns. In *Proceedings of the Fourth Canadian Conference on Computer and Robot Vision*, CRV '07, pages 216–226, 2007.

[40] M. Donoser and H. Bischof. Real time appearance based hand tracking. In *Proceedings of 19th International Conference on Pattern Recognition*, pages 1–4, 2008.

[41] E. Ong and R. Bowden. A boosted classifier tree for hand shape detection. In *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pages 889–894, 2004.

[42] E. Osuna, R. Freund, and F. Girosit. Training support vector machines: an application to face detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 130 –136, 1997.

[43] D. W. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3-d rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9(5-6):272–290, 1997.

[44] T. Etzion. Constructions for perfect maps and pseudorandom arrays. *IEEE Transactions on Information Theory*, 34(5):1308–1316, 1988.

[45] P. Fechteler and P. Eisert. Adaptive colour classification for structured light systems. *IET Computer Vision*, 3(2):49–59, 2009.

[46] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal Computer Vision*, 59(2):167–181, 2004.

[47] Fitriani and W.-B. Goh. Interacting with projected media on deformable surfaces. In *Proceedings of IEEE 11th International Conference on Computer Vision*, pages 1–6, 2007.

[48] D. Fofi, T. Sliwa, and Y. Voisin. A comparative survey on invisible structured light. In *Proceedings of Machine Vision Applications in Industrial Inspection XII*, pages 90–98, 2004.

[49] G. Bradski and J. Davis. Motion segmentation and pose recognition with motion history gradients. *Machine Vision and Applications*, 13(3):174–184, 2002.

[50] G. Hager and P. Belhumeur. Real-time tracking of image regions with changes in geometry and illumination. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 403–410, 1996.

[51] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *Proceedings of European Conference on Computer Vision*, pages 666–680, 2002.

[52] D. Gavrila and L. Davis. 3-d model-based tracking of humans in action: a multi-view approach. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 73–80, 1996.

[53] A. H. Gee and R. Cipolla. Fast visual tracking by temporal consensus. *Image and Vision Computing*, 14(2):105–114, 1996.

[54] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '10, pages 2376–2383, 2010.

[55] A. Griesser and L. Van Gool. Automatic interactive calibration of multi-projector-camera systems. In *Proceedings of IEEE International Workshop on Projector-Camera Systems*, 2006.

[56] A. Grundhöfer, M. Seeger, F. Hantsch, and O. Bimber. Dynamic adaptation of projected imperceptible codes. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '07, pages 1–10, 2007.

[57] H. Chen and T. Liu. Trust-region methods for real-time tracking. In *Proceedings of International Conference on Computer Vision*, pages 717–722, 2001.

[58] H. S. Yoon, J. Soh, Y. J. Bae, and H. S. Yang. Hand gesture recognition using combined features of location, angle and velocity. *Pattern Recognition*, 34:1491–1501, 2001.

[59] H. Spoelder, F. Vos and et al. Some aspects of pseudo random binary array-based surface characterization. *IEEE Transactions on Instrument and Measurement*, 49(6):1331–1336, 2000.

[60] H.A. Rowley, S. Baluja and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[61] Hanhoon Park, Byung-Kuk Seo and Jong-Il Park. Subjective evaluation on visual perceptibility of embedding complementary patterns for nonintrusive projection-based augmented reality. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(5):687–696, 2010.

[62] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, pages 545–552, 2006.

[63] C. Harrison, H. Benko, and A. D. Wilson. Omnitouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User Interface Software and Technology*, UIST '11, pages 441–450, 2011.

[64] C. Harrison, D. Tan, and D. Morris. Skinput: appropriating the body as an input surface. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems*, CHI '10, pages 453–462, 2010.

[65] T. T. Hewett, R. Baecker, S. Card, T. Carey, J. Gasen, M. Mantei, G. Perlman, G. Strong, and W. Verplank. *ACM SIGCHI Curricula for Human-Computer Interaction*. Association for Computing Machinery, 2009.

[66] H. Hiroshi Kawasaki, R. Furukawa, R. Sagawa, and Y. Yagi. Dynamic scene shape reconstruction using a single structured light pattern. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[67] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '07, 2007.

[68] Y. Hu, L. Chen, Y. Zhou, and H. Zhang. Estimating face pose by facial asymmetry and geometry. In *Proceedings of Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 651–656, 2004.

[69] K. Huang and M. Trivedi. Robust real-time detection, tracking, and pose estimation of faces in video streams. In *Proceedings of the 17th International Conference on Pattern Recognition*, pages 965–968 Vol.3, 2004.

[70] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[71] J. Bruske, E. Abraham-Mumm, J. Pauli, and G. Sommer. Head-pose estimation from facial images with subspace neural networks. In *Proceedings of International Conference Neural Networks and Brain*, pages 528–531, 1998.

[72] J. Cai and A. Goshtasby. Detecting human faces in color images. *Image and Vision Computing*, 18(1):63–75, 1999.

[73] J. Eisenstein, S. Ghandeharizadeh, L. Golubchik, C. Shahabi, Donghui Y., and R. Zimmermann. Device independence and extensibility in gesture recognition. In *Proceedings of IEEE Virtual Reality*, pages 207–214, 2003.

[74] J. G. Wang and E. Sung. Em enhancement of 3D head pose estimated by point at infinity. *Image and Vision Computing*, 25(12):1864–1874, 2007.

[75] J. Huang, X. Shao and H. Wechsler. Face pose discrimination using support vector machines (svm). In *Proceedings of International Conference on Pattern Recognition*, volume 1, pages 154–156, 1998.

[76] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proceedings of International Conference on Computer Vision*, pages 572–578, 1999.

[77] J. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proceedings of European Conference on Computer Vision*, pages 3–19, 2000.

[78] J. Martin, V. Devin, and J. Crowley. Active hand tracking. In *Proceedings of IEEE Conference on Automatic Face and Gesture Recognition*, pages 573–578, 1998.

[79] J. Rehg and T. Kanade. Digiteyes: Vision-based hand tracking for human-computer interaction. In *Proceedings of Workshop on Motion of Non-Rigid and Articulated Bodies*, pages 16–24, 1994.

[80] J. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *Proceedings of International Conference on Computer Vision*, pages 612–617, 1995.

[81] J. Salvi, J. Pagés and J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37(4):827–849, 2004.

[82] J. Salvi, S. Fernandez, T. Pribanic and X. Llado. A state of the art in structured light patterns for surface profilometry. *Pattern Recognition*, 43(8):2666–2680, 2010.

[83] J. Sullivan and S. Carlsson. Recognizing and tracking human action. In *Proceedings of European Conference on Computer Vision*, pages 629–644, 2002.

[84] J. Triesch and C. Malsburg. Robust classification of hand postures against complex background. In *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pages 170–175, 1996.

[85] J. Triesch and C. Von der Malsburg. A gesture interface for human-robot-interaction. In *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pages 546–551, 1998.

[86] J. Weaver, T. Starner, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):1371–1378, 1998.

[87] J. Wu and M. Trivedi. A two-stage head pose estimation framework and evaluation. *Pattern Recognition*, 41(3):1138–1158, 2008.

[88] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in timesequential images using hidden markov model. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 379–385, 1992.

[89] T. Jebara and A. Pentland. Parametrized structure from motion for 3d adaptive feedback tracking of faces. In *Proceedings of 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 144–150, 1997.

[90] P. Jimenez, J. Nuevo, L. Bergasa, and M. Sotelo. Face tracking and pose estimation with automatic three-dimensional model construction. *IET Computer Vision*, 3(2):93–102, 2009.

[91] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.

[92] M. Kimura, M. Mochimaru, and T. Kanade. Projector calibration using arbitrary planes and calibrated camera. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–2, 2007.

[93] S. Kiyasu, H. Hoshino, K. Yano, and S. Fujimura. Measurement of the 3-d shape of specular polyhedrons using an m-array coded light source. *IEEE Transactions on Instrumentation and Measurement*, 44(3):775–778, 1995.

[94] R. Kjeldsen, C. Pinhanez, G. Pingali, J. Hartman, T. Levas, and M. Podlaseck. Interacting with steerable projected displays. In *Proceedings of Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 402–407, 2002.

[95] T. Koninckx and L. Van Gool. Real-time range acquisition by adaptive structured light. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):432–445, 2006.

[96] V. Krüger and G. Sommer. Gabor wavelet networks for efficient head pose estimation. *Image and Vision Computing*, 20(9-10):665–672, 2002.

[97] L. Goncalves, E. di Bernardo, E. Ursella, and P. Perona. Monocular tracking of the human arm in 3d. In *Proceedings of International Conference on Computer Vision*, pages 764–770, 1995.

[98] J. Letessier and F. Bérard. Visual tracking of bare fingers for interactive surfaces. In *Proceedings of the 17th annual ACM Symposium on User Interface Software and Technology*, UIST '04, pages 119–122, 2004.

[99] A. Licsár and T. Szirányi. Hand gesture recognition in camera-projector system. In *ECCV Workshop on HCI*, pages 83–93, 2004.

[100] M. A. Drouin, G. Godin, and S. Roy. An energy formulation for establishing the correspondence used in projector calibration. In *In Proceedings of the Fourth International Symposium on 3D Data Processing, Visualization and Transmission*, 2008.

[101] M. Ashdown and Y. Sato. Steerable projector calibration. In *Proceedings of IEEE International Workshop on Projector-Camera Systems*, 2005.

[102] M. Cote, P. Payeur, and G. Comeau. Comparative study of adaptive segmentation techniques for gesture analysis in unconstrained environments. In *Proceedings of IEEE International Workshop on Imagining Systems and Techniques*, pages 28–33, 2006.

[103] M. H. Yang, N. Ahuja, and M. Tabb. Extraction of 2d motion trajectories and its application to hand gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1061–1074, 2002.

[104] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[105] Y.-F. Ma and H.-J. Zhang. Contrast-based image attention analysis by using fuzzy growing. In *Proceedings of the eleventh ACM international conference on Multimedia*, MULTIMEDIA '03, pages 374–381, 2003.

[106] F. MacWilliams and N. Sloane. Pseudo-random sequences and arrays. *Proceedings of the IEEE*, 64(12):1715–1729, 1976.

[107] M. Malciu and F. Preteux. A robust model-based approach for 3d head tracking in video sequences. In *Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 169–174, 2000.

[108] S. Malik and J. Laszlo. Visual touchpad: a two-handed gestural input device. In *Proceedings of the 6th international conference on Multimodal interfaces*, ICMI '04, pages 289–296, 2004.

[109] J. Marshall, T. Pridmore, M. Pound, S. Benford, and B. Koleva. Pressing the flesh: Sensing multiple touch and finger pressure on arbitrary surfaces. In *Proceedings of the 6th International Conference on Pervasive Computing*, Pervasive '08, pages 38–55, 2008.

[110] M. Maruyama and S. Abe. Range sensing by projecting multiple slits with random cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):647–651, 1993.

[111] Y. Matsumoto, N. Sasao, T. Suenaga, and T. Ogasawara. 3d model-based 6-dof head tracking by a single camera for human-robot interaction. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3194–3199, 2009.

[112] N. Matsushita and J. Rekimoto. Holowall: designing a finger, hand, body, and object sensitive wall. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, UIST '97, pages 209–210, 1997.

[113] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

[114] T. Maurer and C. von der Malsburg. Tracking and learning graphs and pose on image sequences of faces. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 176–181, 1996.

[115] R. Morano, C. Ozturk, R. Conn, S. Dubin, S. Zietz, and J. Nissano. Structured light using pseudorandom codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):322–327, 1998.

[116] L.-P. Morency, A. Rahimi, and T. Darrell. Adaptive view-based appearance models. In *Proceedings of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 803–810 vol.1, 2003.

[117] H. Morita, K. Yajima, and S. Sakata. Reconstruction of surfaces of 3-d objects by m-array pattern projection method. In *Proceedings of Second International Conference on Computer Vision*, pages 468–473, 1988.

[118] E. Murphy-Chutorian and M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):607–626, 2009.

[119] E. Murphy-Chutorian and M. Trivedi. Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):300–311, 2010.

[120] E. D. Mynatt, T. Igarashi, W. K. Edwards, and A. LaMarca. Flatland: new dimensions in office whiteboards. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '99, pages 346–353, 1999.

[121] R. Newman, Y. Matsumoto, S. Rougeaux, and A. Zelinsky. Real-time stereo tracking for head pose and gaze estimation. In *Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 122–128, 2000.

[122] S. Niyogi and W. Freeman. Example-based head tracking. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 374–378, 1996.

[123] O. Bimber, D. Iwai, G. Wetzstein and A. Grundhöfer. The visual computing of projector-camera systems. In *ACM SIGGRAPH 2008 classes*, SIGGRAPH '08, pages 1–25, 2008.

[124] S. Ohayon and E. Rivlin. Robust 3d head tracking using camera pose estimation. In *Proceedings of 18th International Conference on Pattern Recognition*, pages 1063–1066, 2006.

[125] J. Pages, C. Collewet, F. Chaumette, and J. Salvi. An approach to visual servoing based on coded light. In *Proceedings of 2006 IEEE International Conference on Robotics and Automation*, pages 4118–4123, 2006.

[126] J. Pages, J. Salvi, and J. Forest. A new optimised de bruijn coding strategy for structured light patterns. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 4, pages 284–287, 2004.

[127] H. Park, M.-H. Lee, B.-K. Seo, Y. Jin, and J.-I. Park. Content adaptive embedding of complementary patterns for nonintrusive direct-projected augmented reality. In *Proceedings of the 2nd International Conference on Virtual Reality*, ICVR'07, pages 132–141, 2007.

[128] J. Patten, H. Ishii, J. Hines, and G. Pangaro. Sensetable: a wireless object tracking platform for tangible user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '01, pages 253–260, 2001.

[129] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and M. Brady. A linguistic feature vector for the visual interpretation of sign language. In *Proceedings of European Conference on Computer Vision*, pages 391–401, 2004.

[130] R. Cutler and M. Turk. View-based interpretation of real-time optical flow for gesture recognition. In *Proceedings of International Conference on Face and Gesture Recognition*, pages 416–421, 1998.

[131] R. Hartley, and A. Zisserman. *Multiple View Geometry in Computer Vision(2e)*. Cambridge University Press, 2004.

[132] R. Lienhart, A. Kuranov and V.Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *In DAGM 25th Pattern Recognition Symposium*, pages 297–304, 2003.

[133] R. Rosales, V. Athitsos, L. Sigal, and S. Sclaroff. 3d hand pose reconstruction using specialized mappings. In *Proceedings of International Conference on Computer Vision*, pages 378–385, 2001.

[134] R. Rae and H. Ritter. Recognition of human head orientation based on artificial neural networks. *IEEE Transactions on Neural Networks*, 9(2):257–265, 1998.

[135] R. Raskar and P. Beardsley. A self-correcting projector. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 504–508 vol.2, 2001.

[136] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The office of the future: a unified approach to image-based modeling and spatially immersive displays. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 179–188, 1998.

[137] J. Rekimoto. Smartskin: an infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '02, pages 113–120, 2002.

[138] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004.

[139] S. Audet and J. R. Cooperstock. Shadow removal in front projection environments using object tracking. In *Proceedings of IEEE International Workshop on Projector-Camera Systems*, 2007.

[140] S. Audet and M. Okutomi. A user-friendly method to geometrically calibrate projector-camera systems. In *Proceedings of IEEE International Workshop on Projector-Camera Systems*, 2009.

[141] S. Baker, I. Matthews, J. Xiao, R. Gross, T. Kanade and T. Ishikawa. Real-time non-rigid driver head tracking for driver mental state estimation. In *Proceedings of 11th World Congress Intlligent Transportation Systems*, 2004.

[142] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[143] S. McKenna, Y. Raja, and S. Gong. Tracking color objects using adaptive mixture models. *Image and Vision Computing*, 17(3):225–231, 1999.

[144] S. Zollmann and O. Bimber. Imperceptible calibration for radiometric compensation. In *Proceedings of Eurographics*, 2007.

[145] J. Salvi, J. Batlle, and E. Mouaddib. A robust-coded pattern projection for dynamic 3d scene measurement. *Pattern Recognition Letters*, 19(11):1055–1065, 1998.

[146] J. Salvi, J. Batlle, and E. Mouaddib. A robust-coded pattern projection for dynamic 3D scene measurement. *Pattern Recognition Letters*, 19(11):1055–1065, 1998.

[147] J. Salvi, J. Pagés, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37(4):827–849, 2004.

[148] Y. Sato, Y. Kobayashi, and H. Koike. Fast tracking of hands and fingertips in infrared images for augmented desk interface. In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, FG '00, 2000.

[149] D. Saxe and R. Foulds. Toward robust skin identification in video images. In *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, pages 379–384, 1996.

[150] L. Sigal, S. Sclaroff, and V. Athitsos. Skin color-based video segmentation under time-varying illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):862–877, 2004.

[151] R. Sodhi, H. Benko, and A. Wilson. Lightguide: projected visualizations for hand movement guidance. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pages 179–188, 2012.

[152] P. Song, S. Winkler, S. O. Gilani, and Z. Zhou. Vision-based projected tabletop interface for finger interactions. In *Proceedings of the 2007 IEEE international conference on Human-computer interaction*, HCI'07, pages 49–58, 2007.

[153] Z. Song and C.-K. Chung. Determining both surface position and orientation in structured-light-based sensing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1770–1780, 2010.

[154] Z. Song and R. Chung. Use of lcd panel for calibrating structured-light-based range sensing system. *IEEE Transactions on Instrumentation and Measurement*, 57(11):2623–2630, 2008.

[155] S. Suzuki and K. Abe. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.

[156] T. Cootes, C. Taylor, D. Cooper and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[157] T. Cootes, G. Edwards and C. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[158] T. Cootes, K. Walker and C. Taylor. View-based active appearance models. In *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pages 227–232, 2000.

[159] T. Darrell, I. Essa, and A. Pentland. Task-specific gesture analysis in real-time using interpolated views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1236–1242, 1996.

[160] T. Horprasert, Y. Yacoob and L. Davis. Computing 3-D head orientation from a monocular image sequence. In *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pages 242–247, 1996.

[161] B. Ullmer and H. Ishii. The metadesk: models and prototypes for tangible user interfaces. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, UIST '97, pages 223–232, 1997.

[162] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

[163] J. Underkoffler and H. Ishii. Illuminating light: an optical design tool with a luminous-tangible interface. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '98, pages 542–549, 1998.

[164] A. Utsumi and J. Ohya. Image segmentation for human tracking using sequential-image-based hierarchical adaptation. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 911–916, 1998.

[165] V. Athitsos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 432–439, 2003.

[166] M. Van den Bergh and L. Van Gool. Combining RGB and tof cameras for real-time 3D hand gesture interaction. In *Proceedings of the 2011 IEEE Workshop on Applications of Computer Vision*, WACV '11, pages 66–72, 2011.

[167] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518, 2001.

[168] P. A. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[169] C. von Hardenberg and F. Bérard. Bare-hand human-computer interaction. In *Proceedings of the 2001 Workshop on Perceptive User Interfaces*, PUI '01, pages 1–8, 2001.

[170] Q. Wang, X. Chen, and W. Gao. Skin color weighted disparity competition for hand segmentation from stereo camera. In *Proceedings of the British Machine Vision Conference*, pages 1–11, 2010.

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

[172] K. D. Willis. A pre-history of handheld projector-based interaction. *Personal Ubiquitous Computing*, 16(1):5–15, 2012.

[173] A. D. Wilson. Touchlight: an imaging touch screen and display for gesture-based interaction. In *Proceedings of the 6th international conference on Multimodal interfaces*, ICMI '04, pages 69–76, 2004.

[174] A. D. Wilson. Playanywhere: a compact interactive tabletop projection-vision system. In *Proceedings of the 18th annual ACM Symposium on User Interface Software and Technology*, UIST '05, pages 83–92, 2005.

[175] A. D. Wilson. Using a depth camera as a touch sensor. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, pages 69–72, 2010.

[176] A. D. Wilson and H. Benko. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of the 23nd annual ACM symposium on User Interface Software and Technology*, UIST '10, pages 273–282, 2010.

[177] J. Wu and M. M. Trivedi. A two-stage head pose estimation framework and evaluation. *Pattern Recognition*, 41(3):1138–1158, 2008.

[178] Y. Wu and K. Toyama. Wide-range, person- and illumination-insensitive head orientation estimation. In *Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 183–188, 2000.

[179] X.Zabulis, H. Baltzakis, A. A. Argyros. *Vision-based Hand Gesture Recognition for Human Computer Interaction*, chapter 34, pages 34.1–34.30. The Universal Access Handbook. Lawrence Erlbaum Associates, Inc., 2009.

[180] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[181] Y. Wu and T. S. Huang. View-independent recognition of hand postures. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, pages 84–94, 2000.

[182] Y. Wu, J. Lin, and T. Huang. Capturing natural hand articulation. In *Proceedings of International Conference on Computer Vision*, pages 426–432, 2001.

[183] P. Yao, G. Evans, and A. Calway. Using affine correspondence to estimate 3-d facial pose. In *Proceedings of 2001 International Conference on Image Processing*, pages 919–922 vol.3, 2001.

[184] Y. Zhai and M. Shah. Visual attention detection in video sequences using spatiotemporal cues. In *Proceedings of the 14th annual ACM international conference on Multimedia*, MULTIMEDIA '06, pages 815–824, 2006.

[185] Z. Zhang, Y. Hu, M. Liu, and T. Huang. Head pose estimation in seminar room using multi view face detectors. In *Proceedings of the 1st international evaluation conference on Classification of events, activities and relationships*, pages 299–304, 2007.

[186] G. Zhao, L. Chen, J. Song, and G. Chen. Large head movement tracking using sift-based registration. In *Proceedings of the 15th international conference on Multimedia*, MULTIMEDIA '07, pages 807–810, 2007.

[187] L. Zhao, G. Pingali, and I. Carlbom. Real-time head orientation estimation using neural networks. In *Proceedings of International Conference on Image Processing*, pages 297–300 vol.1, 2002.

[188] J. Zhou and J. Hoang. Real time robust human detection and tracking system. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 2005.