Light Field Vision for Transparent Object Recognition

徐, 軼超

https://doi.org/10.15017/1543993

出版情報：九州大学, 2015, 博士（学術）, 課程博士
バージョン：

Light Field Vision for Transparent Object Recognition

Yichao Xu

Graduate School of Information Science and Electrical Engineering

Kyushu University

This dissertation is submitted for the degree of

Doctor of Philosophy

July 2015

I would like to dedicate this thesis to my wife and parents for their love and encouragement.

Declaration

I hereby declare that except where specific reference is made to the work of others, this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except as specified in the text and Acknowledgements.

Yichao Xu
July 2015

Acknowledgements

This work would not have been possible without a great number of people. I would like to express my sincere gratitude to my advisor Prof. Hajime Nagahara for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I would like to thank the rest of my advisory committee: Prof. Rinichiro Taniguchi and Prof. Yasuhiro Mukaigawa, for their encouragement and insightful comments. Thanks also to Prof. Atsushi Shimada and Prof. Kurazume who provide invaluable feedback in my research. My sincere thanks also goes to Ms. Kiyoko Furuta for her help of many trivial affairs in the research and daily life.

I am also very much obliged to Prof Svinin Mikhail, for offering me the teaching assistant and academic supporter opportunity in International Undergraduate Program and leading me communicating with the students from all over the world.

I thank my fellow lab members in Laboratory of Image and Media Understanding (LIMU): Xing Xu, Toshiki Sonoda, Ngo Thanh Trung, Chao Ma, Zehao Zhang, Hao Liu, Yupeng Zhang, Yao Guo and Yuheng Lu, for the stimulating discussions, and for all the fun we have had in the past years. Also I thank my friends at Kyushu University, Leyuan Liu, Chengming Li, Rong Huang, Xiaojuan Liao, Dong Hao, Jingyu Hua and Laiping Zhao, for all kinds of help in my research and life in Japan.

I also wish to acknowledge Mr. Kazuki Maeno who helped lay the foundations upon which I build today.

Last but not the least, I would like to thank my family for their constant support, and especially to my wife Fang Wang, for her endless loving and supporting my study.

Abstract

The object recognition task in computer vision system is to detect or identify objects in an image. It is an important part of the artificial intelligent systems to know their working environment, and to decide what they can do or cannot do. Transparent object cannot be well recognized by conventional methods that only use the appearance information since the appearance of a transparent object dramatically changes when the background varies. In this work, the author uses a light field camera to observe the visual information of the environment. The light field camera can record light rays from every direction through every point in the 3-Dimensional (3D) world. The captured light field data can help the computer vision system recognize the 3D world more easily, even for the transparent object.

Because the applications require to know the relationship between the recorded light rays and the 3D world, camera calibration is an essential step for the light field acquisition. After calibration, camera parameters are known and ray geometry can be understood. In this thesis, the author propose a calibration method for a camera array and a rectification method for generating a light field image from the captured images. The proposed camera array calibration approach is a two-step algorithm consisting of closed form initialization and nonlinear refinement, which extends Zhang's well-known method to the camera array. More importantly, the author introduce a rigid camera constraint whereby the array of cameras is rigidly aligned in the camera array and utilize this constraint in the calibration. Using this constraint, the calibration process is much faster, and the calibration results are getting more accurate in the experiments.

After obtain the known geometrical information of the captured light field, special features can be extracted from the light field. The feature extracted from the light field image have more advantages than conventional features. A background-invariant feature which is called the light field distortion (LFD) feature is proposed. The LFD feature comes from the transparent object is very different from that comes from the Lambertian object because the linearity in the light field space is different. The light field linearity (LF-linearity) can be used for measuring the likelihood of a point comes from the transparent object or not. And the occlusion detector is designed to locate the occlusion boundary in the light field image.

Recognizing the object category and detecting a certain object in the image are two important object recognition tasks, but previous appearance-based methods cannot deal with the transparent objects. The proposed methods in this thesis overcome previous problems using the novel feature extracted from a light-field image. Transparent object categorization is performed by incorporating the LFD feature into the bag-of-features approach for recognizing the category of transparent object. Transparent object segmentation is realized by solving the pixel labeling problem. An energy function is defined and Graph-cut algorithm is applied for optimizing the pixel labeling problem. The regional term and boundary term are from the LF-linearity and occlusion detector output. Light field datasets are acquired for the transparent object categorization and segmentation. The results demonstrate that the proposed methods successfully categorize and segment transparent objects from a light field image.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Computer vision techniques, which are inspired by theories and observations of visual perception, have been developed rapidly since it appeared in 1966 [1]. The computer vision systems acquire the image, video and multi-dimensional data from the vision sensors, and they can apply the theories and models to solve various problems. Typical computer vision problems include object recognition, scene understanding, video tracking, motion estimation and so on. The solutions to these problems are very useful for the artificial intelligent systems, and computer vision usually plays an important role in the intelligent robotics.

Nowadays, the intelligent systems are not far from our daily life, and they can make our life better. For example, when we use a smart phone to take a photo of a monument as shown in Fig. 1.1a, the computer vision system can tell us the related knowledge of the photo, such as what is the monument for and who made this monument, and then we can know more about what we see. The driverless car will be true with the help of computer vision system as shown in Fig. 1.1b. It will be much safer and more fun when we travel by a driverless car in the near future. Visual object recognition is one of the key parts in these applications. Visual recognition makes the artificial intelligent systems possible to know their working environment, and to decide what they can do or cannot do.

The visual object recognition task is mainly to detect or identify objects in an image. It is easy for human beings to recognize all kinds of objects in the image, no matter how complex the scene is. But for the computers, this task is very difficult. Many approaches, such as histogram-based [4] or feature-based methods [5], have been proposed to deal with this task. However, transparent objects are still an exemption in the object recognition task since the conventional methods which use the appearance information. But the transparent objects do not have their own appearance, they borrow the appearance from the background. When the background changes, the appearance of transparent objects will be totally different (see

(a) Smart phone [2].

(b) Driverless car [3].

Fig. 1.1 Two examples of visual recognition in computer vision applications.

Fig. 1.2). Therefore, the conventional methods cannot recognize same transparent object with different backgrounds.

1.1 Light Field Vision

In order to deal with the transparent object recognition problem, the author introduces a new technique, which is called light field vision to tackle this problem.

As Adelson and Bergen pointed out visual information available to an observer at any point in space and time [6]. Actually, objects can emit or reflect light rays, and we call all the light rays in the space ***light field***. The ***light field*** includes all the visual information in the space.

The visual data is acquired by various vision sensors, and the data can be taken in many kinds of forms. The charge-coupled device (CCD) has been widely used in digital image sensing, because we can use the CCD image sensor to acquire high-quality images and video sequences with low cost. CCD image sensors are easy to use since there are many user-friendly hardware and software available. However, conventional CCD image sensor can only capture sub light field space as shown in Fig. 1.3.

A recently developed vision sensor, which is called light field camera, can capture richer information from the 3-Dimensional (3D) world than conventional cameras. The light field camera can record light rays from every direction through every point in the 3D world. And the captured information can be useful for recognizing the transparent objects with various backgrounds. Therefore, such kind of device is used in this work to tackle the transparent object recognition applications in computer vision. The technique that utilizes light field data to solve computer vision problems is called ***light field vision***.

Fig. 1.2 The appearance of transparent objects are totally different when the background changes.

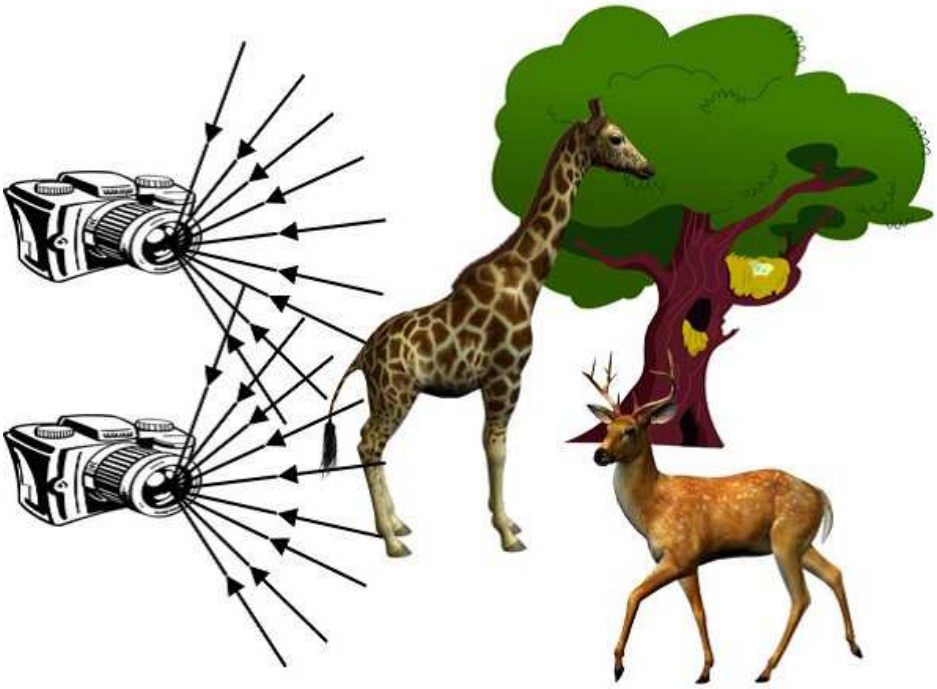
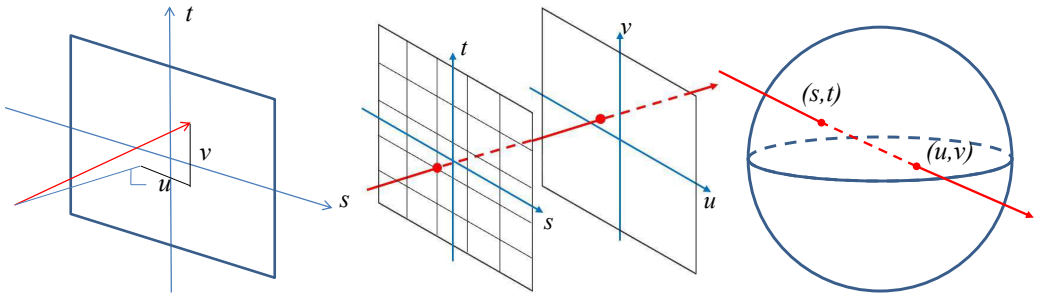


Fig. 1.3 Each viewpoint can only capture sub light field space.



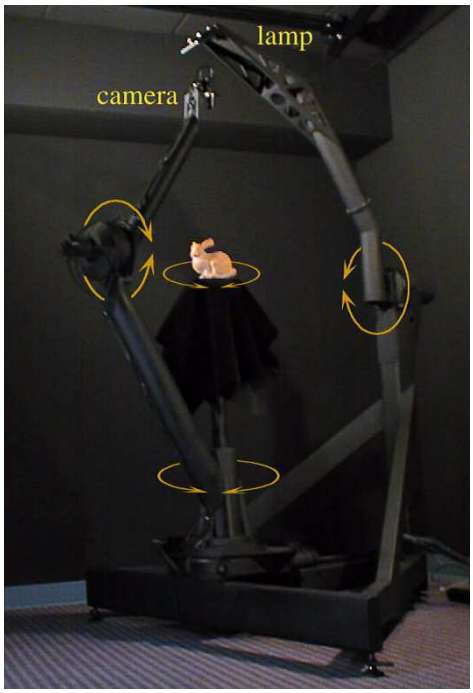
(a) Position-direction (b) Two-plane (c) Sphere

Fig. 1.4 Different parameterizations of the light field.

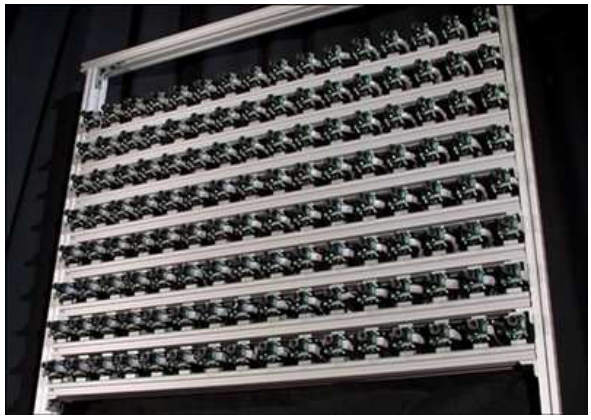
The light rays in the 3D world can be parameterized in the 4D coordinates, and each ray is represented by $L(s,t,u,v)$. There are several different ways to parameterize the light rays, such as position-direction [7], two-plane [8], and spherical [9] parameterizations (see Fig. 1.4). The author uses the position-direction style for the light field representation in this thesis. A 4D light ray is described by the intersection position (s,t) on a plane and its tangential and sagittal angles (u,v).

People use light field camera to capture 4D light rays. The light field camera was originally proposed for image-based rendering for use in the graphics community, and has been used for a variety of different visualization applications, such as generating free-view images, 3D graphics, and digital refocusing. In the early days, light field was obtained by camera on a gantry (Fig. 1.5a). The large cameras array systems was developed in the beginning of 21st century, e.g. Stanford multi-camera array (Fig. 1.5b). These light field camera systems were usually huge and quite expensive, such as the the Stanford spherical gantry costs $130,000 to design and build [10].

Fortunately, recent light field cameras are becoming inexpensive and compact. The small camera array light field acquisition system, Profusion25 (Fig. 1.5c), has already been sold in the commercial market [11]. I believe that the PiCam camera array (Fig. 1.5e), which is suitable for smartphone applications, will be release to the market in the near future. Plenoptic camera which consists of a micro-lens array between the sensor and main lens was first proposed by Ng et al. [12], and a plenoptic 2.0 camera was proposed later by Georgiev et al. [13]. Moreover, such kind of cameras (Fig. 1.5d) have been available in the commercial market now [14, 15]. All these camera systems can capture 4D light rays which are from different positions and directions, but the regular cameras can only obtain position information and integral all the directional information when projecting 3D scene to a 2D image.



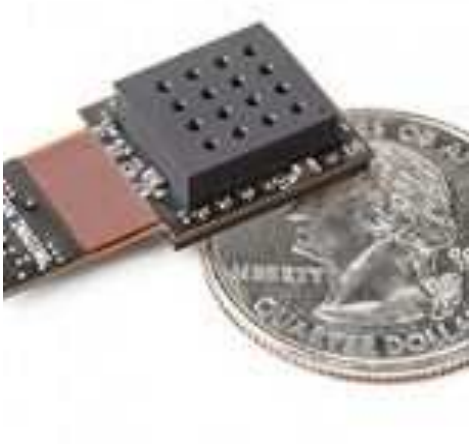
(a) Stanford spherical gantry [8]



(b) Stanford multi-camera array [16]

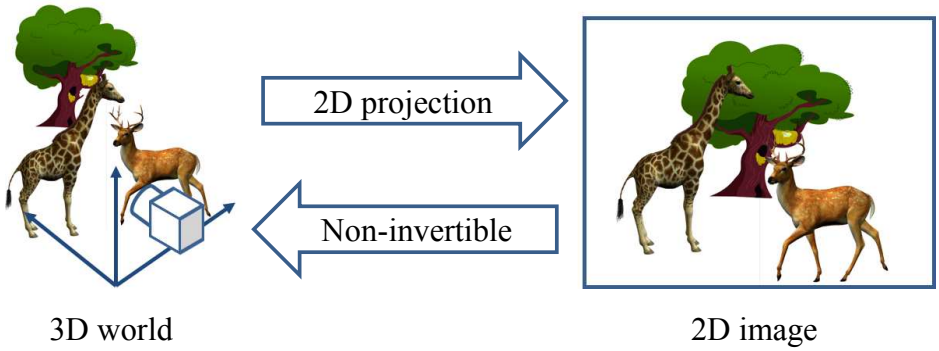
(c) Profusion25 [11]

(d) Lytro Illum [14]

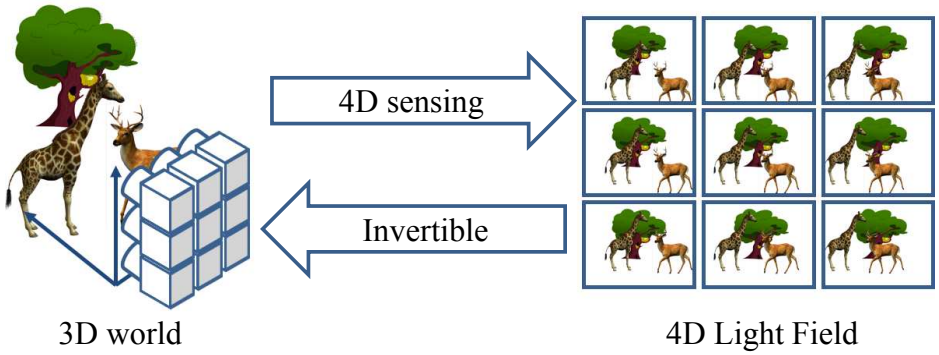


(e) PiCam Camera Array [17]

Fig. 1.5 Light field cameras.



(a) Regular computer vision.



(b) Light field vision

Fig. 1.6 Regular computer vision and light field vision.

Since the data captured by light field cameras has richer information than that captured by conventional cameras, light field cameras are becoming popular in computer vision applications. The comparison of regular computer vision and light field vision are shown in Fig. 1.6. The regular computer vision applications are based on the images captured by single viewpoint camera as shown in Fig. 1.6a. The actual 3D scene is projected to a 2D image. The depth information of the light rays disappear after the projection. Consequently, we cannot know how far is the object from a single image, and it is difficult to recognize objects and scenes in the real 3D world from the image. This work uses a light field camera to capture the data as shown in Fig. 1.6b. The light field image maintain the 2D positional information, and 2D directional information of light rays from the 3D scene. The redundant information makes it easier to understand the 3D world.

Researchers have used light field cameras for computer vision applications in the recent years, such as surveillance [18], consistent depth estimation [19], salience detection [20]. And these applications show that light field vision has better performance than previous computer vision approaches. More examples can be found in Chapter 2 where the author introduces the related work. In particular, this work demonstrates the advantages of light

field vision applies to the transparent object categorization and segmentation from a single light field image, which is not well addressed with regular computer vision approaches. The light field vision can utilize the geometrical relationship between different viewpoints, and overcome the limitations in approaches only use the appearance information in a 2D image. In order to use the geometrical information of the light rays, the relationship between the captured light rays and the corresponding point in the 3D world should be known. This can be done by camera calibration. The author will address the light field calibration problem in this thesis as well.

1.2 Contribution

The following topics are addressed in this thesis:

1. Camera array calibration for light field acquisition;
2. Feature descriptors from light field;
3. Light field vision for transparent object categorization;
4. Light field vision for transparent object segmentation.

The specific contributions of each topic are:
Camera array calibration

- The projection and physical setup of the camera array is modeled, and the model includes the rigid constraint between cameras with different viewpoints;
- Calibration process is performed by a two-step algorithm that combines initial parameter estimation using a closed form solution and non-linear refinement of the parameters using the rigid constraint;
- The captured array images are transformed to light field ray space, which is convenient for light field applications.

Feature descriptors from light field

- Light field distortion (LFD) feature is proposed to describe the deformation of light rays in a light field image;
- Light field linearity (LF-linearity) is proposed to describe the likelihood of a light ray comes from Lambertian object or not;

- Occlusion detector is proposed to describe the light rays are occluded in the reference viewpoint or not.

Transparent object categorization

- A challenge computer vision problem, transparent object categorization, is performed with a single-shot image;
- LFD feature and LF-linearity verified to be effective in the application;
- The performance is evaluated in various conditions.

Transparent object segmentation

- A light field vision method for transparent object segmentation, which is automatic, requiring no human interaction;
- An energy function is defined using the LF-linearity, and occlusion detector;
- Comparisons show that the proposed method obtains better results than previous method based on regular camera [21].

This work partially published in [22–24].

1.3 Outline

Since the light field cameras are becoming popular, several computer vision applications using light field cameras have been proposed recent years. **Chapter 2** reviews different types of light field sensors that can be used for computer vision, and shows the previous computer vision applications in surveillance, depth estimation, salience detection, etc. The related work for transparent objects applications are introduced in **Chapter 2** as well. It reviews methods using special sensors or devices for transparent object 3D reconstruction and detection.

Chapter 3 introduces the light field acquisition approaches. Calibration is an essential step to know the know the relationship between the captured light rays and the 3D world. This work focus on the applications using a camera array. An efficient and accuracy calibration method is proposed for light field acquisition using camera array. The performance of proposed method is analyzed with simulated and real data, and the comparison with other methods are shown in this chapter as well.

After acquiring the calibrated light filed data, the novel features can be extracted from the light field. In **Chapter 4**, novel descriptors are proposed, which are useful for the transparent

object applications. LFD feature is firstly proposed to describe the deformation of light rays in a light field image. And then LF-linearity is proposed to describe the likelihood of a light ray comes from Lambertian object or not. Finally, occlusion detector is proposed to describe the light rays are occluded in the reference viewpoint or not.

The extracted novel descriptors are successfully applied to two transparent object recognition applications. In **Chapter 5**, transparent object categorization method is proposed to answers the question "which category of the transparent object is recognized?". The LFD feature and LF-linearity are successfully applied to transparent object categorization task, and the experimental results show the proposed method using light field image outperforms the conventional method using the image captured by a regular camera. In **Chapter 6**, transparent object segmentation method is proposed to recognize whether there is a transparent object and where it is in the image. The LF-linearity and occlusion detector are successfully applied to transparent object segmentation task, and the experimental results show the proposed method using light field image can get stable and impressive results, which outperforms the conventional method using an image captured by a regular camera.

Chapter 7 draws conclusions and indicates directions of future work.

Chapter 2

Related Work

2.1 Light Field for Computer Vision

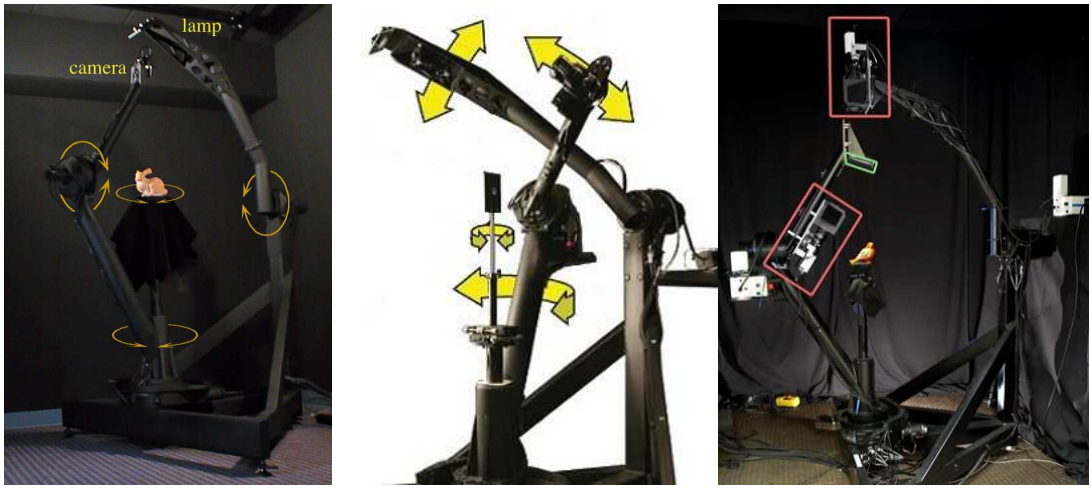
The light field images are used for computer vision applications in this work. The light field acquisition systems are important input devices, and there are various types of systems can obtain light field image. Different applications require different light field images. The light field acquisition systems and the applications for computer vision is reviewed in this section.

2.1.1 Light Field Acquisition Systems

Gantry camera

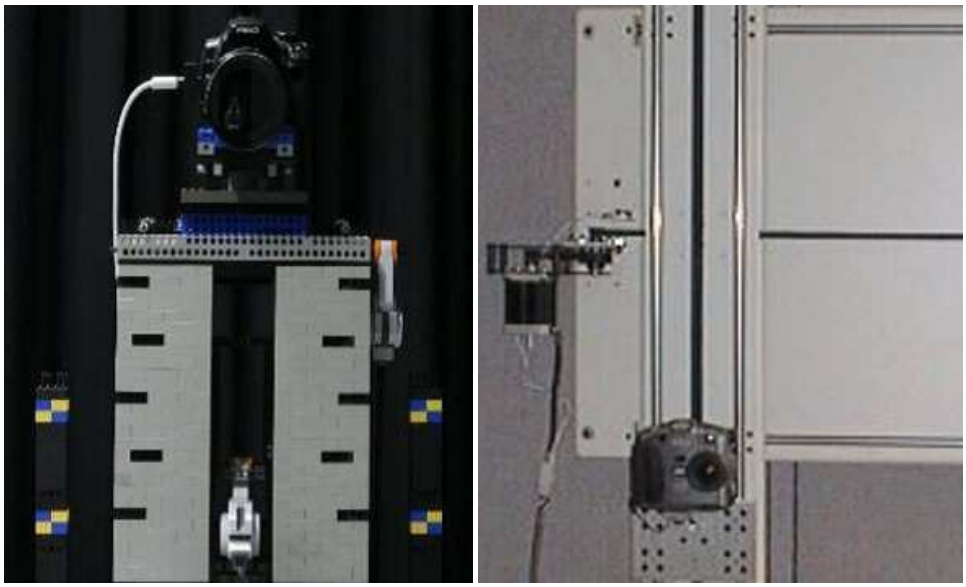
A simple way to acquire the light field is to put a camera on the moving gantry as shown in Fig. 2.1. Researchers at Stanford University first built a gantry (Fig. 2.1a) for light field rendering [8], and the specifications of their gantry are available on the website of Cyberware [25]. Researchers from Cornell University then built a gantry (Fig. 2.1b) that improves the mounting arrangement at the ends of the arms [26]. It makes more flexibility in the lamp and camera that can be attached to these arms. Researchers from University of Virginia have also built a gantry (Fig. 2.1c) with similar design, but the light and camera are coaxially mounted to each of the arms [27].

The cost of a spherical gantry is very high, and it is not worthy to build such expensive equipment only for light field acquisition. Researchers found some simple and inexpensive ways to acquire the light field. The Lego Mindstorms gantry can be used to capture a light field (Fig. 2.2a). We can just move a camera left, right, up, and down on the Lego gantry. And the researchers from MIT graphics group also built a simple vertical XY-table to capture the light field (Fig. 2.2b).



(a) Stanford spherical gantry. (b) Cornell spherical gantry. (c) Virginia spherical gantry.

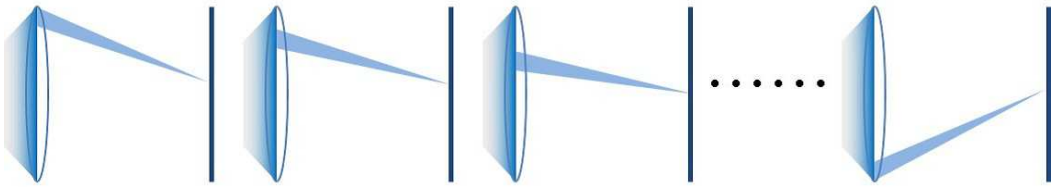
Fig. 2.1 Spherical gantry cameras for light field acquisition.



(a) Lego Mindstorms gantry [28]. (b) A vertical XY-table [29].

Fig. 2.2 Planar gantry cameras for light field acquisition.

(a) Programmable aperture cameras.



(b) Sequential light field acquisition using the coded aperture.

Fig. 2.3 Examples of coded aperture cameras and their sequential sampling mode.

Camera array

Moving a single camera on the gantry can only capture the static scenes. In order to capture dynamic scenes, camera array systems have been developed to acquire the light field. Researchers at Stanford University built several large camera array to perform computational photography applications [16]. These camera array systems allow them to capture light field video. In the recent years, camera array systems are becoming more compact. A camera array with 25 viewpoints, which is called Profusion25, has already been available in the commercial market [11]. It can be easily connected to a Desktop PC or a laptop, which do not require specific control equipment like the large camera array. The latest camera array is even smaller than a coin as shown in Fig. 1.5e. This camera supports both stills and video, low light capable, and it is small enough to be included in the next generation of mobile devices including smartphones [17].

Coded aperture

Light field can also be captured by a camera with a series different aperture shapes. A straightforward way is to blocks all undesirable light rays and leaves a subset of light field to be obtained one by one, as shown in Fig. 2.3b. This can be simply realized by programable aperture cameras [30, 31] as shown in Fig. 2.3a, but the light efficiency is very low because

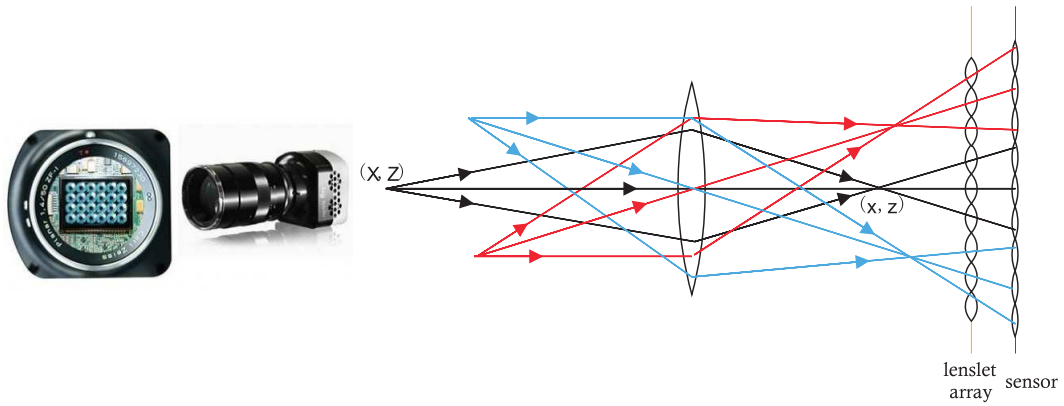


Fig. 2.4 The lenslet-based light field camera and its projection model.

of the small aperture size. In order to overcome this limitation, some well designed aperture shapes is used to capture the images and light field can be recovered by computational methods [32, 33].

Lenslet camera

Similar to the gantry camera, the camera with coded aperture cannot capture dynamic scenes as well. However, if we put a lenslet array in front of the image sensor as shown in Fig. 2.4, the camera can capture the light field with one shot. Ng et al. proposed the first hand-held lenslet camera [12], and this is the prototype of the commercial light field camera Lytro [14]. However, the resolution of first generation lenslet camera is pretty low. The second generation lenslet camera, which is called focused plenoptic camera, has been proposed by Georgiev et al. [7, 13] to increase the image resolution. The focused plenoptic camera is also available on commercial market [15].

2.1.2 Light Field Vision Applications

The light field cameras have already been used in many computer vision applications. Synthetic aperture photography generates multi-focus images from the light field. Digital refocusing [34] that freely changes the focusing position after capturing the image is the most popular application in light field imaging. This technique can be utilized for occluded surface reconstruction [35]. If the synthetic aperture is large enough, occluding objects in front of the focal plane are blurred to the extent that they effectively disappear, and the occluded surfaces can then be seen. It is very useful in surveillance applications. Shimada et al. also used the light field camera for video surveillance application [18]. They proposed a new feature called Local Ray Pattern (LRP) which is used to evaluate the spatial consistency of light rays. The LRP feature and GMM-based background modeling are combined to detect

objects on the selected in-focus plane. Scene geometry reconstruction from multiple views has been an active area of research in computer vision [36] for some time. A light field camera can capture multiple viewpoint images simultaneously, so that the 3D geometry can be recovered efficiently from a single image. The depth map estimated from a light field image is more accurate and consistent than conventional methods [19]. Recently, light field images are also used to detect the salience and get better performance than single-view based methods [20].

2.2 Conventional Methods for Transparent Object

Although transparent object is very difficult to deal with in computer vision applications, researchers have utilized many special approaches to tackle the problems with transparent object. There has been much research on measuring refraction responses in transparent objects using cameras to obtain physical parameters, such as surface curvature or refractive index. It is well known that refraction polarizes light. Miyazaki et al. measured light intensities from transparent objects through polarizing filters [37, 38]. Schlieren photography [39, 40] has also been used for fluid, gas flows, and shock wave analysis. This method visualizes the refraction response in a scene as a gray-scale or color image by using special optics, although it requires high-quality optics and precise alignment. Hence, its applicability is restricted to laboratory environments, and not for common practical use. Multiple video cameras are used to build passive tomography system to measure turbulence strength of the invisible fluid [41]. Underwater cameras are often suffer from the distortion caused by random waves in the water-air interface. A special sensor is designed to deal with such kind of distortion for the underwater imaging [42].

Recent years, light field also plays an important role in reconstruction of transparent surfaces. Wetzstein et al. [43] proposed light-field background-oriented Schlieren photography that obtains Schlieren photos using a common hand-held camera and a special-purpose optical sheet which is called light field probe (LF-probe). By using this technique, the transparent surface can be reconstructed [44]. Similarly, Ji et al. [45] also utilized the LF-probe and multiple viewpoints to reconstruct the invisible gas flow. Although this technique can reconstruct the the transparent surface and invisible gas flow, it also has restricted practical use as the LF-probe is always required as a background object. Ding et al. [46] proposed a method to acquire the dynamic 3D fluid surfaces. They used camera array as the capture device and checkerboard pattern as the background. Ye et al. [47] proposed an approach to acquire the dynamic 3D fluid surfaces with a single camera, but they utilized a special

background which is called Bokode to emulates a pinhole projector for capturing ray-ray correspondences.

Similar to the target applications of this work, learning-based method [21, 48] has been proposed for finding glass in a single view image. Fritz et al. [49] used SIFT feature and L-DA for learning a transparent object and detecting its location and region as a bounding box. Wang et al. [50, 51] used RGB-D image for glass object segmentation. The depth image was utilized as one of the cues for transparency that the depth information is missing in the glass region, since the glass refracts the active light from the sensor. For multi-view images as input, the epipolar-plane-image (EPI) analysis method was used to extract layers with specular properties [52]. Approaches, similar to this work, obtain shape from optical flow caused by refraction. In particular, Ben-Ezra et al. [53] proposed a model-based method to recover shape and pose from video taken with known camera motions. Similarly, Aagrwal et al. [54] recovered shape from video acquired while the background behind the object moves. Morris et al. [55] used two calibrated cameras to estimate the refractive indices over time-varying liquid surfaces from distortions of known grid patterns at the bottom of a tank. In contrast to these approaches, the novelty of this work is to apply refraction to transparent object categorization and segmentation realized from a single shot image, using a light field camera as an input device. Unlike previous methods, there are no constraints on background texture, camera motion or known parameters. And the proposed method also has the potential for glass and specular objects.

2.3 Summary

In this chapter, different types of light field cameras are reviewed first, and then the author shows the previous computer vision applications using light field camera for surveillance, depth estimation, salience detection, etc. The related work for transparent objects applications are introduced as well.

Chapter 3

Camera Calibration for Light Field Acquisition

As introduced in the previous section, light field can be captured by many types of cameras. Some light field acquisition systems are made by researchers themselves, and some of them can be bought from the commercial market. No matter using what kind of camera to acquire light field images, the calibration is an essential step in computer vision applications. After calibration, the relationship between the captured light rays become known. In this work, relatively large disparities is needed to describe the distortion of the backgrounds, and the system can be working under dynamic scenes. Therefore, relatively large baseline is preferred, and a camera array is used as input device in the applications.

3.1 Background

Over the past few decades, a great deal of work has been done on camera calibration to acquire camera parameters with high accuracy. There are several camera calibration approaches including the single camera, multi camera, and structure from motion (SfM), which can be directly applied to the camera array, approaches.

Single camera calibration: Classic camera calibration is performed by observing a 3D reference object with a known Euclidean geometry [56]. This type of approach requires specialized and expensive equipment with an elaborate setup. To overcome these disadvantages, a flexible technique for single camera calibration was proposed by Zhang [57], which requires the camera to observe a planar pattern displayed at a minimum of two different orientations only. The pattern can simply be printed using a laser printer and then attached to a "reasonable" planar surface (e.g., a hard book cover). Either the camera or the planar

pattern can then be moved by hand. The specific motion need not be known. Although this technique is very practical and robust for a single camera, it is not suitable for a light field camera. The rigid transformations between any pair of viewpoints, which can be determined using any captured frame, should be invariant irrespective of the frame by which they were computed. Unfortunately, these transformations are inconsistent when each viewpoint is calibrated independently (see Fig. 3.2a). This inconsistency results in inaccurate estimation of the relative translation between the viewpoints, potentially leading to serious problems if used with light field cameras. To avoid the inconsistency, the model with a rigid constraint between the viewpoints (see Fig. 3.2b) has been adopted in this work.

Multi-camera calibration: Because multi-camera systems are becoming less expensive and more useful, there are increasing requirements for multi-camera calibration. Stereo camera is the simplest multi-camera system, and calibration methods utilizing different constraint were proposed for stereo calibration. Horaud et al. [58] proposed a method for recovering camera parameters from rigid motions. This method relies on linear algebraic techniques and requires the epipolar geometry. Malm and Heyden proposed a method [59] which extends Zhang's single camera calibration method, and also utilizing a planar object. Several methods have been developed to deal with multi-camera systems. Vaish et al. [60] proposed a method using a plane plus parallax to calibrate a multi-camera array for light field acquisition. Assuming that the images of the light field were aligned on some reference plane in the world, they were able to measure the parallax of some points in the scene not lying on this reference plane. This method, however, assumes that all cameras lie on a plane parallel to the reference plane, and the projection to the reference plane must be calculated in advance. This calibration technique can only be used with cameras well on the assumed plane and applications that do not require accurate parameters. Svoboda et al. [61] proposed a method for multi-camera system calibration using point light source. They captured image sequences of the multi-camera while point light source moving in a working volume. The method used the factorization method for solving projective matrices as well as the light source positions from the sequences. Ueshiba et al. [62] proposed a method that uses a planar checkerboard pattern like the proposed method. They calculate homography matrices between the calibration chart and the images captured by the multi-cameras, then also apply the factorization method for estimating checkerboard chart positions and the projection matrices from the homographies. These methods used factorization method, but it is commonly time consuming algorithm. Zhang's method is chosen for estimating the initial parameters in this work, since it is simpler and less computational method and the accuracy is satisfactory to make the re-projection error converge to minima in the refinement step.

Structure from motion: SfM techniques aim to reconstruct simultaneously the unknown 3D scene structure and camera positions and orientations from a set of feature correspondences. Related methods such as bundle adjustment have made their way into computer vision and are now regarded as the gold standard for performing optimal 3D reconstruction from correspondences [63]. Bundler [64] is a popular tool for SfM. It can also estimate camera parameters from multi-images by bundle adjustment. Bundler was designed for applying either a moving camera or multiple cameras. It has great flexibility in that each viewpoint of the captured image can be freely moving. There are no constraints on camera positions and it independently estimates the multi-camera parameters. As a result of the flexibility, Bundler wastes computation and loses calibration accuracy by not using a rigid camera constraint. The comparison of the accuracy and computational cost between Bundler and the proposed method will be shown in the experiments in Section 3.5.

Previous calibration methods for single cameras [56, 57], multi-cameras [61] and SfM [64] did not include a rigid constraint. I incorporate the use of a rigid constraint derived from the architecture of the camera array to improve accuracy and reduce the relative computational cost.

Similar to the conventional methods [65, 66], rectification can be performed when the calibrated parameters are obtained. There are also approaches for rectifying the light field image without explicitly calibrating the camera parameters [67, 68]. In this work, the author rectify and project the captured multiple images to light field space by calibrated parameters.

3.2 Light Field Camera Model

The author will start by modeling the projection of a single camera which can be moving on the gantry, and then extend this model to a camera array with a rigid constraint.

3.2.1 Gantry Camera

The gantry camera captures the light field by fixing a single camera on the moving gantry. The single camera model is enough for get all the parameters since the relationship between different viewpoints can be directly obtained from the gantry movement.

The projection model for a single camera is shown in Fig. 3.1. A 3D point $\mathbf{M} = [X, Y, Z]^T$ in the global coordinate system can be projected onto a 2D point $\mathbf{m} = [x, y]^T$ on the ideal image plane via a rotation $\mathbf{R}$ and translation $\mathbf{t}$,

$$s \begin{bmatrix} \mathbf{m} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} \mathbf{M} \\ 1 \end{bmatrix} \tag{3.1}$$

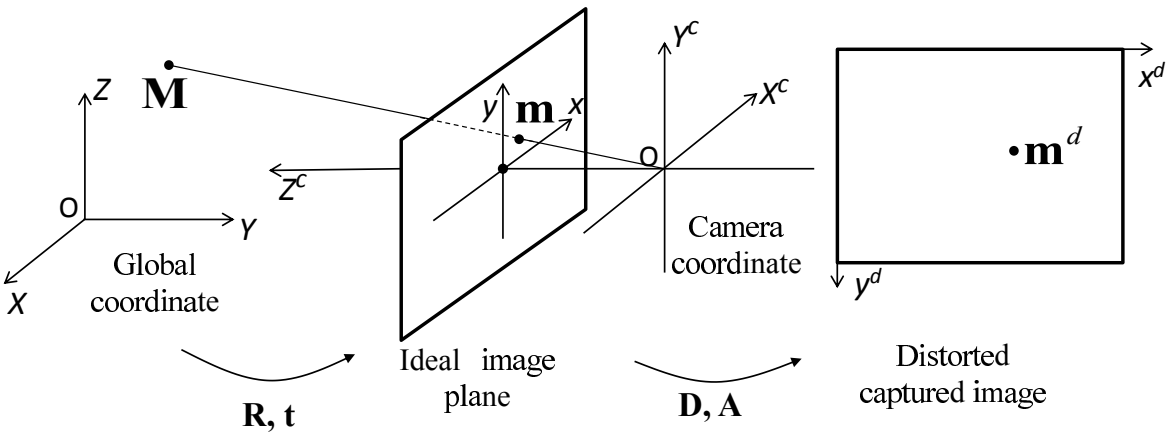


Fig. 3.1 Projection model for a single camera.

where s is an arbitrary scale factor.

Camera lenses usually suffer from distortion, particularly radial distortion and slight tangential distortion. Therefore, lens distortion should be carefully considered during the projection step. According to previous works [69, 70], the 2D point $\mathbf{m}^d = [x^d, y^d]^T$ after distortion in the captured image is modeled as

$$s \begin{bmatrix} \mathbf{m}^d \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x(1+d_1r^2+d_2r^4)+2d_3xy+d_4(2x^2+r^2) \\ y(1+d_1r^2+d_2r^4)+d_3(2x^2+r^2)+2d_4xy \\ 1 \end{bmatrix}, \tag{3.2}$$

where $r^2 = x^2+y^2$, d_1 and d_2 are the radial distortion coefficients, and d_3 and d_4 are the tangential distortion coefficients. $\mathbf{A}$ is called the intrinsic matrix, which is given by

$$\mathbf{A} = \begin{bmatrix} \alpha & \gamma & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

and contains the coordinates of the principal point (c_x, c_y), the focal lengths α and β in the horizontal and vertical directions, respectively, and the skew of the two image axes, γ.

$\mathbf{D}$ is used to represent the distortion coefficients $[d_1, d_2, d_3, d_4]$ for short. If there is no distortion, $\mathbf{D} = \mathbf{0}$ and Eq. (3.2) defines the linear case:

$$s \begin{bmatrix} \mathbf{m}^d \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{m} \\ 1 \end{bmatrix}. \tag{3.4}$$

These days the image axes of charge-coupled device (CCD) sensors are almost perpendicular. As suggested in [57], I also assume γ=0 in the model.

3.2.2 Camera Array

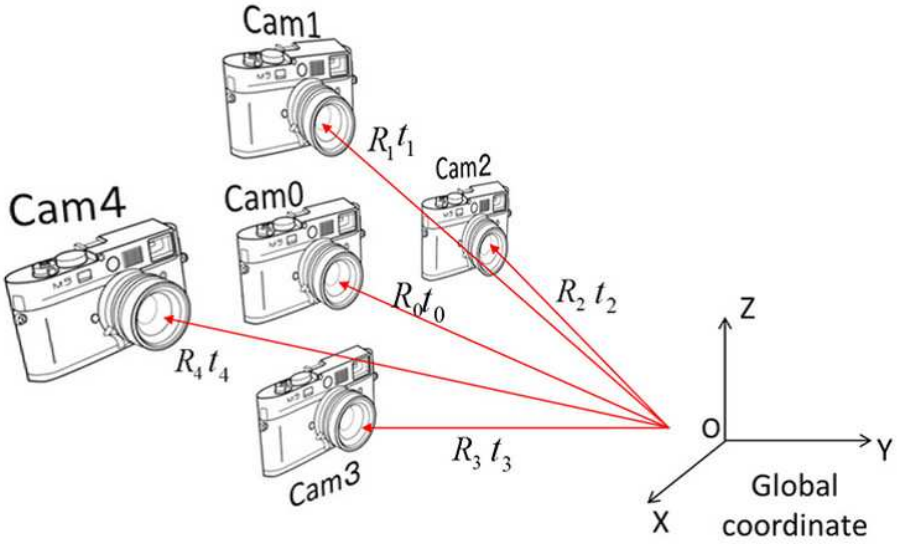
Let us consider a camera array system that contains N cameras, such that the light field camera can simultaneously capture images from N viewpoints. Each viewpoint independently records its own 2D image from its position. If the single camera model is simply applied to the camera array, it becomes the model without a rigid constraint as shown in Fig. 3.2a.

However, a camera array has the physical constraint that each viewpoint is rigidly aligned with every other and the relative positions and rotations of the viewpoints do not change from one frame to the next. The author refer to this constraint as the rigid constraint in this work. A projection model is introduced with an array coordinate system for the rigid constraint as shown in Fig. 3.2b. In this model, $\mathbf{R}$ and $\mathbf{t}$ denote, respectively, the rotation and translation of the camera array. Thus, $\mathbf{R}$ and $\mathbf{t}$ describe the rotation and translation of the entire camera array system in the global coordinate system at each frame.

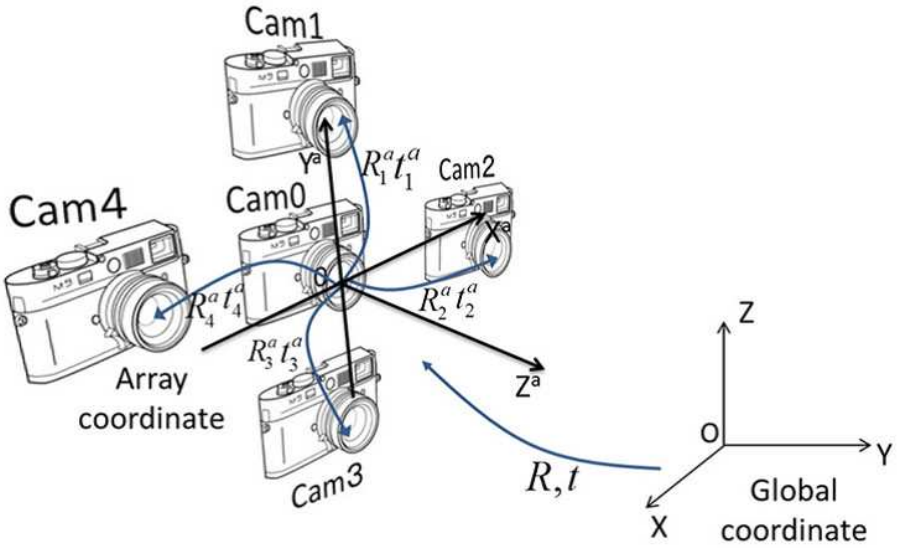
Contrarily, $\mathbf{R}_i^a$ and $\mathbf{t}_i^a$ denote, respectively, the relative rotation and translation of the i-th viewpoint in the camera array coordinate system. $\mathbf{R}_i^a$ and $\mathbf{t}_i^a$ are fixed for all frames, since the relative positions and translations do not change from one frame to the next. Hence, the author splits the extrinsic parameters of the single camera into camera array motion $[\mathbf{R} \quad \mathbf{t}]$ and relative motion $[\mathbf{R}^a \quad \mathbf{t}^a]$ in order to introduce the rigid constraint. Obviously, if the camera array coordinate system is set to the camera coordinate system of viewpoint 0, $\mathbf{t}_0^a = \mathbf{0}$ and $\mathbf{R}_0^a = \mathbf{I}$. Thus, the array coordinate system is equivalent to one of the camera coordinate systems and the other viewpoints are described relative to the assigned viewpoint. The relation between the extrinsic parameters of single camera model $[\mathbf{R}_i \quad \mathbf{t}_i]$ and those of the proposed model using camera array coordinates $[\mathbf{R} \quad \mathbf{t}]$ and $[\mathbf{R}_i^a \quad \mathbf{t}_i^a]$ can be formulated as:

$$[\mathbf{R}_i \quad \mathbf{t}_i] = [\mathbf{R}_i^a \quad \mathbf{t}_i^a] \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{3.5}$$

Once we have the extrinsic parameters $[\mathbf{R}_i \quad \mathbf{t}_i]$, the intrinsic matrix $\mathbf{A}_i$, and the distortion coefficients $\mathbf{D}_i$, the projection of a 3D point $\mathbf{M} = [X,Y,Z]^T$ to a 2D pixel $\mathbf{m}_i^d = [x_i^d, y_i^d]^T$ in the captured image of the i-th viewpoint can be computed from the single camera model using Eqs. (3.1), (3.2), and (3.3). An overview of the capturing process is illustrated in Fig. 3.6.



(a) Camera model without rigid constraint



(b) Camera model with rigid constraint

Fig. 3.2 Camera array based light field acquisition system geometry.

Until now the light field camera model with the rigid constraint between viewpoints has been constructed, while the unknown parameters can be solved using the linear closed form and refined by nonlinear optimization.

3.3 Camera Array Calibration

There are two steps in the proposed method for parameter calibration. The author first calculate the initial values using the closed form solution, and then a non-linear iterative algorithm is applied to refine the initial values. The distortion coefficients, initially set to zero, are refined in the optimization step.

3.3.1 Closed form solution

By applying Zhang's calibration method [57] to each viewpoint, closed form solution can be obtained for the intrinsic matrix $\mathbf{A}_i$ and the extrinsic parameters $[\mathbf{R}_i \quad \mathbf{t}_i]$. The extrinsic parameters of the *i*-th viewpoint in the array coordinate system can be then computed from the inverse form of Eq. (3.5).

$$[\mathbf{R}_i^a \quad \mathbf{t}_i^a] = [\mathbf{R}_i \quad \mathbf{t}_i] \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}^{-1} \tag{3.6}$$

Theoretically, the extrinsic parameters in the array coordinate system should be the same for every frame. However, the results vary for different frames in the presence of noise, since closed form solutions are computed independently for each viewpoint. To obtain reasonable initial values, the author calculates extrinsic parameters in the array coordinate system for all captured frames and then choose the median values.

3.3.2 Global optimization

Thus far, the intrinsic and extrinsic parameters have been obtained through a series of linear methods. However, these parameters are not optimal. Moreover, the linear methods cannot deal with lens distortion. As suggested in [57, 62], nonlinear optimization is needed to refine the linear solutions.

Suppose that the total number of viewpoints is N. We capture T frames of a model plane with K points on this model plane. Assuming that these image points $\mathbf{m}_{ijk}$ are corrupted by independent and identically distributed noise, the maximum likelihood estimation of the

<table>
<tr><td rowspan="2">Intrinsic Parameters
(4+4) x Cameras</td><td>Extrinsic Parameters of Camera Array
(In the global coordinate,
change for different frames)
6 x Frames</td></tr>
<tr><td>Relative Extrinsic Parameters between the viewpoints
(In the array coordinate,
keep same for different frames)
6 x (Cameras-1)</td></tr>
</table>

Fig. 3.3 Parameters for optimization.

intrinsic and extrinsic parameters can be obtained by minimizing the following function:

$$||\varepsilon||^2 = \sum_{i=0}^{N-1}\sum_{j=0}^{T-1}\sum_{k=0}^{K-1} ||\mathbf{m}_{ijk} - \mathbf{E}_{ijk}||^2, \tag{3.7}$$

where $\mathbf{E}_{ijk} = \hat{\mathbf{m}}(\mathbf{A}_i, \mathbf{D}_i, \mathbf{R}_i^a, \mathbf{t}_i^a, \mathbf{R}_j, \mathbf{t}_j, \mathbf{M}_k)$ is the estimated projection of point $\mathbf{M}_k$ in the j-th frame of the i-th viewpoint, $\mathbf{D}_i$ represents the distortion coefficients of the i-th viewpoint $[d_{1i}, d_{2i}, d_{3i}, d_{4i}]$, $\mathbf{R}_i^a$, $\mathbf{t}_i^a$ represent the extrinsic parameters of the i-th viewpoint in the array coordinate system, and $\mathbf{R}_j$, $\mathbf{t}_j$ represent the extrinsic parameters of the j-th frame for the camera array in the global coordinate system. ε is a vector composed of the re-projection error from each point $\varepsilon_{ijk} = \mathbf{m}_{ijk} - \mathbf{E}_{ijk}$.

Minimization is performed using the Levenberg-Marquardt algorithm [71, 72], which is initialized with the linear solution obtained from the closed form. The Levenberg-Marquardt algorithm is an improved version of Gauss-Newton algorithm, and the normal equations in Gauss-Newton algorithm are replaced by the augmented normal equations $(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I})\Delta = -\mathbf{J}\varepsilon$. $\mathbf{J}$ is the Jacobian matrix at certain values of the parameters, and $\mathbf{I}$ is the identity matrix. λ is a non-negative damping factor which varies from iteration to iteration, and Δ is a vector composed of the parameter increment.

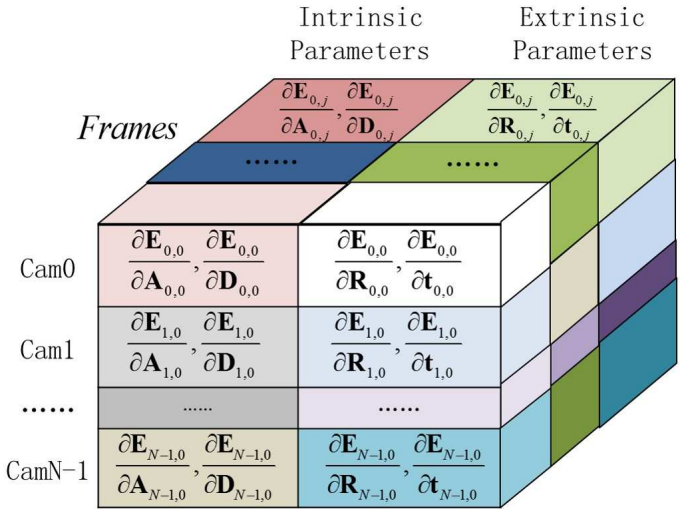
Optimization can be carried out for all parameters simultaneously, including the intrinsic matrices, the distortion coefficients, and all extrinsic parameters. The components of the parameters for optimization are shown in Fig. 3.3. The implementation of proposed method is very flexible in that there is no limit on the number of viewpoints, and the intrinsic parameters are optional for users at runtime.

As we see from the augmented normal equations in Levenberg-Marquardt algorithm, updating the Jacobian matrix is the major work at each iteration. Fig. 3.4 illustrates the structure of the Jacobian matrix. The left and right columns are related to the intrinsic

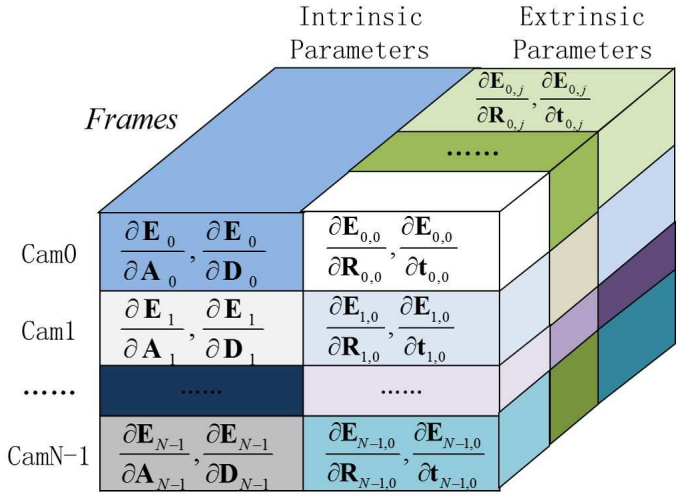
Algorithm 1 Global optimization

```
repeat
    for all viewpoints do
        for all frames do
            Compose the extrinsic parameters in the global coordinate system for each viewpoint
            Compute the re-projection error for the current viewpoint and frame
            Calculate the derivative of the extrinsic parameters of the camera array
            Update the elements of the Jacobian matrix related to the extrinsic parameters of the camera array
        end for
        Compute the projection error of all frames for the current viewpoint
        Update the elements of the Jacobian matrix related to the extrinsic parameter in the array coordinate system
        Update the elements of the Jacobian matrix related to the intrinsic parameters(optional)
    end for
    Compute the total re-projection error of all viewpoints and frames
    Launch the Levenberg-Marquardt algorithm to update all parameters
until termination criteria are met
```

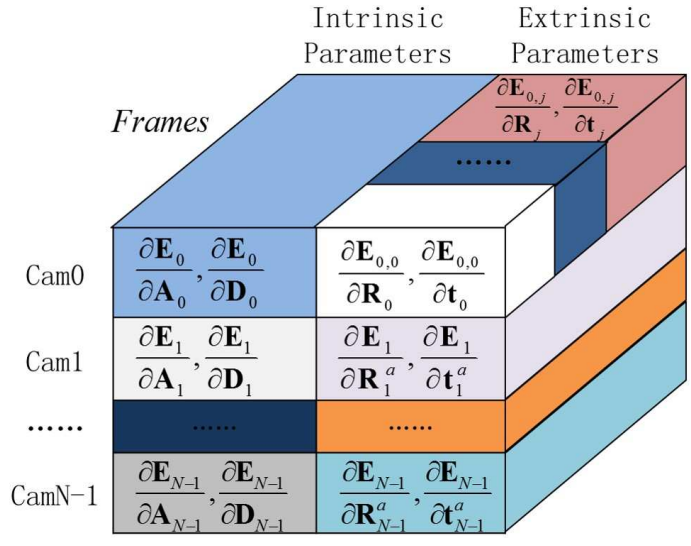
and extrinsic parameters, respectively. Each row represents a different viewpoint, while the z-axis is related to time, representing different frames. The values in each cell are the derivative of estimated points $\mathbf{E}_{ij}$ with respect to the corresponding parameters. Here the size of the checkerboard pattern is known, so the 3D point $\mathbf{M}_k$ is the known parameter which is unnecessary to estimate. The author assumes that the parameters are independent and have rigid constraints. Fig. 3.4c shows the Jacobian matrix with the rigid constraint used in the proposed method. The extrinsic parameters in the array coordinate system remain unchanged for each viewpoint when the frame changes, and the intrinsic parameters are also fixed for each viewpoint. On the contrary, Fig. 3.4a and Fig. 3.4b show the Jacobian matrix without the rigid constraint. The structure used in a method like Bundler [64] is shown in Fig. 3.4a. In this case, the values in each cell need to be updated during iterations of the refinement process. Fig. 3.4b can be considered to be a combination of the single camera refinements. Each viewpoint has fixed intrinsic parameters, but still does not include a rigid constraint between viewpoints, which will cause inconsistency. Compared with the other two structures, we can see that the proposed method is better suited to a camera array and requires less computation during optimization. As previously discussed, the main task of Algorithm 1 is to update the Jacobian matrix and then pass it to the Levenberg-Marquardt procedure.



(a) Jacobian with full freedom



(b) Jacobian without rigid constraint



(c) Jacobian with rigid constraint

Fig. 3.4 Structure of the Jacobian matrix for optimization.

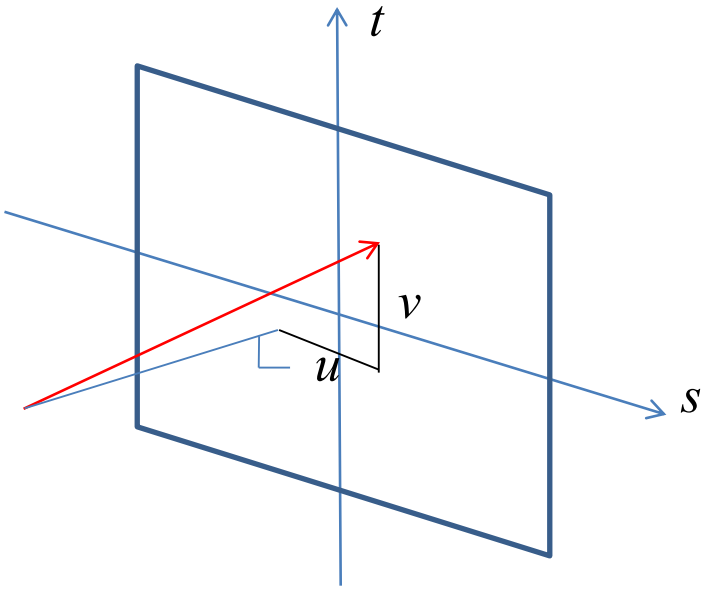


Fig. 3.5 Light field representation.

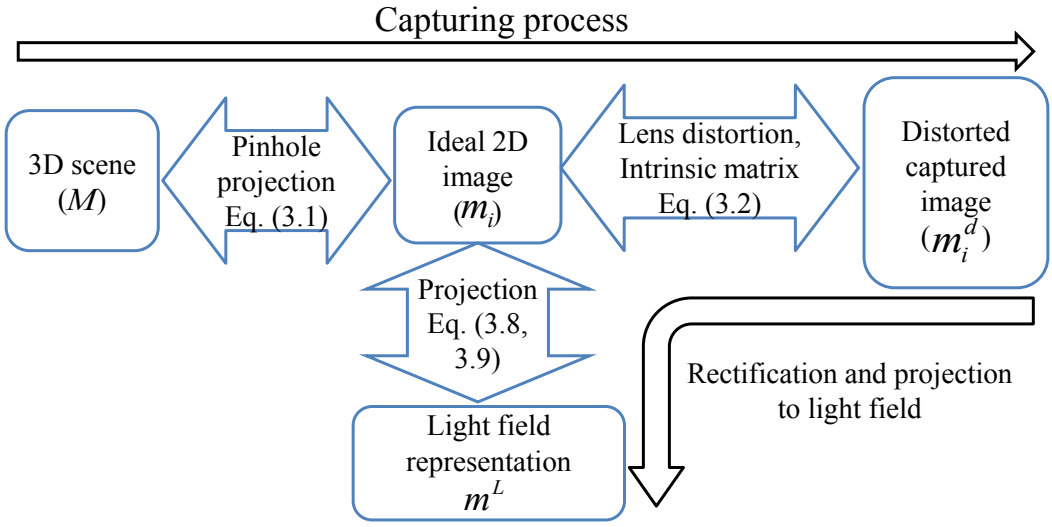


Fig. 3.6 Relationship between the different spaces.

3.4 Light Field Representation and Projection

There are several different light field representations, such as two-plane [8], spherical [9], and position-direction [7] parameterizations. The author uses the position-direction style for the light field representation (Fig. 3.5). A ray is described by a 4D vector (s,t,u,v) as the intersection of plane (s,t) and tangential and sagittal angles (u,v) in this position-direction representation. In this section, the process of rectification and projection from the

N captured images $\mathbf{m}_i^d$ on the camera array to the 4D image point $\mathbf{m}^L = [s,t,u,v]^T$ on the light field is described.

The image captured from the camera array are independently recorded by different viewpoints as shown in Fig. 3.7a. It is not convenient to use the captured images directly for light field applications. The captured images need to be transformed to light field ray space, so that each ray can be represented by a 4D vector (s,t,u,v) as shown in Fig. 3.7b.

The intrinsic and extrinsic parameters are estimated as described in Section 3.3. The projective relation between the captured image space $\mathbf{m}^d$ and the ideal image space $\mathbf{m}$ can be obtained by using Eq. (3.2) and the intrinsic parameters for each viewpoint as shown in Fig. 3.6. Here the author will describe the projection from the ideal image space $\mathbf{m}_i$ to the light field space $\mathbf{m}^L$.

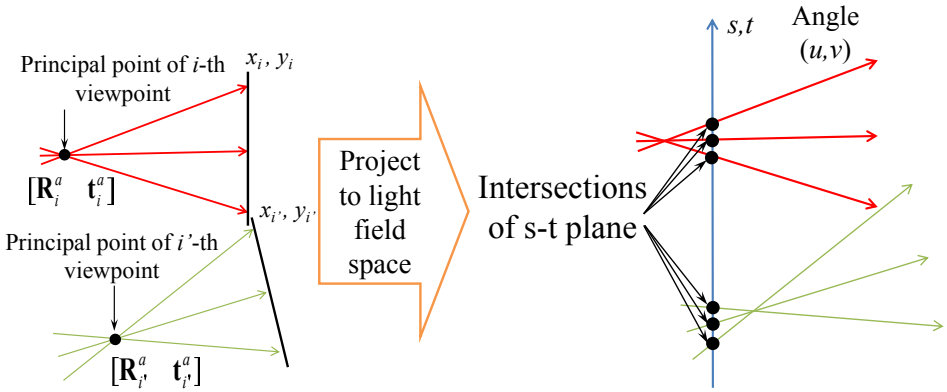
Lens distortion and intrinsic parameter effect, such as difference in focal length, have been rectified in the ideal image space. However, the principal axes of the viewpoints are not yet aligned. To parallelize these and project image point $\mathbf{m}_i$ on each viewpoint into the same (u,v) space, the inverse of the rotation matrices is applied to the ideal image.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = (\mathbf{R}_i^a)^{-1} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{3.8}$$

There is no guarantee that the projective center of the viewpoints are aligned on the same plane or the s-t plane, even though the viewpoints in the camera array are carefully and precisely aligned. In addition, since there is a camera array that is arranged as a dome, the intersection of s-t plane (s, t) have to be calculated as shown in Fig. 3.7. We can obtain the projection from the point on each viewpoint (x_i,y_i) with an estimated translation vector of the viewpoint $\mathbf{t}_i = [t1_i,t2_i,t3_i]^T$, described as follows.

$$\begin{bmatrix} s \\ t \\ 1 \end{bmatrix} = t3_i^a \cdot (\mathbf{R}_i^a)^{-1} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} + \begin{bmatrix} t1_i^a \\ t2_i^a \\ 1 \end{bmatrix} \tag{3.9}$$

Using the projective relation in Eqs. (3.2), (3.8), and (3.9) and calculating the corresponding intensity from the captured image $I(x_i,y_i)$ by interpolation, the light field image $I(s,t,u,v)$ can be obtained from the multiple captured images as shown in Fig. 3.8.



(a) Separated camera coordinates (b) Unified light field coordinates

Fig. 3.7 Viewpoint projection.

3.5 Calibration Results and Analysis

The proposed algorithm has been tested on both computer simulated data and real data.

3.5.1 Simulation results

The author performed simulation experiments for a 25-viewpoint (5 horizontal$\times$ 5 vertical) light field camera. Fig. 3.9 shows the configuration of the simulation. The resolution for each viewpoint is 640×480 pixels. There is an interval of 10 mm between neighboring viewpoints, and it is assumed that all the viewpoints are on the same plane. The simulated light field camera has the following intrinsic parameters for all viewpoints: $\alpha = \beta = 700$ pixels, $c_x = 320, c_y = 240$. The author simulated 10 frames for the system, with each frame painted with $7 \times 10 = 70$ reference points at 20 mm intervals. The distance and orientation of the frames were varied in the simulation. Independent Gaussian noise with 0 mean and σ standard deviation (noise level) was added to the simulated image points. The estimated camera parameters were then compared with the ground truth. For each noise level, 100 independent trials were conducted, the average results of which are shown in Fig. 3.10 and Table 3.1.

The author measured the errors with respect to the ground truth for focal lengths α and β for all the viewpoints, as well as the principal point c_x and c_y. The noise level varied from 0.2 pixels to 1.8 pixels. The author compared the results with linear closed form, Zhang's method [57], which refines each viewpoint independently, and the state-of-art bundle adjustment tool, Bundler [64]. As can be seen in Fig. 3.10, the errors in the estimated intrinsic parameters increase with the noise level. However, the proposed method yields much better results than the other methods, because the proposed method is constrained by a geometric

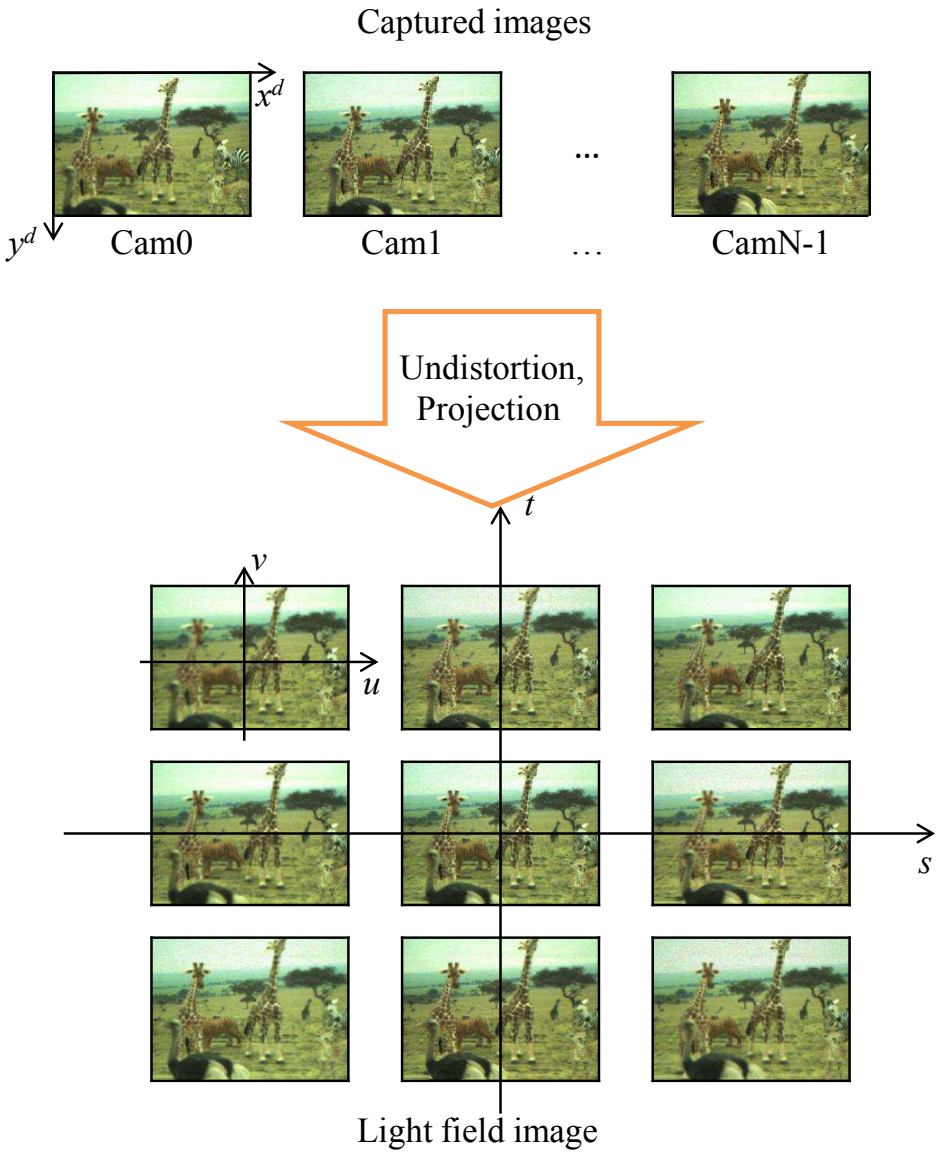
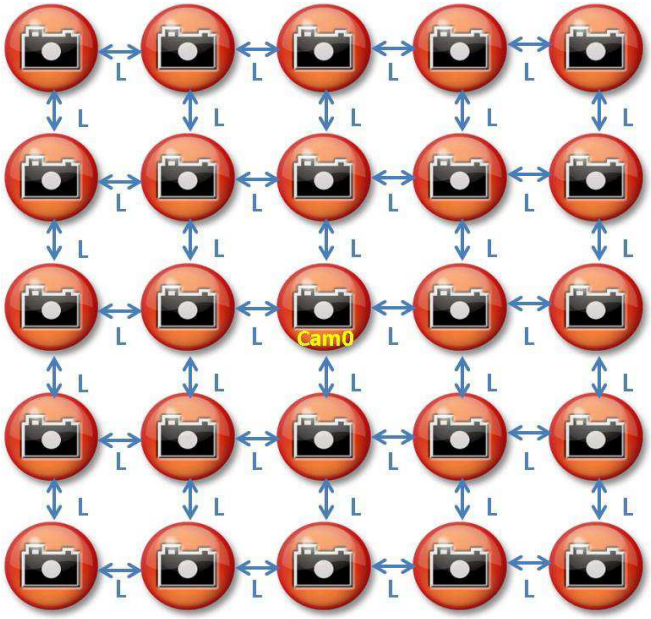


Fig. 3.8 Light field representation in 4D space.

relationship between the viewpoints and therefore suffers less from over-fitting the noise in a single viewpoint. The input for Bundler consists of the matched points and initial estimation of focal length from the linear closed form. Bundler estimates the intrinsic and extrinsic parameters for every frame and adjusts them to minimize the re-projection error. The error in the estimated focal length by Bundler increased so rapidly that a large scaled axis (on the right) is needed to plot Fig. 3.10a. Since the linear method and Bundler always set the principal point at the center of the image, the author did not plot these two methods in Fig. 3.10b.

In Fig. 3.11, the author compared the error in the extrinsic parameters. In Table 3.1, the



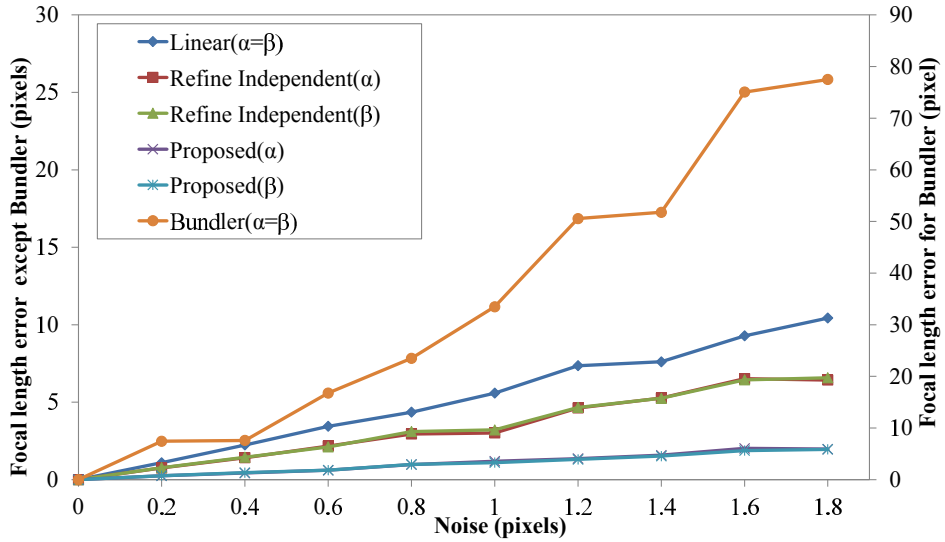
L=10mm **Resolution:** 640x480
Focal length: $\alpha = \beta = 700$
Principal Point: $c_x = 320$, $c_y = 240$

Fig. 3.9 Simulation setup.

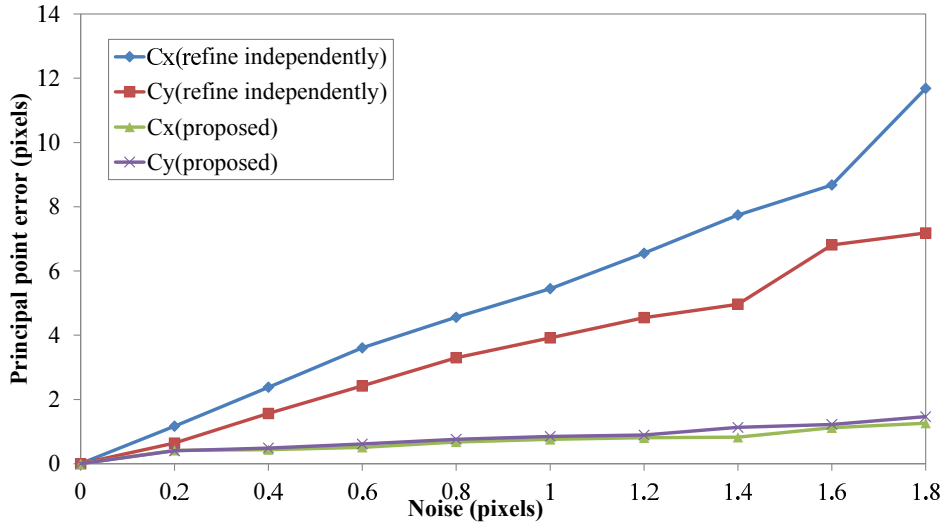
Table 3.1 Comparison of parameter errors.

Method	Focal length error (pixels)	Focal length deviation (pixels)	Translation error (Euclidean distance) (mm)	Translation deviation (Euclidean distance) (mm)	Rotation error (Frobenius norm)	Rotation deviation (Frobenius norm)
Linear	1.12	0.79	0.64	0.48	0.0009	0.0004
Independently refined	0.84	0.50	0.62	0.29	0.0026	0.0013
Bundler	9.12	32.90	8.27	25.27	0.0045	0.0047
Proposed	0.43	0.24	0.12	0.07	0.0005	0.0002

author also shows the comparison of the parameter errors with a 0.2 pixel noise level. The relative translation, rotation errors and focal length were calculated from all the viewpoints and frames. The author computed the errors with respect to the corresponding ground truth and then averaged them. The translation error is the Euclidean distance with respect to the ground truth, while the rotation error is the Frobenius norm. As expected, the errors in the estimated extrinsic parameters increase with the noise level, and the proposed method

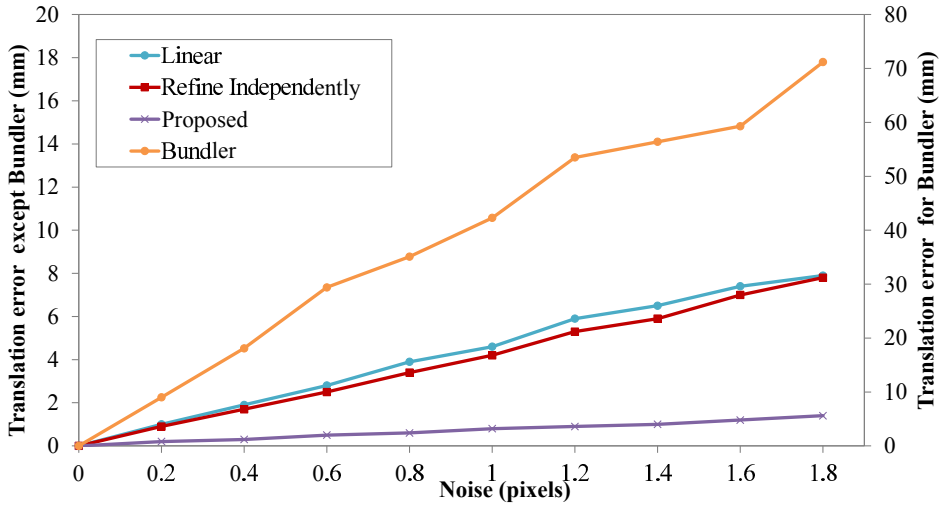


(a) Error in focal length

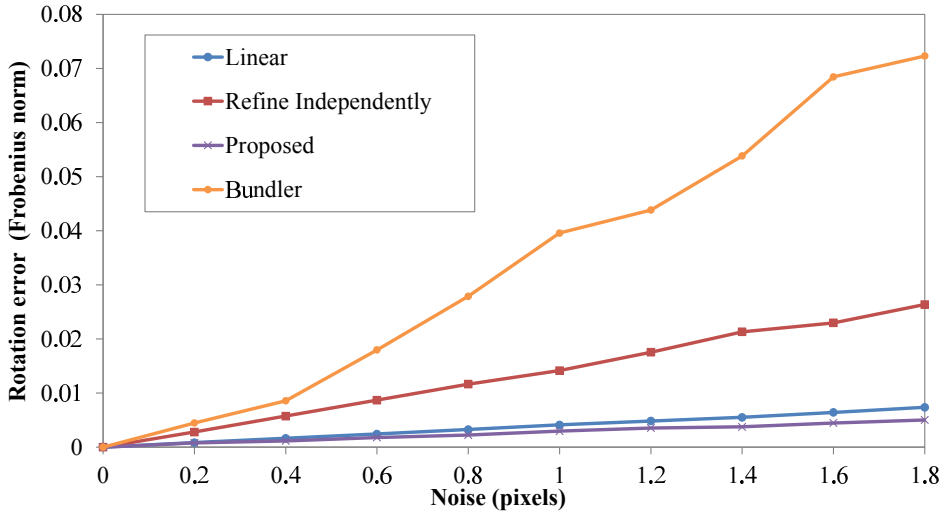


(b) Error in principal point

Fig. 3.10 Error vs. noise level of the image point.



(a) Error in viewpoint translation



(b) Error in viewpoint rotation

Fig. 3.11 Relative extrinsic parameter error vs. noise level of the image point.

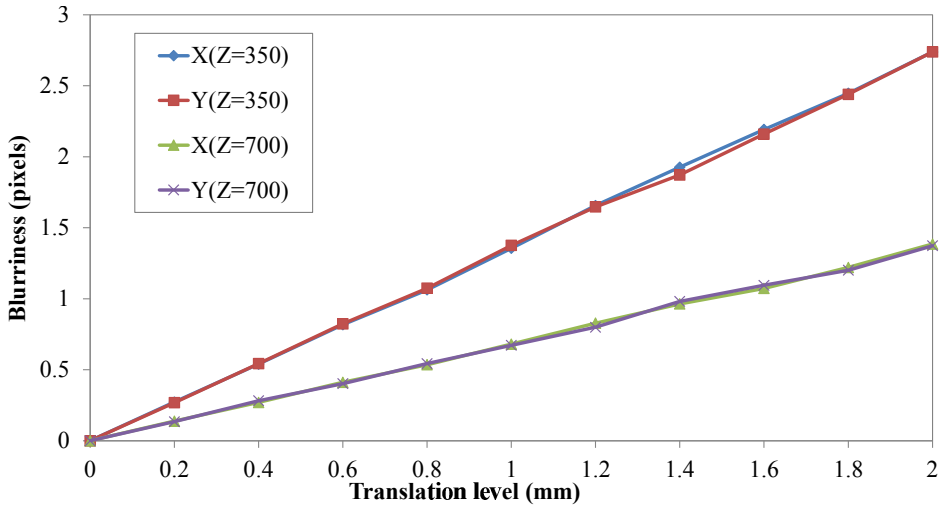
Table 3.2 Estimation of lens distortion.

	Independently refined	Proposed
$\lvert d_1 - \hat{d_1}\rvert / \lvert d_1\rvert$	1.5×10^{-8}	1.4×10^{-7}
$\lvert d_2 - \hat{d_2}\rvert / \lvert d_2\rvert$	6.0×10^{-8}	2.2×10^{-7}
$\lvert d_3 - \hat{d_3}\rvert / \lvert d_3\rvert$	2.2×10^{-8}	1.4×10^{-8}
$\lvert d_4 - \hat{d_4}\rvert / \lvert d_4\rvert$	7.0×10^{-8}	6.7×10^{-8}

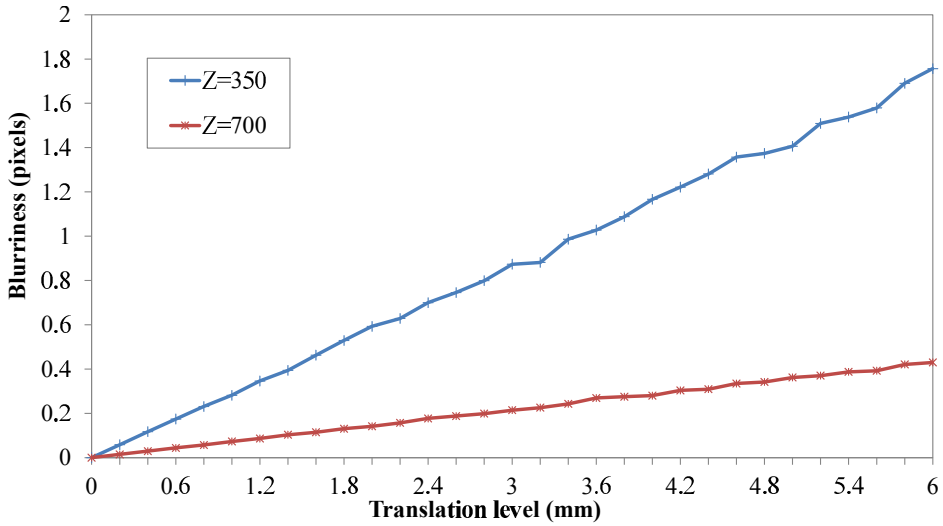
yields better results than the other methods. The errors in the results from Bundler are greater than the other methods. The reason for this is that Bundler does not incorporate the constraint of the geometric relationship between the viewpoints. In addition, it does not even maintain the same intrinsic parameters for the same viewpoint. It adjusts the intrinsic and extrinsic parameters to minimize the re-projection error. When the estimated focal length deviates from the ground truth, the translation vector and rotation matrix also deviate from the ground truth, and therefore the translation and rotation errors increase rapidly.

In order to evaluate the calibration results of lens distortion, the reference points with distortion are generated. The ground truth of the distortion coefficients are $d_1 = -0.4$, $d_2 = 0.2$, $d_3 = 0.01$, $d_4 = 0.005$, and other settings are same as previous. The estimated results are shown in Table 3.2. $\hat{d_i}$ is the estimated distortion coefficients. The linear method does not deal with lens distortion and Bundler only deal with the undistorted images, so the author compared the results estimated from independent refinement and the proposed method. We can see that the both two methods have good accuracy for the calculated coefficients.

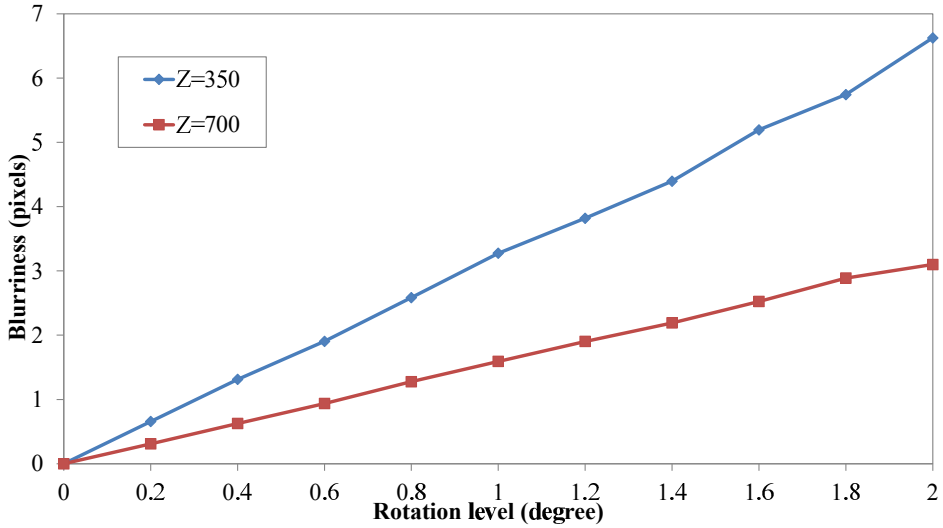
In addition, the author evaluated that how important the accuracy of the extrinsic parameters brings on the actual applications. This is performed by calculating the blurriness of the refocused light field when the parameters are deviated from the ground truth. The blurriness of refocused light field is defined as the average of the standard deviation for all the reference points in the refocused image. Here the author still simulated 70 corner points in the checkerboard pattern as the reference points as previously did. Fig. 3.12 shows the evaluated blurriness when the translation and rotation parameters deviated from the ground truth. The author perform the digital refocusing in two different depth, $Z = 350mm$ and $Z = 700mm$. The simulated deviation is with 0 mean and a certain standard deviation for 25 viewpoints. We can see that nearer focused plane is more sensitive to the deviation of the parameters. And the deviation in Z axis can be negligible when the focus plane is relatively faraway.



(a) Deviation in X and Y axis



(b) Deviation in Z axis



(c) Deviation in rotation angle

Fig. 3.12 Blurriness vs. parameter deviation.

Table 3.3 Comparison of re-projection errors (pixels).

Method	Re-projection error	Standard deviation
Linear	2.0672	0.7805
Independently refined	0.6898	0.2670
Bundler	0.5962	0.1496
Stereo pair	0.4145	0.0519
Proposed	0.3952	0.0679

3.5.2 Results with real data

The proposed algorithm was also applied to the calibration of a real light field camera. Here the author give an example of the calibration of a commercial product, the Pro Fusion25 (ViewPlus Inc., Tokyo, Japan), which has 25 VGA resolution (640×480 pixel) cameras. This camera system can simultaneously capture images from 25 viewpoints (5 horizontal×5 vertical). The central camera in this system is assigned as the first viewpoint.

The author used the light field camera to capture several checkerboard pattern images. Each checkerboard pattern image contains $7 \times 10 = 70$ corner points. A closed form calibration, refined each viewpoint independently, estimated the parameters by Bundler, normal stereo pair calibration and the proposed optimization algorithm with the rigid constraint are performed respectively for comparison. For the stereo pair calibration method, the central viewpoint was used as the left (reference) camera, and one of other viewpoints was used as the right camera, so there are totally 24 pairs. As shown in Table 3.3, the author compared the re-projection error of the proposed method with those of the other four methods. As shown in the table, we can see that the re-projection error and standard deviation of all 25 viewpoints calculated using the proposed method is smaller compared with the values for the other methods.

When the initial parameters calculated by the closed form procedure are used in the model, the re-projection error is greater than 2 pixels. When the parameters of each viewpoint are refined independently, and calculate the median values for the translation and rotation of all captured frames, the total re-projection error is smaller than 1 pixel. However, the re-projection error for each viewpoint varies, with several viewpoints still having re-projection errors greater than 1 pixel. The re-projection error from Bundler seems better, however, the physical parameters deviate from the ground truth in order to over-fit the noise as previously discussed. The re-projection error of stereo pair method is calculated from the average re-projection error of all 24 pairs. In this method, the parameters of the central viewpoint is

Table 3.4 Comparison of computational cost.

Method	10 frames (17500 points)	100 frames (175000 points)
Linear	1.4 s	14 s
Independently refined	33 s	35255 s ≈ 9.79 hours
Bundler	684 s	279596 s ≈ 3.24 days
Stereo pair	22s	32498 s ≈ 9.03 hours
Proposed	277 s	5448 s ≈ 1.51 hours

calibrated 24 times, and the intrinsic parameters for the central viewpoint are different for every pair, because this method tries to minimize the re-projection error for each pair and also cause the over-fitting. Finally, when all parameters are refined with the rigid constraint, a total re-projection error smaller than 0.4 pixels can be obtained. The author also confirmed that the re-projection error for each viewpoint is within 1 pixel, i.e., the calibrated image from every viewpoint has no disparity at infinity.

3.5.3 Computational cost

The experiments were carried out on a PC with an Intel Xeon E5620 CPU (2.40 GHz 4-core) and 24 GB memory. The author evaluated the computational cost using two datasets with different numbers of frames: one with 10 frames and the other with 100 frames. Each frame has 70 points and both datasets have 25 viewpoints. The proposed method is implemented in C++, as is Bundler. Table 3.4 shows a comparison of the computational cost. When estimating the parameters for 10 frames, it takes about 2.5× longer for Bundler to reach convergence. However, with 100 frames, the computational cost increases dramatically. Refinement using Bundler takes more than three days and stereo pair calibration takes more than 9 hours, whereas the proposed method takes just 1.51 hours.

3.5.4 Digital refocusing

One of the most popular applications of light field cameras is image refocusing after the light field image has been captured. However, if the position and orientation of each viewpoint

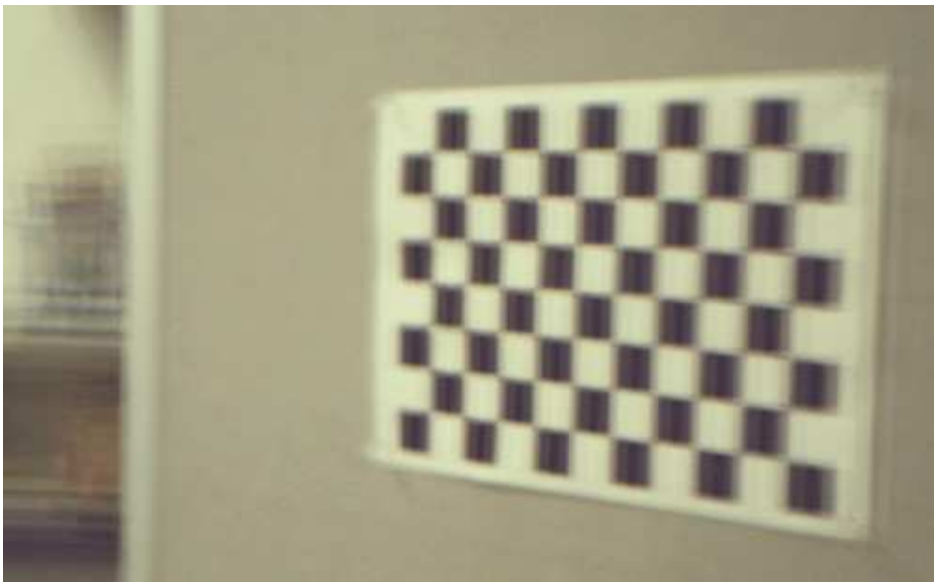
is unknown, we may never obtain a refocused image. Large lens distortion also affects the results for the refocused image.

The author attempted to generate an image focusing on the checkerboard (as shown in Fig. 3.13). The upper image was generated from the rough calibrated light field images. The author used the parameters provided by the official specification and performed refocusing using the captured images. The author selected the sharpest image visually, but this was still blurred (Fig. 3.13a). The bottom image was generated from the calibrated and rectified light field images by the proposed method. The author used the positions and orientations obtained from the calibration process. Then, the author performed a warping transform, which re-projects the light field image onto the target plane, and then simply sums and averages the images from all viewpoints. We can see that the resulting image is clearly focused on the checkerboard (Fig. 3.13b).

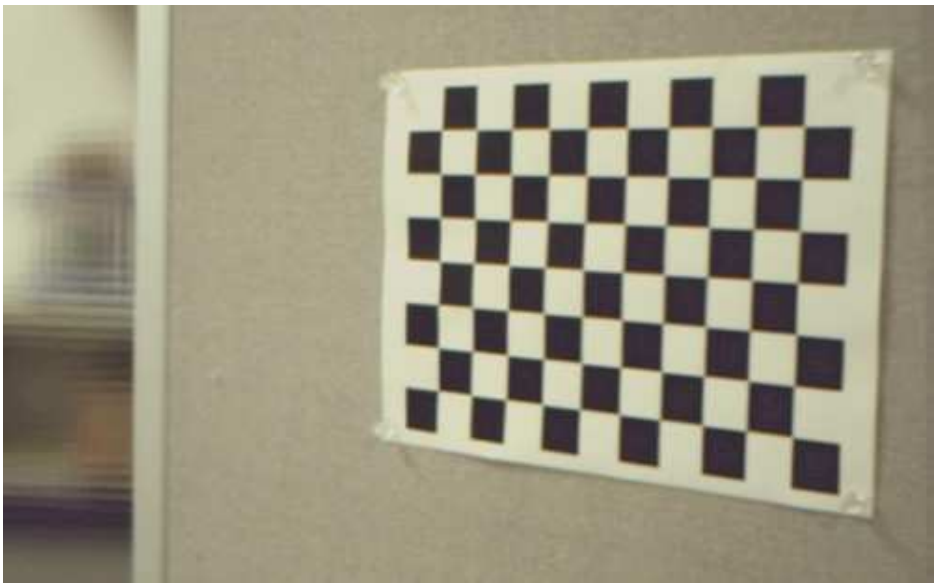
The author also captured some light field images for a real scene using the commercial product, and then rectified the raw light field images using the proposed algorithm. Thereafter, images focusing on different objects (as shown in the bottom row of Fig. 3.14) can be rendered. The images from left to right were refocused on the near, middle, and distant objects, respectively. It is obvious that the images generated with rough calibrated light field images (as shown in the top row of Fig. 3.14) do not focus on the desired object, whereas the images in the bottom row are refocused on the desired object.

3.6 Summary

In this chapter, a model of a camera array is derived for light field acquisition, which includes a rigid constraint between viewpoints. A calibration algorithm with global optimization based on the rigid-constrained model is proposed for the light field camera. The proposed algorithm extends Zhang's well-known method, and consists of two steps: closed form initialization and non-linear refinement. It is rather faster than conventional calibration methods that do not have the rigid-constraint. The proposed method is not restricted with respect to the number of viewpoints or alignment of cameras. Simulation experiments show that the proposed algorithm yields better results than the linear solution, the method where each viewpoint is refined independently, and Bundler. The proposed method was also applied to the calibration of a commercial light field camera, and the results show that all intrinsic and extrinsic parameters were optimized with a total re-projection error smaller than 0.4 pixels. A digital refocusing experiment was performed on a captured light field image as well. The calibrated light field image was effectively refocused on the required target, whereas this was not possible for the rough calibrated light field image.



(a) Refocused image with rough calibrated light field



(b) Refocused image with calibrated light field by proposed method

Fig. 3.13 Refocused image of the calibration chart.

(a) Near object cannot be refocused (b) Middle object cannot be refocused (c) Distant object cannot be refocused

(d) Well refocused on near object (e) Well refocused on middle object (f) Well refocused on distant object

Fig. 3.14 Refocused images of a real scene using rough calibrated(top) and well calibrated(bottom) light fields.

Chapter 4

Feature Descriptors from Light Field

When the light field is captured and rectified by the calibrated parameters, special features can be extracted from the light field. These features are mainly from the geometrical information of the light field image, and they can describe the characteristics of transparent objects and the occlusion boundary in the central view.

4.1 Light Field Distortion Feature

A transparent object can deform the background scene by refraction. Because refraction by objects is affected by shape and refractive index, different objects produce different images of the same scene as shown in Figure 4.1. The author utilized the background distortion caused by refraction to categorize different types of transparent objects in the proposed method. In fact, the author modeled the background distortion to the appearance difference from different perspectives (Figure 4.2). Theocratically, the modeled distortion itself is independent of background texture. Although the background determines image appearance, the distortion for corresponding points from different viewpoints is maintained. Therefore, the proposal is to model the refraction of transparent objects as a distortion of multiple viewpoints captured by the light field camera. In this section, the author will define the LFD feature which is useful in transparent object categorization. The light field is a function that describes the amount of light emitting in every direction from every point in a scene. Conventional cameras integrate light field along the angular domain, and lose the information about the light distribution entering from the world [12]. In contrast, a light field camera obtains a 4D light field image which includes both the angular and positional information of the light rays. There are various representations of the light field. The 4D-ray representation of the light field $L(s,t,u,v)$ adopted here is same as that defined in the previous chapter, which is determined by the intersection of the viewpoint plane (s,t) and the slant of ray

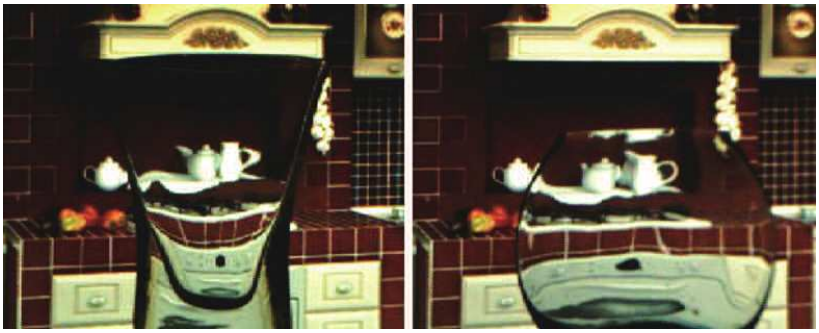


Fig. 4.1 Background distortion from different objects.

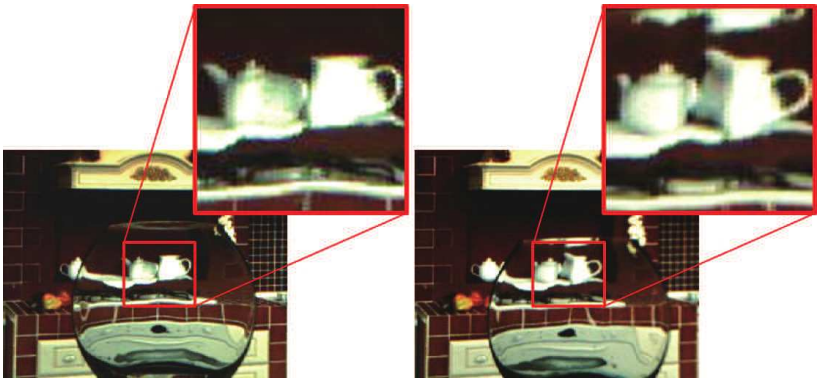
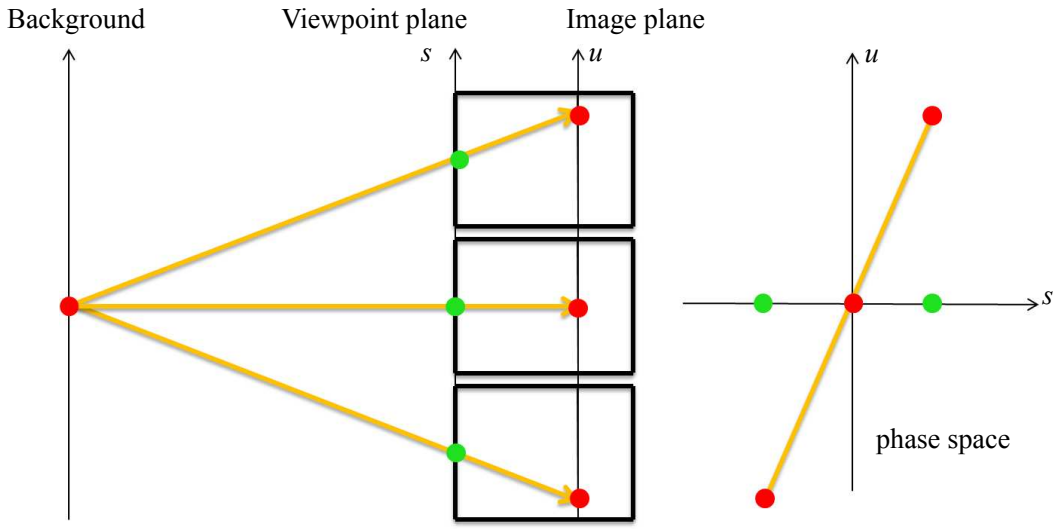
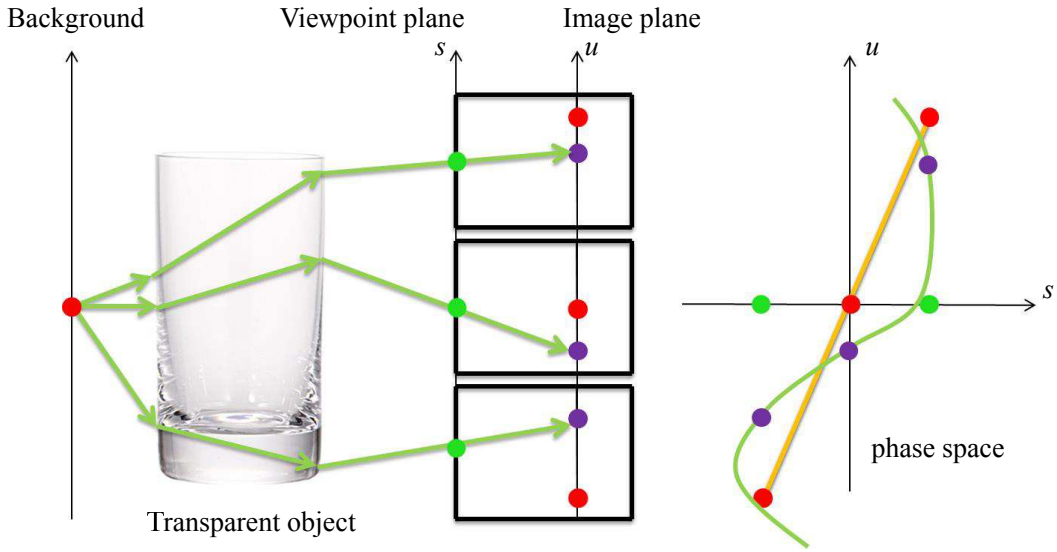


Fig. 4.2 Background distortion from changing viewpoints.

(u, v) (see Figure 3.5). Figure 4.3a illustrates the functioning of a camera array and shows the relation between light field and phase space representations. Figure 4.3a shows only a 2D slice of the light field and phase space for ease in understanding.



(a) Undistorted light field.



(b) Distorted light field.

Fig. 4.3 Light field propagation.

Figure 4.3a depicts a scene where there is no object between background and camera; i.e., light propagates in free-space with no refraction, reflection, scattering, or absorption. As illustrated, if rays emitted from a point in the background are straight, the observed light field has constant disparities over the images for the different viewpoints. The rays from the same point are distributed on a line in the *su*-phase space (Figure 4.3a), and the slope

of the line depends upon the distance between camera and background. In fact, these rays are distributed on a hyperplane in *stuv*-space because the actual light field and phase space is in a 4D space. In contrast, if a transparent object intervenes between background and camera, the ray distribution deviates from the line or the hyperplane (Figure 4.3b). This LFD is caused by refraction occurring within the transparent object, which is characterized by the material (refractive index) and the shape. This is called LFD feature in that it is to be used as a feature in transparent object categorization.

Here, the author denotes an arbitrary point in the image taken from the center viewpoints $(0,0)$ as $p(0,0,u,v)$ and the corresponding point in the image taken from another viewpoint (s,t) as $p'(s,t,u',v')$. To make the LFD feature independent of the position of the point (u,v) and (u',v'), relative differences are used. and the relative differences are defined by the following expression,

$$\begin{cases} \Delta u = u' - u \\ \Delta v = v' - v \end{cases} \tag{4.1}$$

In the experiments, an optical flow algorithm is used to obtain the correspondences between the central viewpoint $view(0,0)$ and viewpoints $view(s,t)$. The author uses large displacement optical flow (LDOF) [73, 74]in this work. Finally, the LFD feature is defined as the set of relative differences,

$$\mathbf{LFD}(u,v) = \{(s,t,\Delta u,\Delta v) | (s,t) \neq (0,0)\}, \tag{4.2}$$

where $2m+1$ and $2n+1$ are the numbers of viewpoints.

4.2 Light Field Linearity

The LFD feature has an important property called light field linearity (LF-linearity). This property of the LFD feature can be used to describe the likelihood of a pixel being the Lambertian background. It is useful in both transparent categorization and segmentation applications.

As described in Fig. 4.3, the disparities in a transparent object include the refraction effect. Thus, the LFD features coming from the transparent object are more distorted than features from the background, and these features deviate from the hyperplane given by the Lambertian reflection in the phase space. The hyperplane in the *stuv*-space containing point $p(0,0,u,v)$ can be described as:

$$n_1 s + n_2 t + n_3 \Delta u + n_4 \Delta v = 0, \tag{4.3}$$

where $(s,t,\Delta u,\Delta v)$ is as before, i.e., the viewpoint coordinates and the difference between the corresponding image points. As the positions of the viewpoints should be calibrated in advance, we can easily obtain their coordinates. (n_1,n_2,n_3,n_4) is the unit normal vector $\vec{\mathbf{n}}$ of the hyperplane. This vector is estimated by fitting $(s,t,\Delta u,\Delta v)$ from all M viewpoints:

$$\underbrace{\begin{bmatrix} (s,\ t,\ \Delta u,\ \Delta v)_1 \\ (s,\ t,\ \Delta u,\ \Delta v)_2 \\ \cdots \\ (s,\ t,\ \Delta u,\ \Delta v)_M \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix}}_{\vec{\mathbf{n}}} = \mathbf{0}. \tag{4.4}$$

Then singular value decomposition method is used to calculate $\mathbf{A}^\top\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^\top$, and the linear least-squares solution to $\vec{\mathbf{n}}$ is the column of $\mathbf{U}$ associated with the smallest eigenvalue in $\mathbf{D}$, where the smallest eigenvalue is the least-squares error $E(u,v)$. Smaller errors imply better linearity, and larger errors indicate that the feature deviates strongly from the hyperplane. Because this error $E(u,v)$ describes the linearity of the LFD feature, This is called LF-linearity in this work. This important property is used to define the regional term in the energy function.

4.3 Occlusion Detector

The background can be occluded by foreground objects in different viewpoints. This is an important cue for determining the boundaries between the foreground and background. The occlusion boundary is often detected by comparing the appearance of points over time as the camera or object moves. In a light-field image, the occlusion points can be detected by checking the consistency of the forward and backward matching between a pair of viewpoints, as illustrated in Fig. 4.4.

Let us denote an arbitrary point in the image captured by the central viewpoint $view(0,0)$ as $p(0,0,u,v)$, and the corresponding point in the image captured by another viewpoint $view(s,t)$ as $p'(s,t,u',v')$. Here, (s,t) are the coordinates of the viewpoint $view(s,t)$, and (u,v) are the coordinates of the point in the image plane (as shown in Fig. 4.3). The system also attempts to find the point in the central viewpoint $view(0,0)$ that corresponds to $p'(s,t,u',v')$, which is denoted as $p''(0,0,u'',v'')$.

The consistency is independent of the intensity at each point, so the geometric error of the forward and backward matching can be simply calculated as:

$$e(s,t,u,v) = dist(p(0,0,u,v), p''(0,0,u'',v'')), \tag{4.5}$$

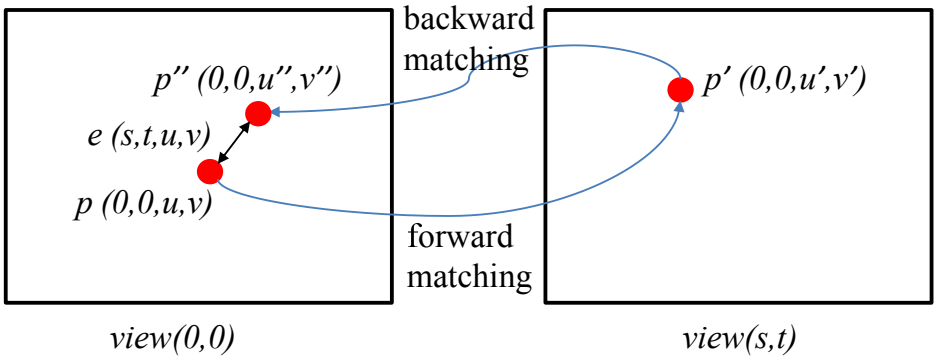


Fig. 4.4 Checking the consistency of the forward and backward matching between a pair of viewpoints.

where $dist(p,p'')$ is the Euclidean distance between p and p''.

In the non-occlusion case, points $p(0,0,u,v)$ and $p''(0,0,u'',v'')$ should be very close, which means the error $e(s,t,u,v)$ will be very small. If this consistency requirement is not satisfied, the point is either occluded in the corresponding viewpoint, or the optical flow has been incorrectly estimated. The small values are mainly from noise, and the large error values do not have much physical meaning. Hence, the LF-consistency $c(s,t,u,v)$ is defined by binarizing the error $e(s,t,u,v)$.

$$c(s,t,u,v) = \begin{cases} 0, & e(s,t,u,v) < \tau \\ 1, & e(s,t,u,v) \geq \tau \end{cases} . \tag{4.6}$$

where τ is a tolerance interval that allows the noise introduced by the optical flow calculation. Zeros are assigned to consistent points and ones to inconsistent points.

The LF-consistency has different patterns when the occlusion boundary appears in different directions. Fig. 4.5 shows an example of a point that has both consistency and inconsistency in different viewpoints. Based on the observations, the author has designed a series of occlusion detectors $F(s,t,\theta)$ to detect the occlusion boundaries between foreground and background. The detectors of 5×5 case, which are used in the experiments, are shown in Fig. 4.6, and θ is the normal direction of the occlusion boundary. The size of occlusion detector is corresponding to the number of viewpoints. The non-zero values in the detector indicate a point is occluded in the corresponding viewpoint.

The LF-consistency $c(s,t,u,v)$ and occlusion detector $F(s,t,\theta)$ are used to decide the likelihood of a pixel (u,v) being the occlusion boundary in the direction θ:

$$O(u,v,\theta) = \sum_s \sum_t c(s,t,u,v) \cdot F(s,t,\theta). \tag{4.7}$$

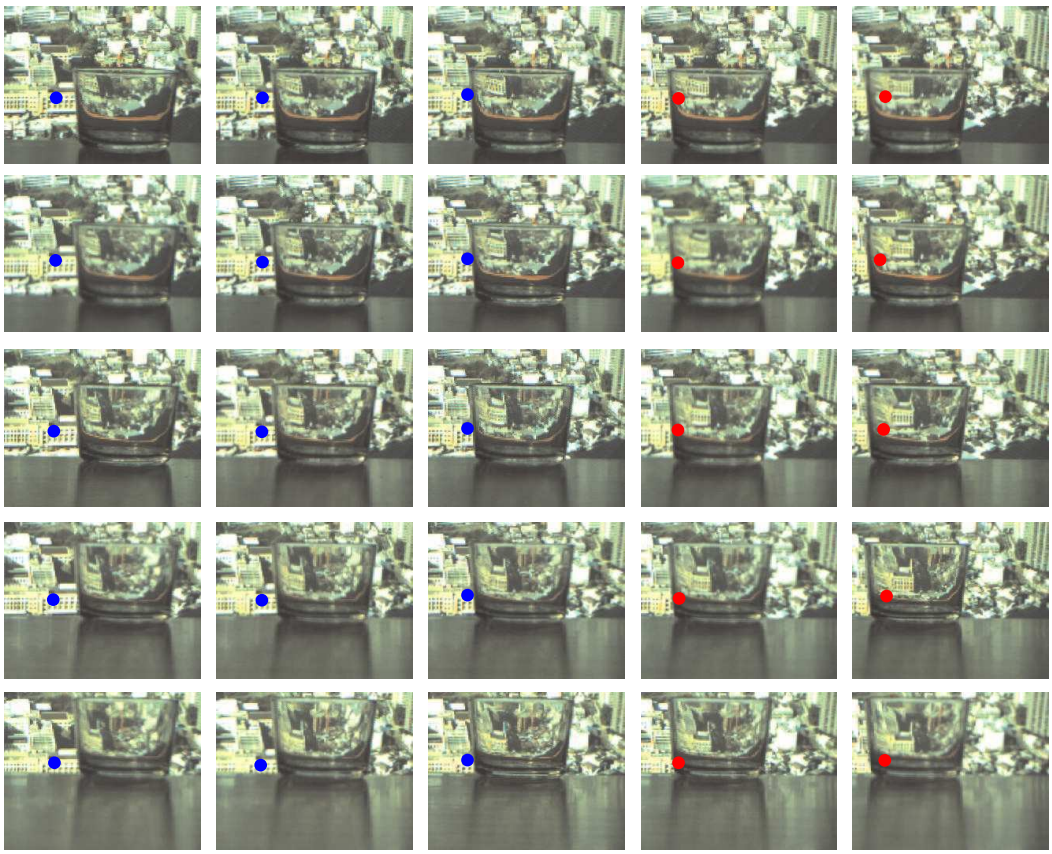


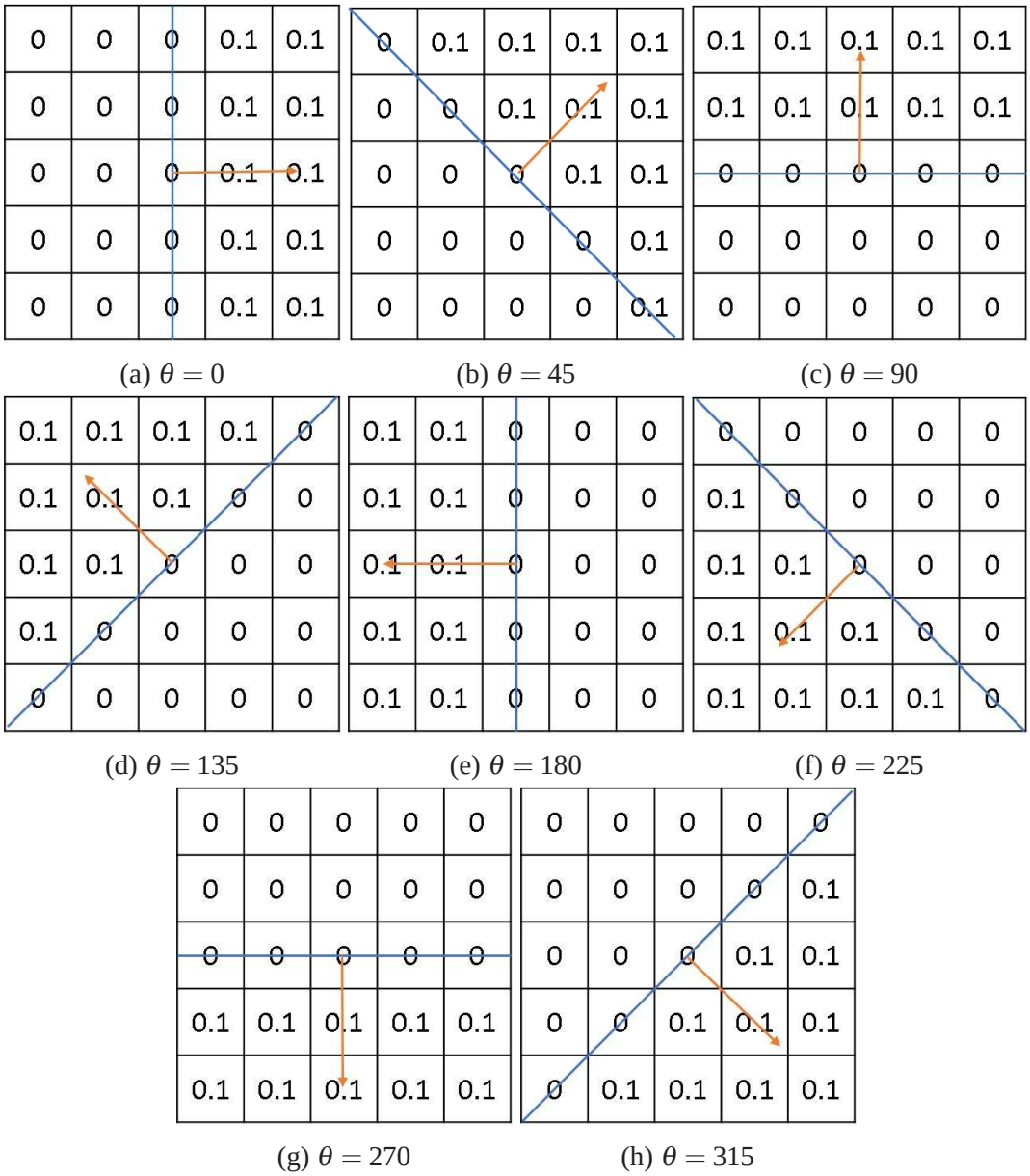
Fig. 4.5 An example of the pixel at occlusion boundary. The pixel in the center viewpoint can find the corresponding point from the viewpoints in the left 3 columns (shown in blue dots), but the corresponding point cannot be found in the right viewpoints where the point is occluded by the foreground object (shown in red dots). The blue dots have good LF-consistency, while the red dots are with poor LF-consistency.

The direction with largest response of all the detectors will be chosen as the occlusion direction:

$$\tilde{\theta}(u,v) = \arg\max_{\theta} O(u,v,\theta). \tag{4.8}$$

4.4 Summary

In this chapter, novel descriptors are proposed, which are useful for the transparent object applications. LFD feature is firstly proposed to describe the deformation of light rays in a light field image. And then LF-linearity is proposed to describe the likelihood of a light ray comes from Lambertian object or not. Finally, occlusion detector is proposed to describe the light rays are occluded in the reference viewpoint or not.



(a) $\theta = 0$ (b) $\theta = 45$ (c) $\theta = 90$

(d) $\theta = 135$ (e) $\theta = 180$ (f) $\theta = 225$

(g) $\theta = 270$ (h) $\theta = 315$

Fig. 4.6 Occlusion detectors $F(s,t,\theta)$ in eight different directions.

Chapter 5

TransCat: Transparent Object Categorization

Transparent objects are made of refractive materials, such as glass or plastics, and distort rays emanating from the background scene. The appearance of a transparent object is highly dependent on the background, from which its texture and colors are largely borrowed. Thus, it is extremely challenging to recognize the transparent object from the background. And there are few techniques for dealing with the transparent object categorization task. However, many scenarios in our everyday life deal with transparent objects. For example, when a robert is working in an environment full of transparent object, such as cleaning glasses in a kitchen, it should recognize the different types of the glasses, and decide what to do for different types of the transparent object.

In this chapter, a methods is designed to deal with the transparent object categorization task. The output of the system answers the question "which category of the transparent object is recognized?"

5.1 Background

Object categorization is the problem of training a classifier to recognize categories of objects, and then using the classifier to indicate an image belongs to which category. In recent years, the BoF-based approach has been attracting much attention in the research of image annotation and object categorization. Local features, such as SIFT, are widely used owing to their invariance to scaling, rotation and illumination [75–77]. Local features are divided into several clusters and a representative feature in each cluster is assigned by vector quantization. Objects in the same category are expected to have similar frequency within this

representative feature. This approach implicitly assumes that the majority of local features are extracted from an object's surface rather than the background. Therefore, if local features are drawn from a more dominant background than an object's surface, existing learning and recognition methods perform poorly. A transparent object yields less information about its appearance. Its actual appearance depends largely on the visible background as viewed through the object. In consequence, extracting background-independent local features from a transparent object area is difficult. Thus, these approaches find local transparent structure by applying a latent factor model before quantizing into a visual word representation [49]. Although such approaches recognize a transparent object without any knowledge of background scenes at test time, the learning step requires many training images in which the transparent object is captured under various environments.

In this work, LFD feature is used to describe the transparent object. Different objects produce different distortions, each carrying intrinsic characteristics of the transparent object, namely the refractive index of material and the shape of object, both of which influence the distortion.

5.2 Proposed method

In this section, the algorithm of the proposed transparent object categorization is described. Figure 5.1 shows the overview of the proposed algorithm. The author used a commercial light field camera, Pro Fusion25 (ViewPlus Inc.), which has 25 VGA resolution (640×480 pixel) cameras. This camera system can simultaneously capture images from 25 viewpoints (5 horizontal $\times$ 5 vertical). The author transformed the 25 captured images to a rectified light field image (s,t,u,v) as shown in Figure 5.2 by the camera array calibration method described in Chapter 3.

In the LFD feature acquisition stage, the correspondences between the image of the central view and those of the other viewpoints are obtained by optical flow algorithm. A disparity between two corresponding points $p(0,0,u,v)$ and $p'(s,t,u',v')$ can be calculated from Eq. 4.1. An LFD feature is composed of these disparities, which is represented by Eq. 4.2. Colors representation is used for indicating LFD features in this thesis. Figure 5.3-top shows the examples of the correspondences between the center of three views. The author describes the 2D disparity vectors to color representations as shown in Fig. 5.3-bottom. Each cell corresponds to a specific (s,t) coordinate; hue and saturation of each cell represent the direction $\arctan(\Delta u/\Delta v)$ and the length $\sqrt{\Delta u^2+\Delta v^2}$ of the vector, respectively. In the experiments, the author uses the large displacement optical flow (LDOF) method [73, 74] to obtain the correspondences between the the center viewpoint and the other 24

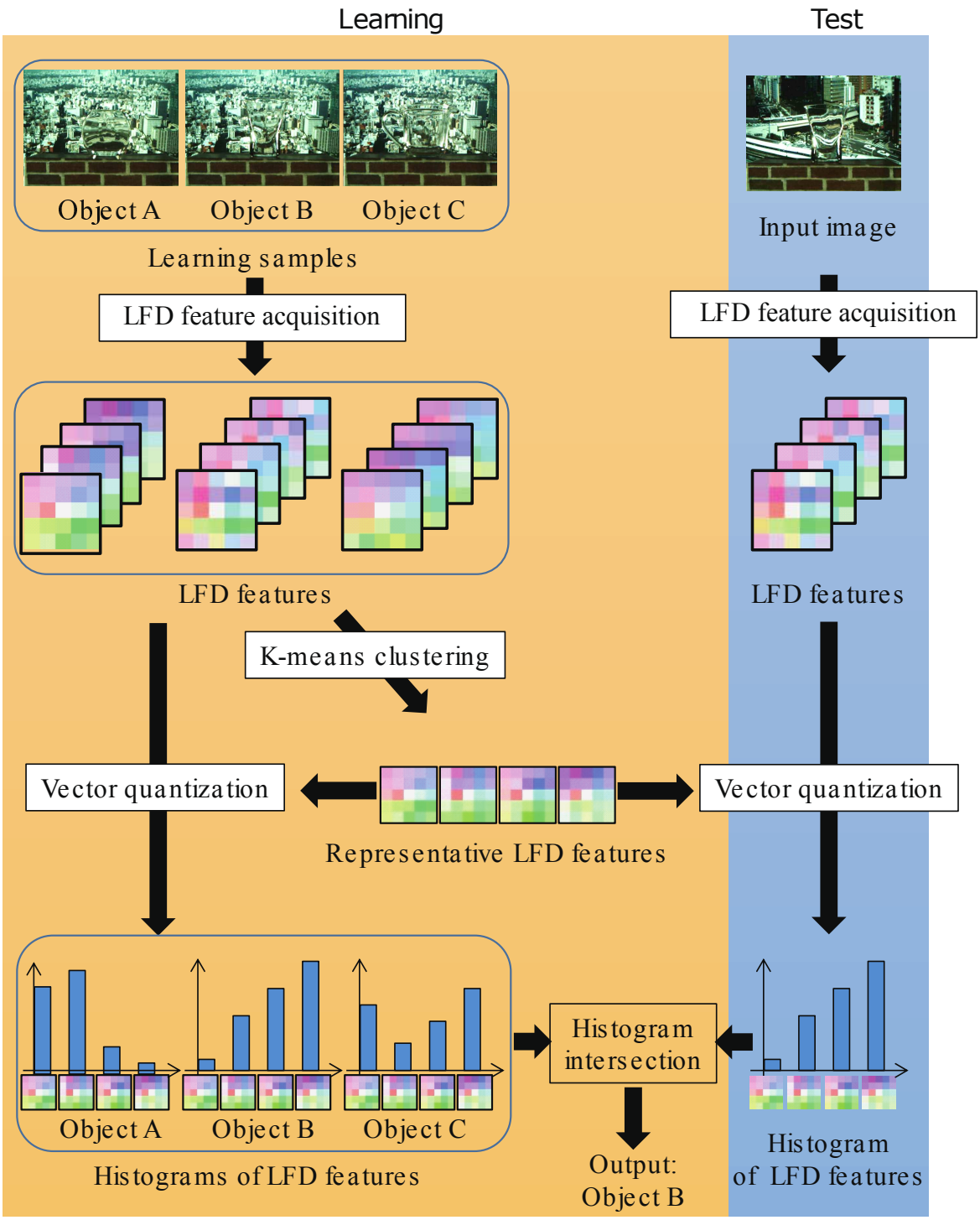


Fig. 5.1 Overview of the algorithm. The input of the proposed algorithm is the light field image as shown in Figure 5.2. The LFD features are obtained by disparities of the center view and those of the other viewpoints. LFD features are described as colored vector representation as shown in Figure 5.3. And the feature vectors are quantized to a set of representative LFD features. Then the histogram is calculated for each input light field image based on the representative LFD features. The output is done by comparing the histogram of the test image and the training data set.

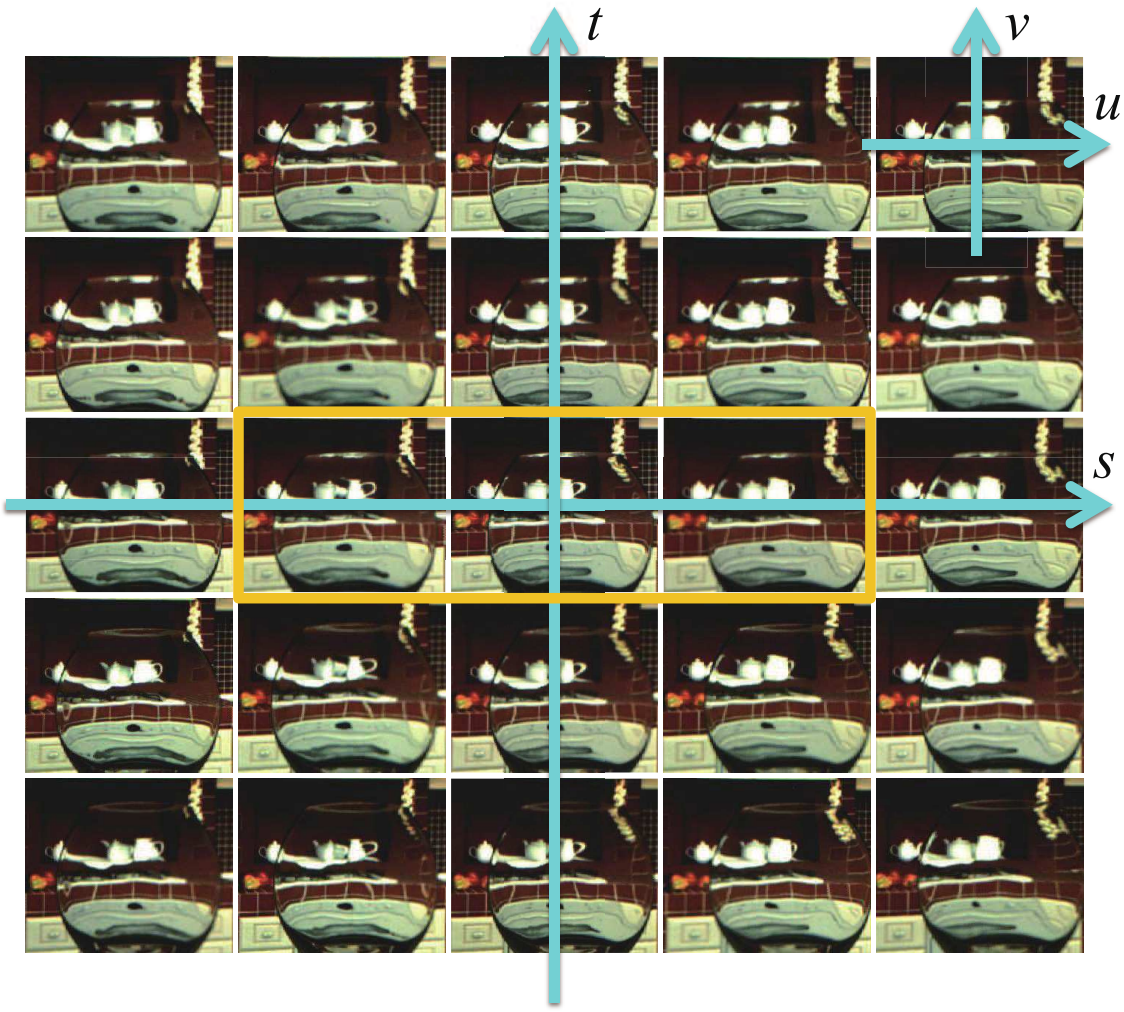


Fig. 5.2 Light field image.

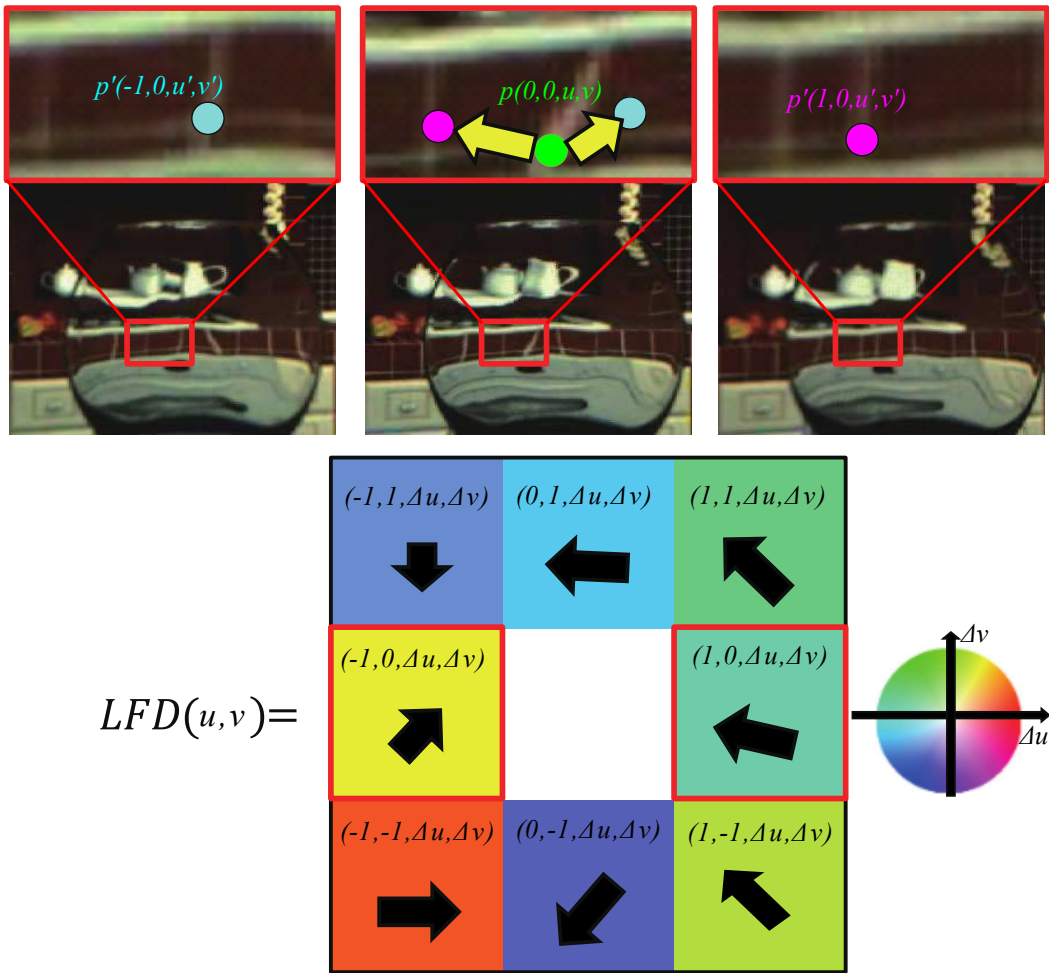


Fig. 5.3 LFD feature and corresponding points. This is an enlargement of the central images of Figure 5.2. The LFD is also an example of 3×3 case; these images are actually taken by a 25-viewpoint light field camera. Hue and saturation of the color represent the direction and length of the $(s,t,\Delta u,\Delta v)$ vectors. The bottom right shows the reference color wheel for visualisation.

viewpoints. As a result, the LFD feature is represented as a 24x2-dimensional vector for each point which describes the disparities between the center view and the other views. The LFD features are pixel-wise extracted in an image. The LFD features coming from the transparent object have larger distortion than these from background, since the disparities containing refraction effect deviate from hyperplane assumed as Lambertian reflection in the phase space as described in Figure 4.3.

The proposed method filtered out the background LFD features according to the LF-linearity. In order to make the measurement of LF-linearity independent of the viewpoint number, the deviation from the hyperplane is defined as the root mean square error,

$$dev = \sqrt{E/M}, \tag{5.1}$$

where E is defined in Sec. 4.2.

The back ground LFD feature is filtered out by a certain threshold value th. When $dev < th$, the author considers the LFD features are from the background; when $dev \geq th$, the LFD features come from the transparent object. The background LFD features are filtered out in this way. As a result, a set of N LFD features can be obtained from the single light field image. The selection of the threshold value th is described in Section 5.4.1.

Training and categorization processes are performed by a typical BoF approach. The LFD features are used as visual words. In the training phase, the LFD features are quantized by k-means clustering for obtaining visual words. The system represents the categories of transparent objects as patterns of histograms of the visual words. In the testing phase, the system extracts LFD features from input image as a same manner described above, and calculates the similarities of the distances by histogram matching for categorization. Finally, the system determines the category of the object as a minimum distance of the matching.

5.3 Experiments

The author performed some experiments in laboratory setting as well as real scenes to evaluate the performance of the proposed method.

5.3.1 Assumption

The author evaluated the proposed method by categorization of transparent objects in a laboratory setting and real environments under the following assumptions;

- There is one transparent object as a categorization target in a scene.

- The target object appears in all of the viewpoints of the LF camera.
- Relative positions and poses of the camera and target object are almost same between a training and testings.
- Background is reasonably far away from the object.
- Background scenes have sufficient textural information.

5.3.2 Experimental Setting

In the experiments, the author used 5×5 viewpoints to acquire the LFD features, and a reference position for learning the bag of LFD features. The camera position is 40 cm in front of the transparent object for both lab setting and real scenes. The background is 150 cm behind the object for lab setting, and farther than 100 cm for real scenes.

The optimal threshold value th for filtering out the LFD features come from the background and the number of clusters K for the BoF approach are determined based on the parameter optimizations experiments (Section 5.4.1). The values $th = 0.25$ and $K = 1000$ are chosen for the evaluation. And the task of proposed method is to categorize 18 various shapes of the objects (Figure 5.4) into the 18 categories under the various background textures.

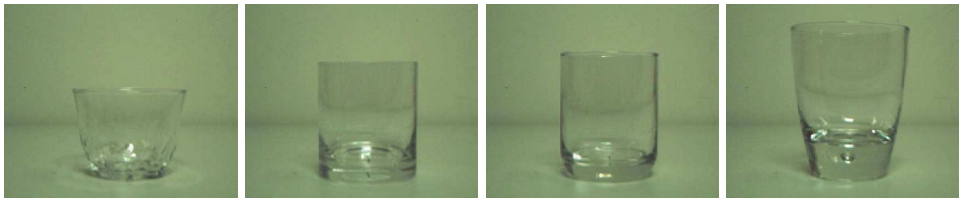
5.3.3 Categorization Results in Lab

The author performed some categorization experiments in a laboratory setting. The author used a projector and screen for a backdrop of a scene. There are 10 different scenes, as depicted in Fig 5.5.

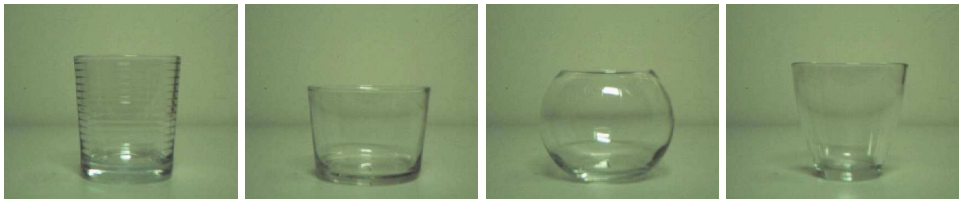
The author calculated recognition ratio among the 18 objects using leave-one-out cross validation for scenes, 1 scene is used for training and the other 9 scenes for testing, so there are totally $10 \times 9 = 90$ test images for each object. The categorization result is shown in Figure 5.6. We can see the categorization accuracy for each transparent object from 5.6a, most of the objects can be well categorized by the proposed method based on LFD feature, while some of them got lower categorization results. The confusion matrix shown in Figure 5.6b can tell us some objects are easy to be miss classified as other objects, such as the Object A is easy to be confused to Object O since their front view is similar, and Object Q and Object E are confused to each other. So these objects got lower recognition ratios than others. The proposed method achieved 84% of average categorization accuracy over

(a) Object A (b) Object B (c) Object C (d) Object D

(e) Object E (f) Object F (g) Object G (h) Object H



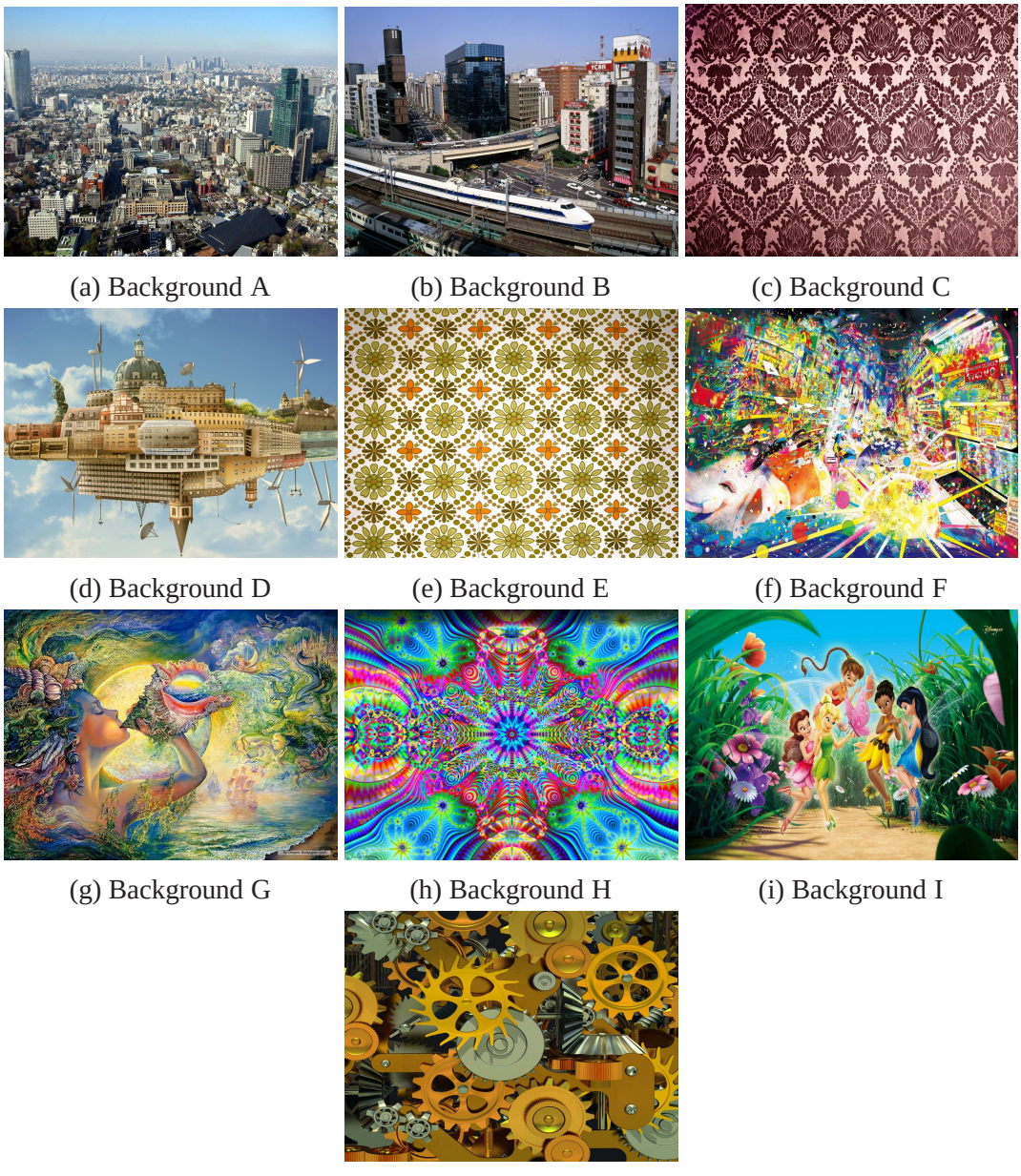
(i) Object I (j) Object J (k) Object K (l) Object L



(m) Object M (n) Object N (o) Object O (p) Object P

(q) Object Q (r) Object R

Fig. 5.4 Target transparent objects.



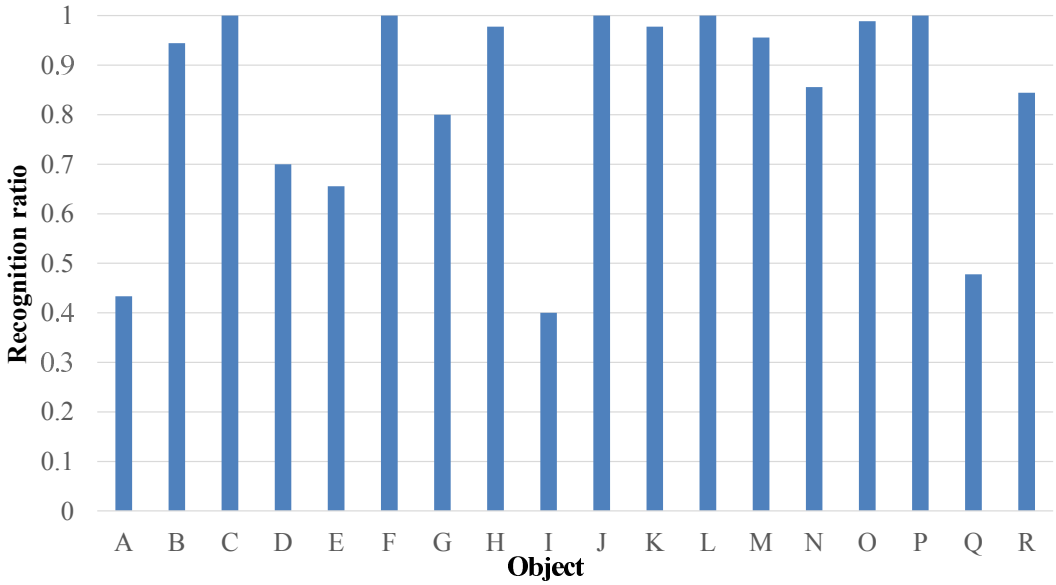
(a) Background A (b) Background B (c) Background C

(d) Background D (e) Background E (f) Background F

(g) Background G (h) Background H (i) Background I

(j) Background J

Fig. 5.5 Ten background patterns.



(a) Recognition ratio for each transparent object.

Actual Class \ Predicted Class	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
A	**0.43**	0.02	0.03	0.01	0	0	0	0.03	0	0.20	0	0.02	0	0.01	0.23	0	0	0
B	0	**0.94**	0	0	0	0	0	0.01	0	0.01	0.01	0.02	0	0	0	0	0	0
C	0	0	**1.00**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	**0.70**	0	0	0	0.11	0	0.09	0.01	0.01	0	0.01	0.04	0.02	0	0
E	0	0	0	0.01	**0.66**	0.09	0	0	0.01	0.02	0.01	0.02	0	0	0.03	0	0.14	0
F	0	0	0	0	0	**1.00**	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0.02	0	0	0	0	**0.80**	0	0	0	0.18	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	**0.98**	0	0.01	0	0.01	0	0	0	0	0	0
I	0.01	0.03	0.02	0	0.06	0.08	0	0.03	**0.40**	0.08	0	0.12	0	0	0.10	0.02	0.04	0
J	0	0	0	0	0	0	0	0	0	**1.00**	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	0.02	**0.98**	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	**1.00**	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0.03	0	**0.96**	0.01	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0.02	0.11	0.01	0	**0.86**	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0.01	0	0	0	0	**0.99**	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1.00**	0	0
Q	0	0	0	0.04	0.20	0.02	0	0	0.01	0.07	0.02	0.01	0	0	0.14	0	**0.48**	0
R	0	0	0.02	0	0	0	0	0	0	0	0.08	0	0.02	0.02	0	0.01	0	**0.84**

(b) Confusion matrix for transparent object categorization.

Fig. 5.6 Categorization result for Lab setting.

the 18 objects in front of 10 different backgrounds, although it realized transparent object categorization from a single-shot image.

Figure 5.7 shows the differences of the 4 of 1000 ($K = 1000$) frequent visual words as primal LFD features described by color representation. Figure 5.7a shows the frequent LFD features obtained by different objects with the same background. The patterns of the LFD features are different for the different objects, despite these objects look similar visually as they were placed in front of the same background. Also the LFD features come from the different regions of the objects. It means that each object was uniquely modeled by the LFDs. Figure 5.7a shows the proposed method utilizes not only the silhouette LFD features but also the inside region of the object.

In contrast, Figure 5.7b shows the LFD features from the same object placed in front of different backgrounds. It shows that these LFD patterns are the similar and coming from similar regions of the object, although the visual appearance so different among the background differences. The author confirmed that LFD feature is irrespective of the background difference, since the LFD feature does not model the intensity pattern but the geometrical distortion caused by object refraction.

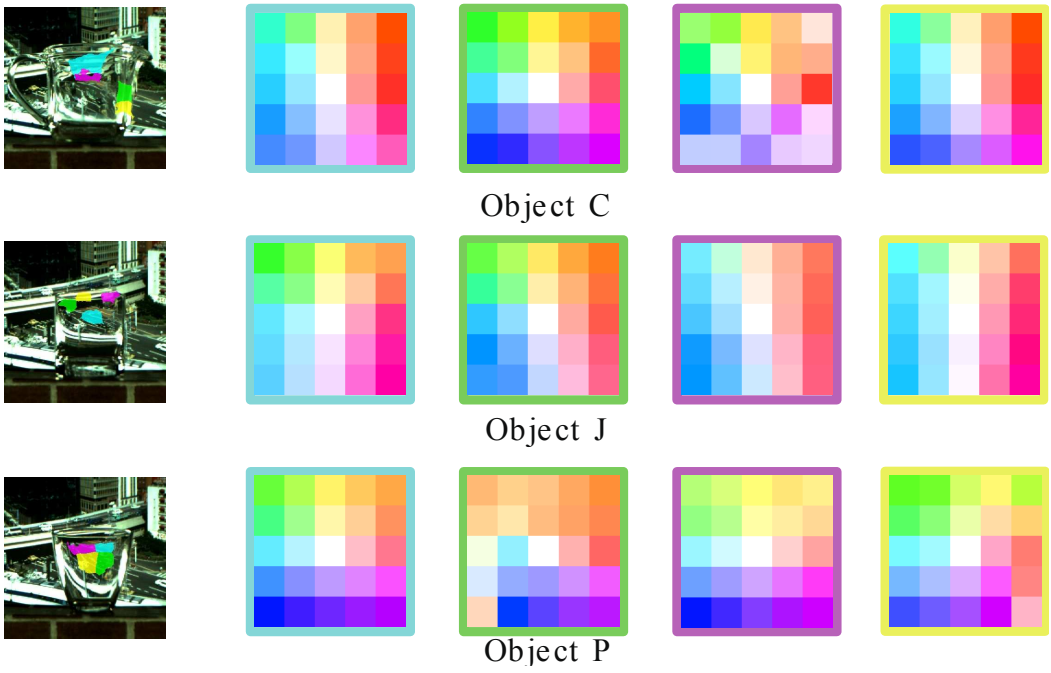
5.3.4 Categorization Results for Real Scenes

The author also performed real experiments in indoor and outdoor settings (Figure 5.8). Objects were placed about 40 cm from the camera against real backgrounds of structures at various depths, i.e., distances sufficiently far (more than 1m) from the objects. The training data set is same to the Lab setting. Table 5.1 shows the average recognition ratios for different number of transparent objects under three different scenes.

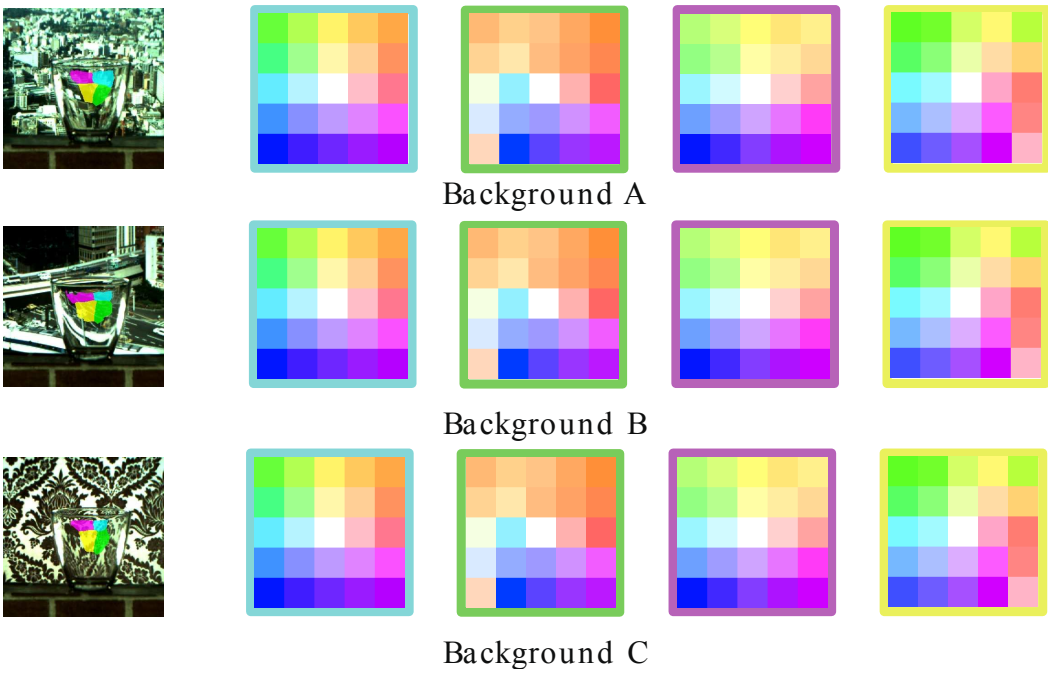
Table 5.1 Recognition ratios for real experiment.

	6 objects	10 objects	15 objects	18 objects
Proposed LFD feature	0.766	0.678	0.587	0.533
Standard SIFT	0.160	0.108	0.075	0.063

The author also used a similar categorization method in using the SIFT feature. Numbers of clusters K for the SIFT approach was also set to $K = 1000$. The Table 5.1 shows that the proposed method achieved an average 76.6% accuracy for 6 transparent objects and 53.3% for 18 transparent objects. On the other hand, the ratios of the standard SIFT are the almost chance rates. It is not discriminative to the object patterns at all, since the SIFT features mainly come from the background patterns.

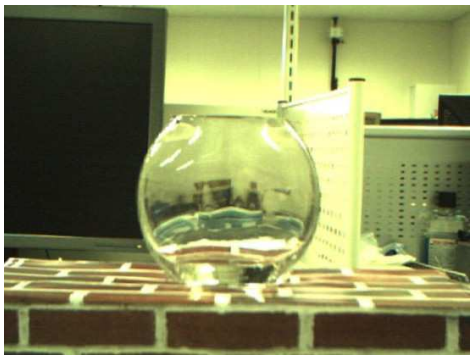


(a) Different objects with same background.



(b) Same object with different backgrounds.

Fig. 5.7 Examples of primal LFD features by color representation. Each row shows the different object or different background. The 1st column shows the objects and regions of the pixels where the primal LFD features come from. The 2-5 columns indicate the frequent LFDs describing the objects. The colors of the bounding box of the LFDs are corresponding to that of the regions in the 1st column. Hue and saturation of the LFDs represent direction and length of the $(s,t,\Delta u,\Delta v)$ vectors on 5×5 viewpoints as similar to Figure 5.3.


(a) Indoor.


(b) Outdoor.

Fig. 5.8 Examples of the real scenes.

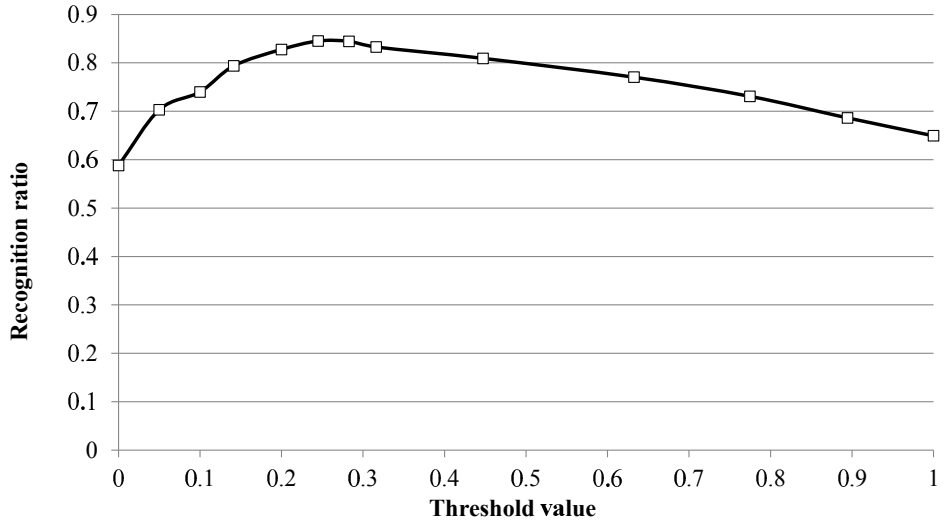
5.4 Performance Analysis

5.4.1 Parameter Optimizations

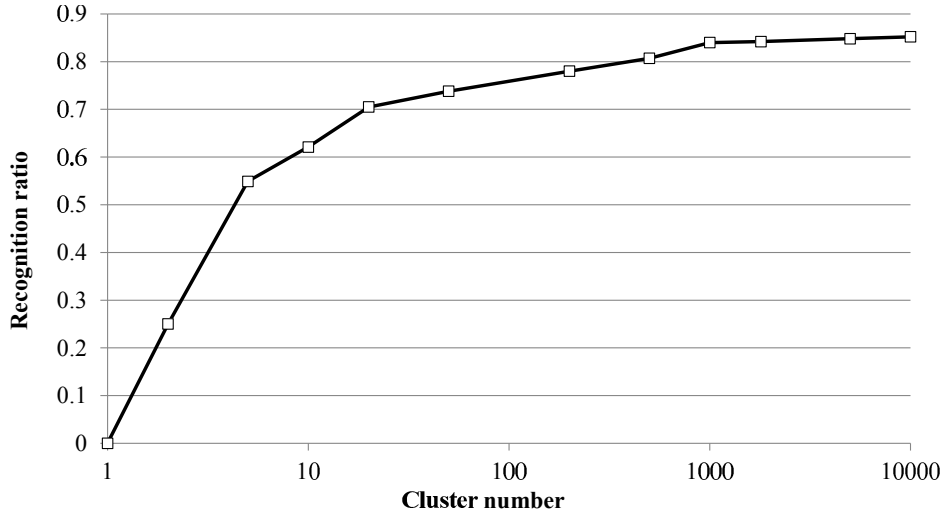
As describe in the overview of the proposed algorithm (Section 5.2), the number of LFD features is depend on the threshold value for filtering, and the number of visual words is decided by K-means clustering. In order to find the optimal parameters in the system, the author conducted a serial of experiments under the same condition described in Section 5.3.3 with different threshold values and number of clusters.

The threshold value is used to filter out the LFD features from the background. When this value is small, the features from the background cannot be filtered out. And when it is getting larger, the LFD features inside the transparent object will be filtered out as well. The cluster number $K = 1000$ is used in this experiment. Figure 5.9a shows the categorization results under different threshold values. When the threshold value is 0, the features from the background will be dominant and the categorization results is near 25% decreased from the best accuracy. The result shows that the best threshold value is $th = 0.25$, and this value is used for all of the other experiments.

The number of clusters is related to the number of visual words in the system. Regarding to the computational cost, this number is very important. The number of clusters should be carefully determined in order to get the balance of categorization speed and accuracy. Figure 5.9b shows that the categorization accuracy is getting better during the number of clusters increase. The recognition ratio almost saturated when the cluster number is over 1000, so the best choice of the cluster number for 18 different objects is $K = 1000$.



(a) Different threshold values.



(b) Different number of clusters.

Fig. 5.9 Recognition ratios for different parameters.

5.4.2 Effect of Camera and Object Settings

The author evaluated how the various camera and object settings affect the performance of the proposed method in this section. The author used the same conditions to the experiments for the laboratory setting as described in Section 5.3.2. The author investigated the effect on the categorization accuracy under various conditions, including camera positions, background distance, object rotations in three directions, additional lighting conditions, noise levels of the images and the number of viewpoints.

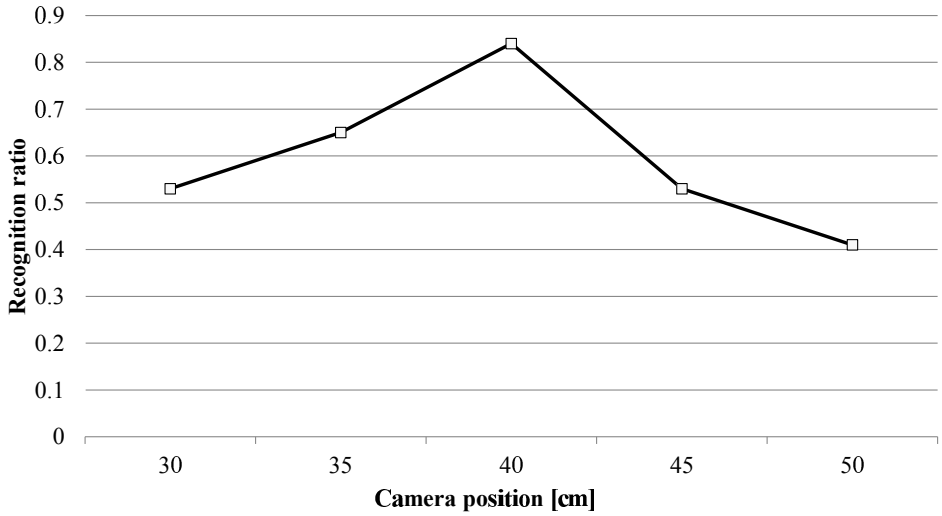


Fig. 5.10 Recognition ratios for different camera positions.

The author moved the camera over a range of ± 10 cm from the reference position 40 cm. Figure 5.10 shows that as the camera moves away from the reference position, recognition ratios worsen because the LFD features are distorted when the distance between the camera and object changes. Moving away from the reference position incurs greater error than moving closer to the object. We could consider the margins for object deviation to be about 5 cm if we accept a 10% decrease in the recognition ratio.

The author also moved the background position over the range of 50 cm to 250 cm from the object, while the reference position of the background was 150 cm. Figure 5.11 shows the recognition ratio decreased when the background displaced from the reference position. The direction of ratio decrease is opposite to that of the camera position change as shown in Figure 5.10. The ratio is not so changed when the background is away from the object, while it is steeply decreased when the background position is approaching to the object. This is because that the LFD difference caused by the depth disparity is nonlinearly occurred, near position is larger and far is small. The background position does not affect much about the recognition ratio when it is far away enough, and this method can be applied to more

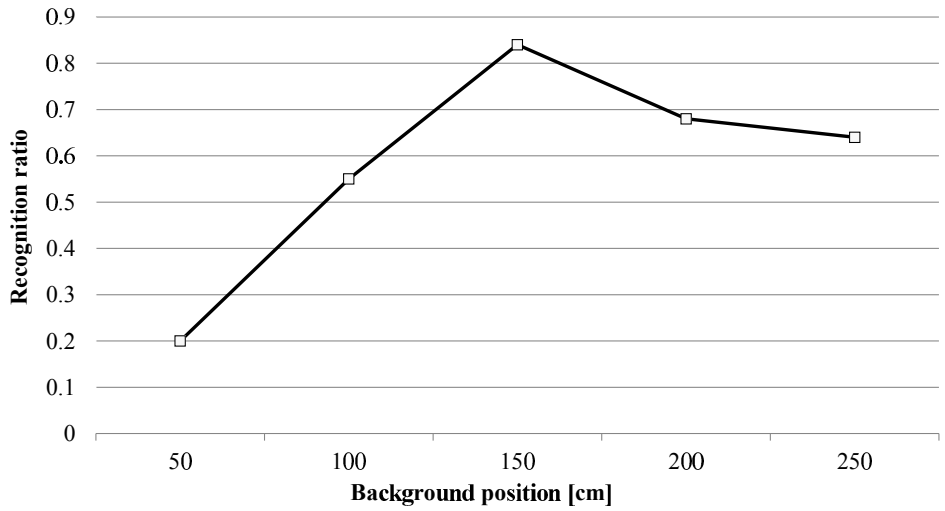


Fig. 5.11 Recognition ratios for different background positions.

realistic no planer scene background, if it can be assumed that the background objects of the scene are placed reasonably far distance, e.g. more than 100 cm.

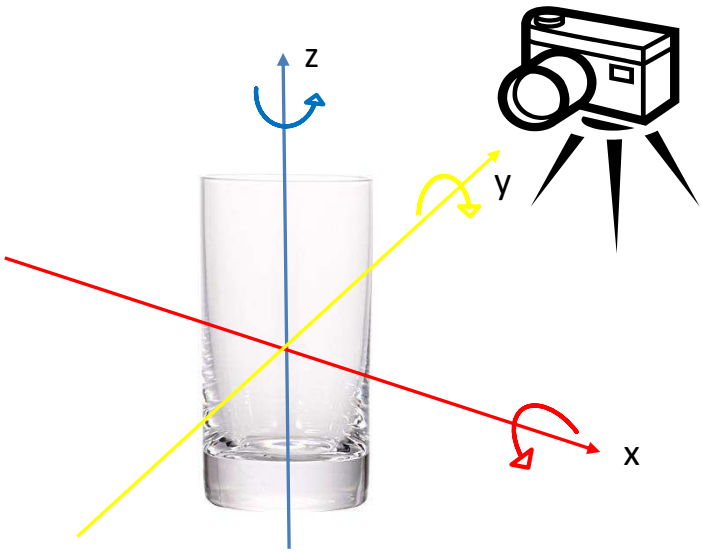
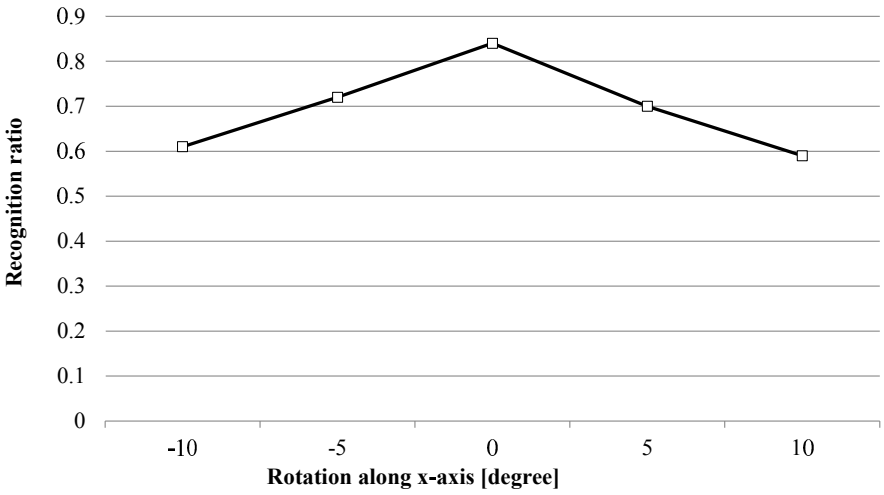
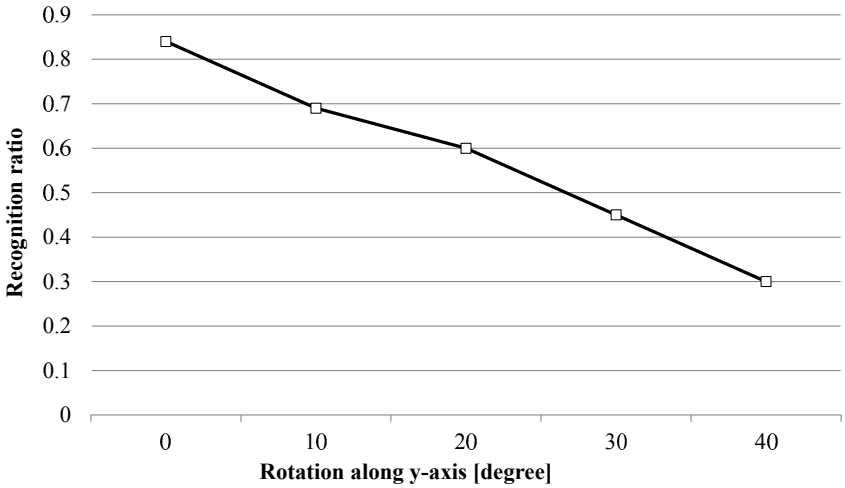


Fig. 5.12 Illustration of rotation.

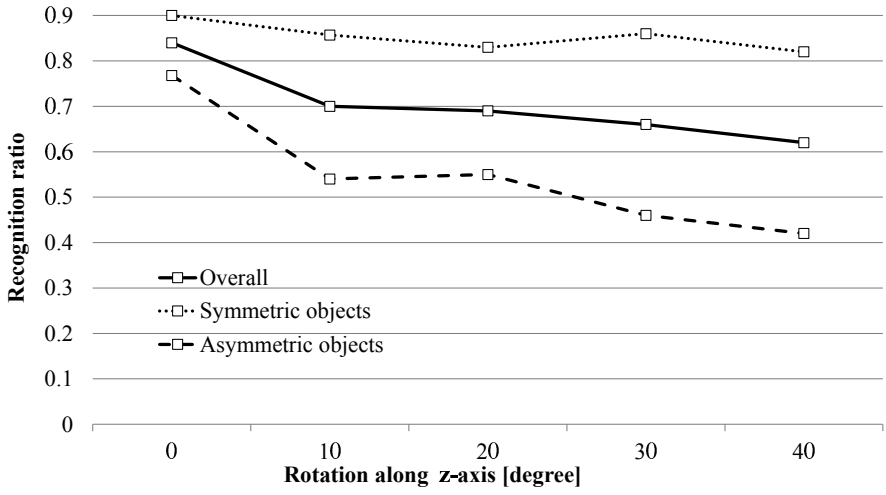
The author analyzed how the rotations affect the performance by rotating the objects or camera in three different directions as illustrated in Figure 5.12. The author rotated the objects up to 40 degrees along their central axes (z-axis). Figure 5.13c shows the results splitting to a symmetric group (Objects A-I) and an asymmetric group (Objects J-R) of the objects, as well as overall ratio. As expected, the ratio of the symmetric group is invariant to poses, since the shape and its LFD features would not be changed. The ratios of asymmetric group decrease gradually and the limitation on object pose variation is within 20 degrees



(a) Different rotation angles along x-axis.



(b) Different rotation angles along y-axis.



(c) Different rotation angles along z-axis.

Fig. 5.13 Recognition ratios for rotation along different directions.

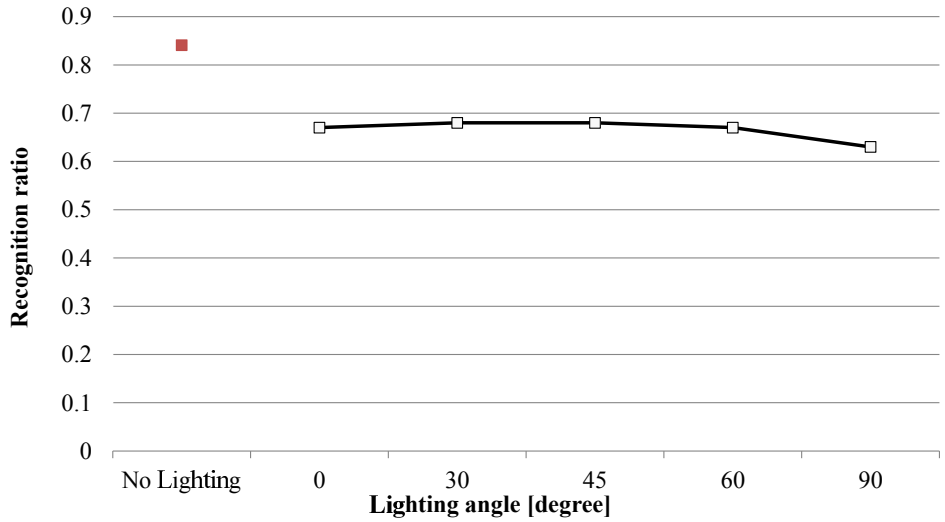


Fig. 5.14 Recognition ratios for different additional illumination angles.

if 20% degradation can be accepted in recognition ratios. Because it is difficult to rotate the object along x and y axes, the author relatively rotated the camera instead. The author rotated the camera over a range of ± 10 degrees along x-axis from the reference position, and the results are shown in Figure 5.13a. The system can get more than 70% accuracy if the rotation is within ± 5 degrees. The rotation along y-axis was up to 40 degrees, and the accuracy gradually decreases as shown in Figure 5.13b. The recognition ratio can stay over 70% if the rotation is within 10 degrees along this direction.

The author also evaluated effects of illumination change for categorization. The author placed an additional point light source to the global illumination that was used in the all of the experiments. The author changed the direction of the light source from above (0 degree) to the side (90 degrees) with respect to the target object. There were inter-reflections and specular reflections from the light source and these effects were changing as the light source was moving. Figure 5.14 shows the recognition ratios across the lighting directions. The left most label indicates the recognition ratio without the additional point light source which is same condition of learning setting. This figure shows that the internal and specular from the light source contaminated the LFD features and decreases averagely 15% of the categorization accuracy. It is not strongly related to the directions of the settings.

Some experiments are conducted to analyze how the input noise affects the performance. The author first estimated the input noise of the light field image by capturing several images with constant intensity. The average intensity was calculated as the constant intensity, and the standard deviation of all the pixels was calculated from the mean intensity. The average standard deviation of input noise is 0.032, and the author intentionally increased the noise by adding zero-mean Gaussian noise to the original light field image. The additional noise

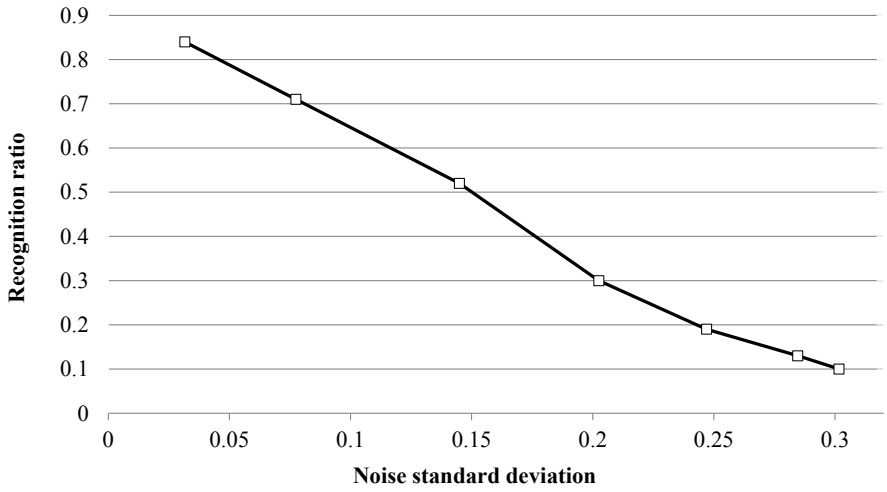
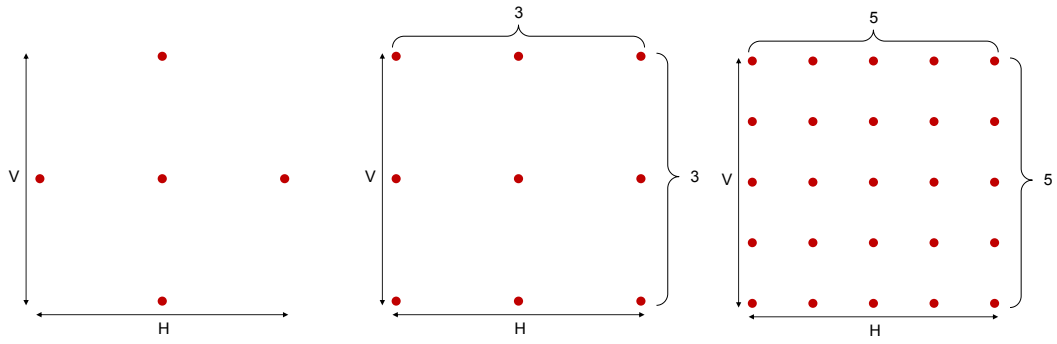
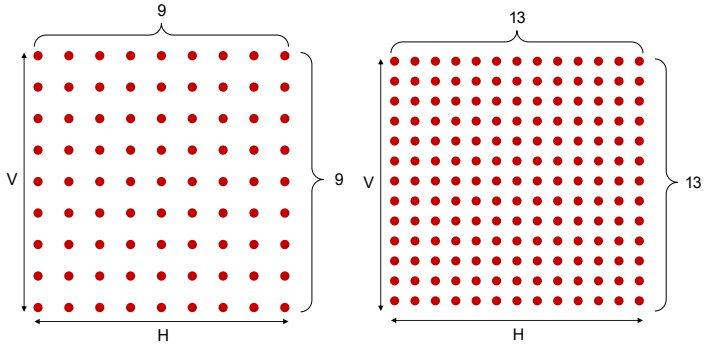


Fig. 5.15 Recognition ratios for different noise level.

standard deviation was up to 0.3. The categorization results can be seen in Figure 5.15. If we want to get the accuracy over 70%, the noise standard deviation should be controlled under 0.08.



(a) 5 viewpoints. (b) 3x3 viewpoints. (c) 5x5 viewpoints.



(d) 9x9 viewpoints. (e) 13x13 viewpoints.

Fig. 5.16 Different viewpoint settings.

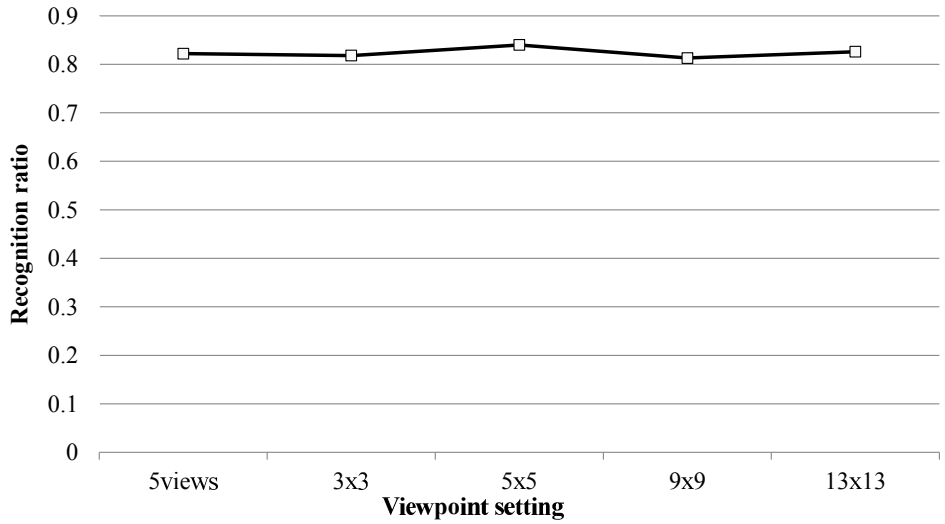


Fig. 5.17 Recognition ratios for different number of viewpoints.

In order to evaluate how the number of cameras affects the categorization results, the author decreased the viewpoints to 5 views (Figure 5.16a) and 3×3 views (Figure 5.16b). By moving the 5×5 viewpoints light field camera on a robot arm, the viewpoints can be increased to 9×9 (Figure 5.16d) and 13×13 (Figure 5.16e) as well. The author kept the same horizontal and vertical baseline for all the settings, and only increased or decreased the density of the viewpoints. Figure 5.17 shows that the recognition ratio, not as expected, is not so affected by the number of the viewpoints. This is because there are only 18 objects in the experiments, and the variation of the LFD for discriminating 18 objects is limited.

5.4.3 Effect of Texture

The background patterns used in the experiment have complex textures (see Fig. 5.5) from which correspondence detection can be easily performed. Meanwhile, LFD features cannot be appropriately extracted in certain background scenes (Figure 5.18). Because textural information is minimal, correspondences between the viewpoints are difficult to find. In Figure 5.18a, the LFD features were extracted from only the edges of the transparent object, with no LFD feature taken interior to the object. For another background (Figure 5.18b), LFD features were wrongly extracted exterior to the transparent object (see the top-left part of the figure). Therefore, the performance is affected by the accuracy in correspondence detection.

The author evaluated how many LFD feature points are needed for accurate categorization in simulation. First, to obtain ideal feature points, a dot pattern was displayed as a background to the transparent object for easy to detect the correspondence of the LFD fea-

(a) Uniform background. (b) False in uniform pattern.

Fig. 5.18 Falsely detected feature points.

tures. A total of N dots were captured; note that the number of N corresponds to the whole number of pixels. Second, a percentage d% of LFD features were randomly selected. Then, leave-one-out cross validation was performed to acquire the categorization accuracy. This procedure was repeated 100 times if d was less than 1%, otherwise, just ten times.

The recognition ratio curve is plotted in Figure 5.19. This figure shows that the proposed approach requires at least 3% of the LFD features to obtain almost 100% categorization accuracy. In terms of practical uses, extracting LFD features for at least 3% of the image size is not such a difficult problem. Therefore, the proposed LFD feature is considered effective in transparent object categorization.

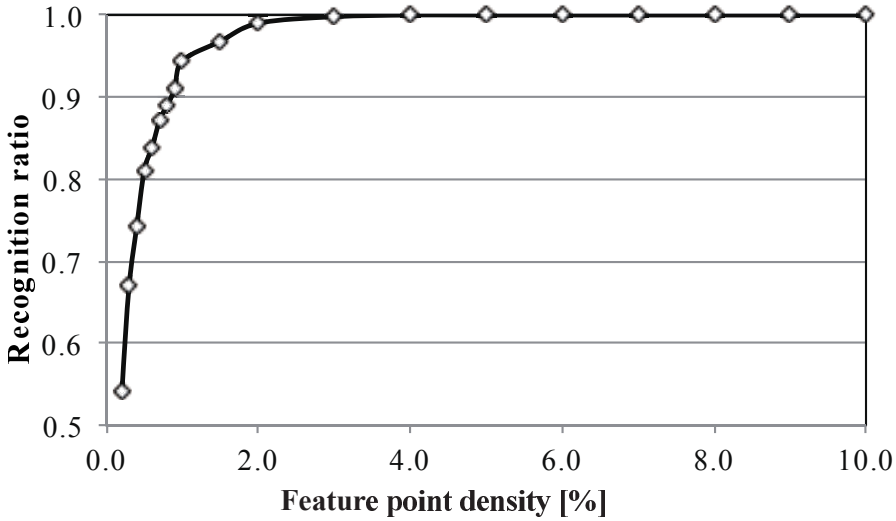


Fig. 5.19 Recognition ratio vs. density of feature points.

The author also evaluated mistracking for estimating the LFD vectors. The author used the same simulated features above and randomly selected 3% of the features. The author

added zero-mean Gaussian noise with different standard deviations to the LFD features to simulate tracking noise. The recognition ratios across different standard deviation of noise (Figure 5.20) show that ratios decrease when error levels increase. The author confirmed that less than 5.0 pixels of the error is required if the desired recognition ratio is over 70%.

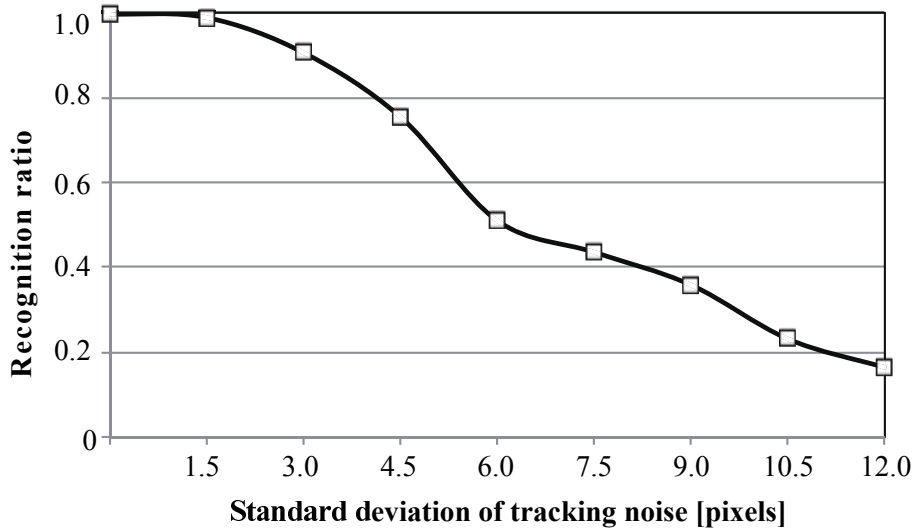


Fig. 5.20 Recognition ratio vs. tracking noise.

5.5 Summary

In this chapter, the LFD feature and LF-linearity are applied to transparent object categorization task. Several related works of object categorization are reviewed, and the problem statement is presented. The experimental results show the proposed method using light field image outperforms the conventional method using the image captured by a regular camera. Moreover, the limitations of the proposed method are analyzed, and applicable condition is clearly described.

Chapter 6

TransCut: Transparent Object Segmentation

Previous chapter proposed a method that can recognize the different categories of the transparent objects. In our daily life, we have another scenario that a machine is operating in living rooms or offices, it should recognize the fragile objects such as glasses, vases, bowls, bottles, jars, to name a few, and avoid touching them. This task is to find the transparent objects in the environment, and it is also challenging to separate the transparent object from the other Lambertian objects.

In this chapter, the author propose a method to deal with the transparent object segmentation task. The output of the system can recognize whether there is a transparent object and where it is in the image.

6.1 Background

Image segmentation is a fundamental problem in computer vision. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze [78]. For example, it is very important to separate foreground objects from the background in applications such as object detection, object recognition [79], and surveillance tasks [80]. Numerous methods have been developed to deal with the image segmentation problem, including techniques based on thresholding [81], partial differential equations [82], and graph partitioning [83, 84]. However, none of these methods are suitable for the segmentation of transparent objects from an image. It is almost impossible to achieve stable transparent object segmentation in a 2D image using these conventional image segmentation approaches. The difficulty of dealing with such objects means

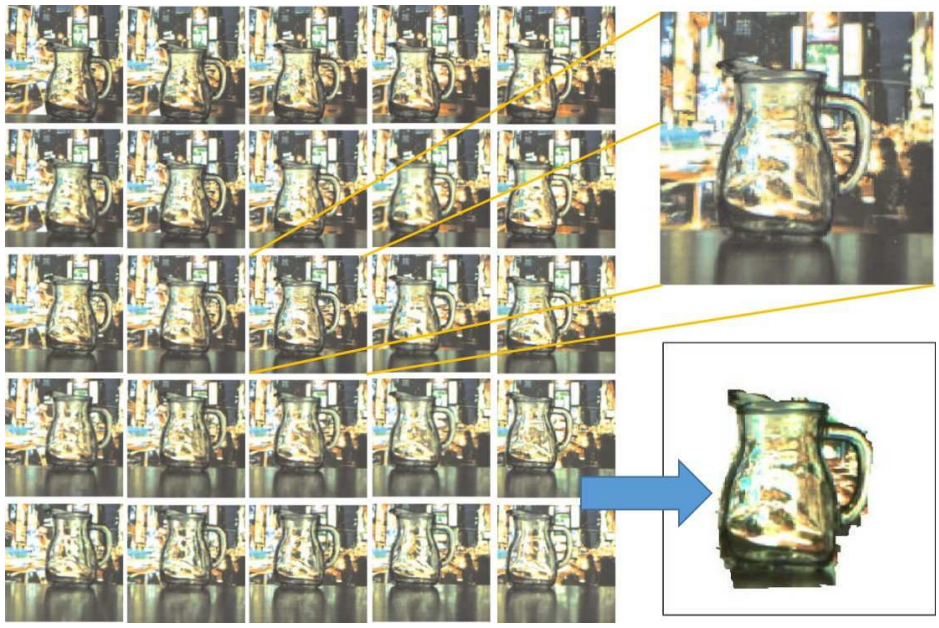


Fig. 6.1 Transparent object segmentation from a light field image. The left side shows the captured light-field image, and the right-hand side is a magnification of the central viewpoint. The output after segmentation of the transparent object is shown on the bottom-right.

that transparent object segmentation is a relatively untouched field. In this work, transparent object can be segmented from the 4D light-field images (see Fig. 6.1). The proposed method can automatically segment the transparent objects without any interaction.

An energy function is defined in the proposed method, which utilizing the LF-linearity and the output of occlusion detector. There are various strategies for optimizing energy functions. The combinatorial min-cut/max-flow graph-cut algorithm is widely used for energy functions defined on a discrete set of variables. Greig et al. [85] were the first to realize that powerful min-cut/max-flow algorithms could be used to minimize certain energy functions in computer vision applications. The energy function encodes both regional object information and the regularization of the image smoothness. The regional information usually comes from user interaction [84, 86], particularly in image editing applications. Automatic segmentation approaches that do not require user interaction have been developed in recent years. An object segmentation framework [87] has been proposed for the automatic extraction of candidate objects by solving a sequence of constrained parametric min-cut problems. Another method [88] estimates whether a pixel is inside the foreground object based on the point-in-polygon problem, whereby any ray starting from a point inside the polygon will intersect the boundary of the polygon an odd number of times. In the proposed method, the author uses occlusion to detect the boundary of a transparent object, and this occlusion boundary also allows us to determine which side is the background. The system detects

the occlusion boundary by designing a series of occlusion detectors to check the pattern of forward-backward matching consistency in all viewpoints. The forward-backward matching consistency has been used in many previous studies such as [89]. For more sophisticated occlusion detection strategies, the author refers to [90] and the references therein.

6.2 Proposed method

The goal of this work is to segment transparent objects by using LF-linearity and occlusion detector. The author formulates the segmentation task as a pixel labeling problem with two labels (transparent objects as the foreground and other objects as the background). Later part of this chapter, the author will describe each pixel as $p = (0,0,u,v)$ and some variables with subscript p indicate the variables at pixel p of the center viewpoint, since the proposed method solves the pixel labeling problem in 2D image space. Similar to other segmentation methods [84, 86], the author defines an energy function to evaluate the labeling problem:

$$E(l) = \sum_{p \in P} R_p(l_p) + \alpha \sum_{(p,q) \in N} B_{p,q} \cdot \delta(l_p, l_q), \tag{6.1}$$

where l_p is the label of an image pixel p ($l_p = 0$ denotes a background pixel, $l_p = 1$ denotes a foreground pixel), $R_p(l_p)$ is the regional term that measures the penalties for assigning l_p to p, $B_{p,q}$ is the boundary term for measuring the interaction potential between pixels p and q, N is the neighborhood set, α adjusts the balance between $R_p(l_p)$ and $B_{p,q} \cdot \delta(l_p, l_q)$, and

$$\delta(l_p, l_q) = \begin{cases} 1, & if \;\; l_p \neq l_q \\ 0, & if \;\; l_p = l_q \end{cases} \tag{6.2}$$

The segmentation task aims to determine the labeling that minimizes Eq. 6.1. The graph-cut algorithm is used to optimize the energy function.

In this section, the regional term $R(l_p)$ and the boundary term $B_{p,q}$ are defined for transparent object segmentation.

6.2.1 Regional term

It is assumed that all Lambertian objects in the image should be labeled as background, and the refractive transparent object should be labeled as the foreground. As illustrated in Fig. 6.2, the background and the occluded areas (shown in blue and green) should be labeled as the background, and the transparent object (orange) should be labeled as foreground.

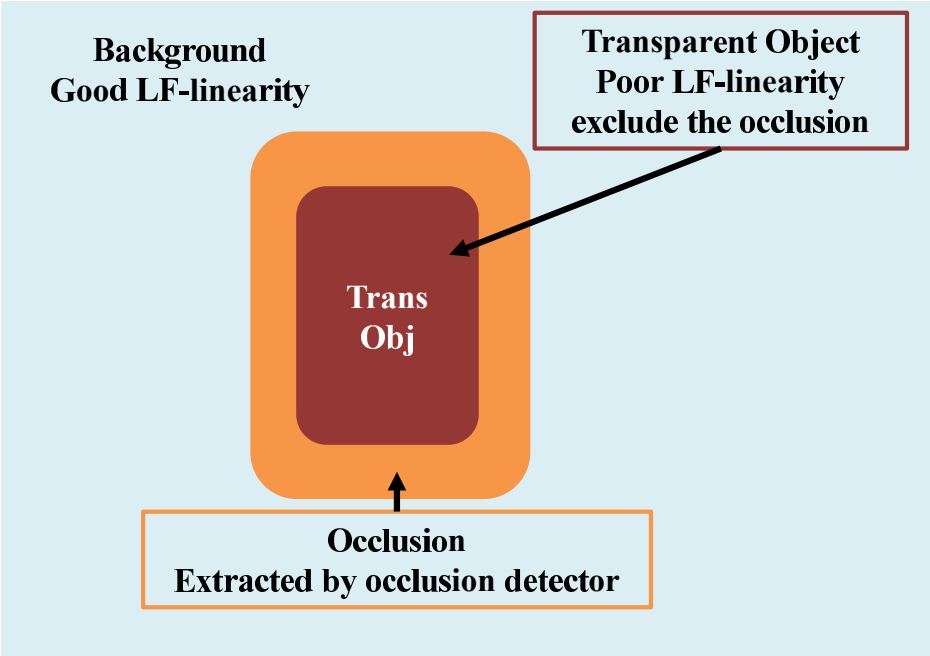


Fig. 6.2 Properties of different components in an image containing a transparent object. The Lambertian background (blue) has good LF-consistency , the transparent object (red) has poor LF-linearity exlude the occlusion area , and the occlusion boundary (orange) can be detected by occlusion detector .

The Lambertian object has good LF-linearity while the transparent object has poor LF-linearity. The occlusion area also has poor LF-linearity and can be detected by the occlusion detector, so the transparent object locates in area with poor LF-linearity other than the occlusion area. The case of the occlusion area with good LF-linearity rarely occurs because, when the forward-backward matching is not consistent, the LF-linearity will be poor. Therefore, the region with good LF-linearity should be background. When a pixel belongs to the background, the penalty for labeling this pixel as a Lambertian object or occlusion area should be low, while the penalty for labeling this pixel as part of a transparent object should be high. The opposite is true when a pixel belongs to the foreground.

Before defining the regional term of the energy function, the LF-linearity $E(u,v)$ is first scaled to the range $[0,1]$ using a sigmoid function:

$$\tilde{E}_p = sigmoid(E(u,v),a,b), \tag{6.3}$$

where $sigmoid(\varphi,a,b)$ is the function:

$$sigmoid(\varphi,a,b) = \frac{1}{1+exp(-a(\varphi-b))}, \tag{6.4}$$

a controls the steepness of the function, and b is the shift, which acts as the threshold value here.

The regional term for a pixel p is defined as:

$$R_p(0) = \beta \tilde{E}_p \cdot (1 - \tilde{O}_p), \tag{6.5}$$

$$R_p(1) = \tilde{E}_p \cdot \tilde{O}_p + (1 - \tilde{E}_p), \tag{6.6}$$

where $\tilde{O}_p = O(u, v, \tilde{\theta})$, which is the maximum response from the occlusion detectors descried in Eq. 4.7 and Eq. 4.8. $R_p(0)$ assigns a large penalty to pixels that have poor LF-linearity exclude the occlusion area, and $R_p(1)$ assigns a large penalty to pixels with poor LF-linearity inside the occlusion area or pixels with good LF-linearity. β adjusts the balance between $R_p(0)$ and $R_p(1)$.

6.2.2 Boundary term

In the boundary term of the energy function, the pairwise potentials between two neighboring pixels should be defined. The proposed method is defined in the 4-neighbor system, so each pixel has two horizontal neighboring pixels and two vertical neighboring pixels. We utilize the maximum response of the occlusion detectors (Eq. 4.7) to assign pairwise potentials.

The boundary term applies a penalty when neighboring pixels p, q are assigned different labels. Given a pixel p (see Fig. 6.3), the weight of its 4 neighboring edges can be described as:

$$\begin{cases} w_{p,q_1} = \tilde{O}_p \\ w_{p,q_2} = w_{p,q_3} = w_{p,q_4} = 0 \end{cases} \text{, if } \tilde{\theta} = 0, \tag{6.7}$$

$$\begin{cases} w_{p,q_1} = w_{p,q_2} = \tilde{O}_p / \sqrt{2} \\ w_{p,q_3} = w_{p,q_4} = 0 \end{cases} \text{, if } \tilde{\theta} = 45, \tag{6.8}$$

and so forth. The entire definition of the edge weight can be found in the Appendix A.

The weight for each edge is calculated twice as $w_{p,q}$ and $w_{q,p}$, and the penalty for assigning different labels to p and q is defined as:

$$B_{p,q} = exp(-\gamma \cdot (w_{p,q} + w_{q,p})). \tag{6.9}$$

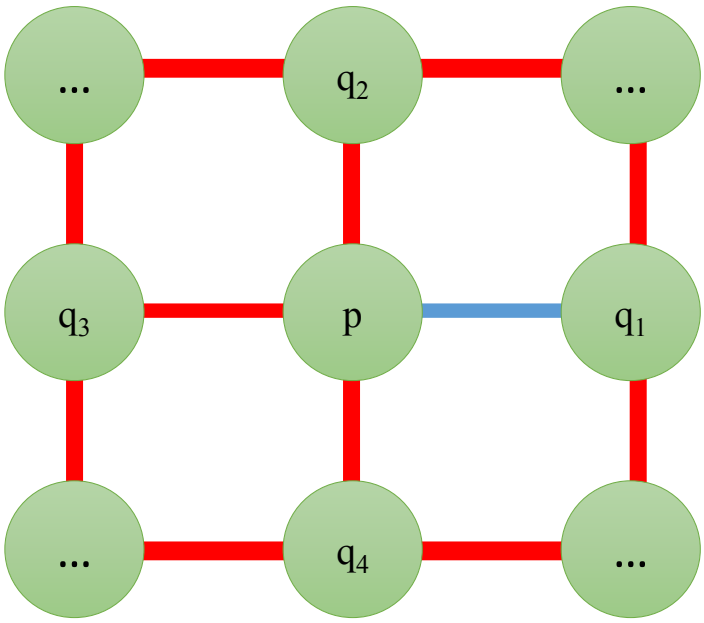


Fig. 6.3 Definition of energy for the pairwise potential $B_{p,q}$. The example shows the maximum response $O_{p,\tilde{\theta}}$ comes from $\tilde{\theta} = 0$, hence a small penalty B_{p,q_1} is assigned to the corresponding edge (blue)

The weight is small in the background and foreground regions. The penalty of the region is high in the case of assigning different labels to the neighboring pixels. It works easy to propagate the same labels in the same regions. In contrast, the occlusion boundary will have large values of $\tilde{O}_p$, and it stop to propagate the label between the different regions. γ controls the rate of the importance of the penalty.

6.3 Experiments

As there are no light field datasets available for the evaluation of transparent object segmentation, the author captured the necessary light field data. The author shall demonstrate the proposed transparent object segmentation method on various examples, including single and multiple objects segmentation with different backgrounds. The author also compared the results with those given by LF-linearity thresholding and the finding glass method [21].

6.3.1 Assumptions

To ensure the effectiveness of the matching process, the experiments were conducted under the following assumptions:

- All viewpoints of the light-field camera can capture the entirety of the target objects.

- The degree of reflection on the surface of the target objects is relatively low.
- The background is relatively far away from the target objects.
- The background scenes have sufficient textural information.

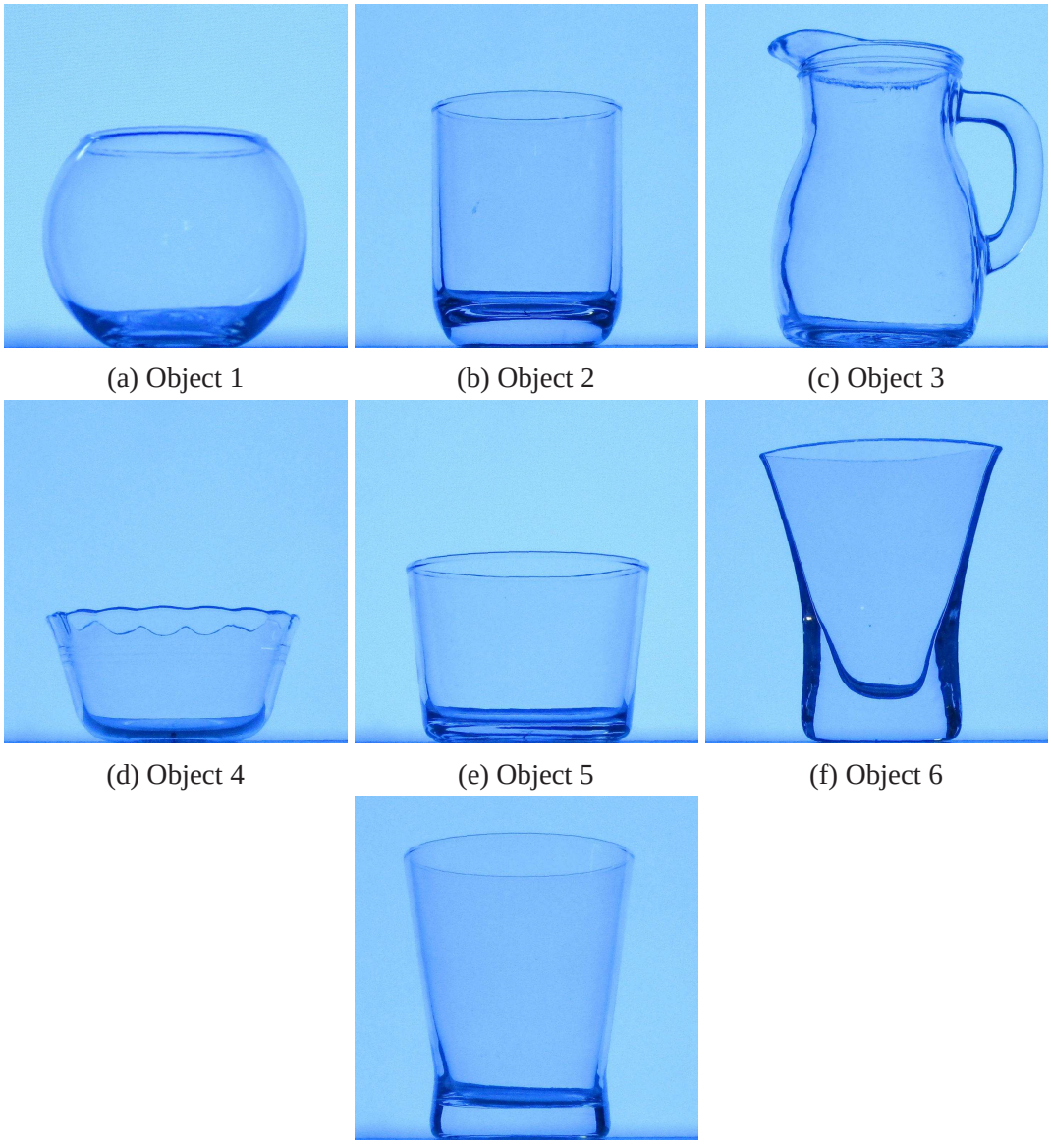
6.3.2 Results and discussion

In the experiments, the author used a light-field camera with 5×5 viewpoints (ProFusion 25, Viewplus Inc.) to acquire the images. The author placed the target objects about 50 cm from the camera, with the background a further 100 cm behind the objects. The author captured seven transparent objects (shown in Fig. 6.4) with seven different background scenes (shown in Fig. 6.5). The backgrounds include indoor scenes such as a library and outdoor scenes such as a city backdrop seen through a window.

As mentioned in Sec. 4.3, an optical flow algorithm is used to obtain the corresponding points p, p' and p'' in the central viewpoint $view(0,0)$ and the other viewpoints $view(s,t)$. The author utilizes the optical flow algorithm proposed in [73], which integrates descriptor matching into variational motion estimation. Although this optical flow algorithm is very accurate, it cannot deal with textureless regions, and such areas will cause problems when the matching is not correct. For this reason, the author removes those textureless regions for which the squared horizontal intensity gradient averaged over a square window of a given size is below a given threshold [91]. The parameters used in all experiments are fixed to the same values. The parameters α, β, γ are determined based on the preliminary experiments, and set $\alpha = 70$, $\beta = 4.5$, $\gamma = 4.5$ which are suitable for the captured dataset. Parameter a decide the thresholding is hard or soft, b is determined by the level of least-squares error, and τ is related to the accuracy of optical flow and image resolution. The values, which are suitable for the captured dataset, are selected as $a = 0.5$, $b = 5$ and $\tau = 8$.

The author compared the proposed segmentation results with those from LF-linearity thresholding and the finding glass method. For the thresholding method, the system simply filtered out the Lambertian background by removing feature points whose least-squares error $E(u,v)$ is below a certain threshold, i.e., $E(u,v) < th$. In the experiments, the author set th=7 which is the optimum value for this method. The finding glass method [21] utilized 6 different cues from the appearance, such as color, blurriness and texture, to detect the edges in the glass region. The author implemented this method as described in [21] and applied to the central view of the captured dataset.

Figure 6.6 shows the results for the same scene with different objects, and Fig. 6.7 shows the segmentation results for the same object with different backgrounds. We can see that, simply LF-linearity thresholding will result in holes inside the target object at points where



(a) Object 1 (b) Object 2 (c) Object 3

(d) Object 4 (e) Object 5 (f) Object 6

(g) Object 7

Fig. 6.4 seven transparent objects of various shapes for the experiments.

(a) Scene 1 (b) Scene 2 (c) Scene 3

(d) Scene 4 (e) Scene 5 (f) Scene 6

(g) Scene 7

Fig. 6.5 Seven different backgrounds for the experiments. These include indoor and outdoor scenes in both day and night.

	F-measure	Recall	Precision
Finding glass	0.30	0.82	0.19
LF-linearity thresholding	0.50	0.65	0.41
Proposed method	0.85	0.96	0.77

Table 6.1 Quantitative comparison of three methods. The results are averaged over the single object dataset with 7 objects and 7 scenes.

the light field is nearly linear, and mismatched regions from outside will be included in the object. The finding glass method falsely detected the rich texture background as glass, since this method is not suitable for rich texture images, which is mentioned as the limitation in the paper. The proposed TransCut method gives very stable results for various objects in different scenes. Further results can be found in the Appendix B.

The author determined the ground truth by manually labeling all pixels, and quantitatively compared the segmentation results. This comparison is tabulated in Table 6.1. F-measure has been used to compare the performance of each algorithm. This metric is the harmonic mean of the precision and recall, i.e.,

$$F = \frac{2 * Precision * Recall}{Precision + Recall}, \tag{6.10}$$

where $Recall = TP/(TP+FN)$, and $Precision = TP/(TP+FP)$ (TP=True Positive, FN=False Negative, FP=False Positive).

Different transparent object combinations in seven different scenes were performed in the experiments. The results of experiments including multiple objects are shown in Fig. 6.8 and Fig. 6.9. These images show that the proposed method is effective when there is more than one object in the scene, whereas the other two methods do not produce good results in such scenarios.

Moreover, the experiments with real scene were conducted as well. Four different objects were captured in the real scene as shown in the left column of Fig. 6.10, and the segmentation results are shown in the right column. The results show that the proposed method works successfully in the real scene although it is not perfect.

6.4 Summary

In this chapter, the LF-linearity and occlusion detector are applied to transparent object segmentation task. The author reviewed several related works of the image segmentation

Fig. 6.6 Comparison of segmentation results for the same scene with different objects. The 1st row shows the image from the central viewpoint. The 2nd, 3rd, and 4th rows show output from the finding glass, LF-linearlity thresholding, and proposed TransCut methods, respectively. The last row shows the manually labeled ground truth.

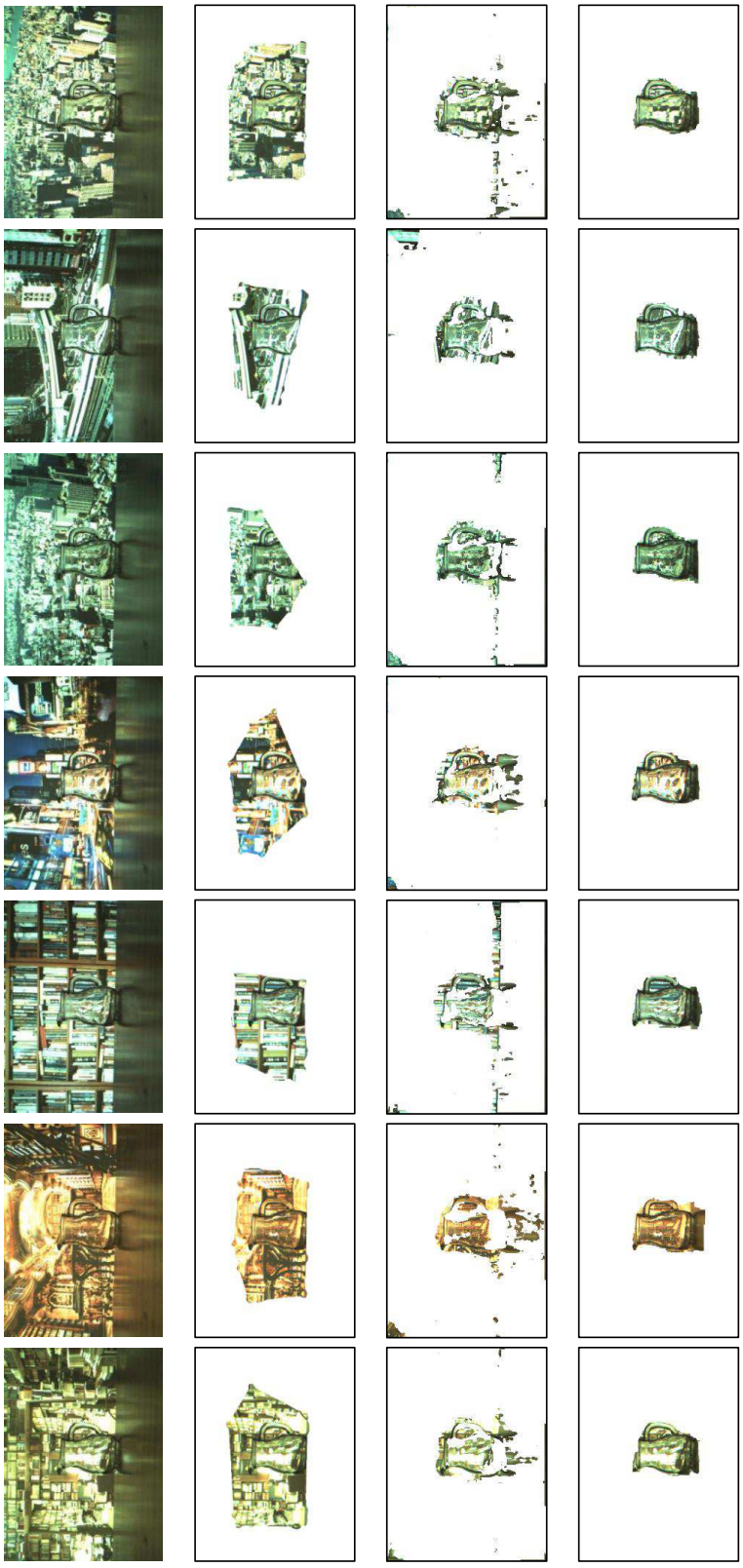


Fig. 6.7 Comparison of segmentation results for the same object in different scenes. The 1st row shows the image from the central viewpoint. The 2nd, 3rd, and 4th rows show output from the finding glass, LF-linearity thresholding, and proposed TransCut methods, respectively. The author refers to the 3rd object in the last row of Fig. 6.6 for the ground truth.

Fig. 6.8 Comparison of segmentation results for object 4 and object 5 combination in different scenes. The 1st row shows the image from the central viewpoint. The 2nd, 3rd, and 4th rows show output from the finding glass, LF-linearity thresholding, and proposed TransCut methods, respectively.

Fig. 6.9 Comparison of segmentation results for object 6 and object 7 combination in different scenes. The 1st row shows the image from the central viewpoint. The 2nd, 3rd, and 4th rows show output from the finding glass, LF-linearity thresholding, and proposed TransCut methods, respectively.

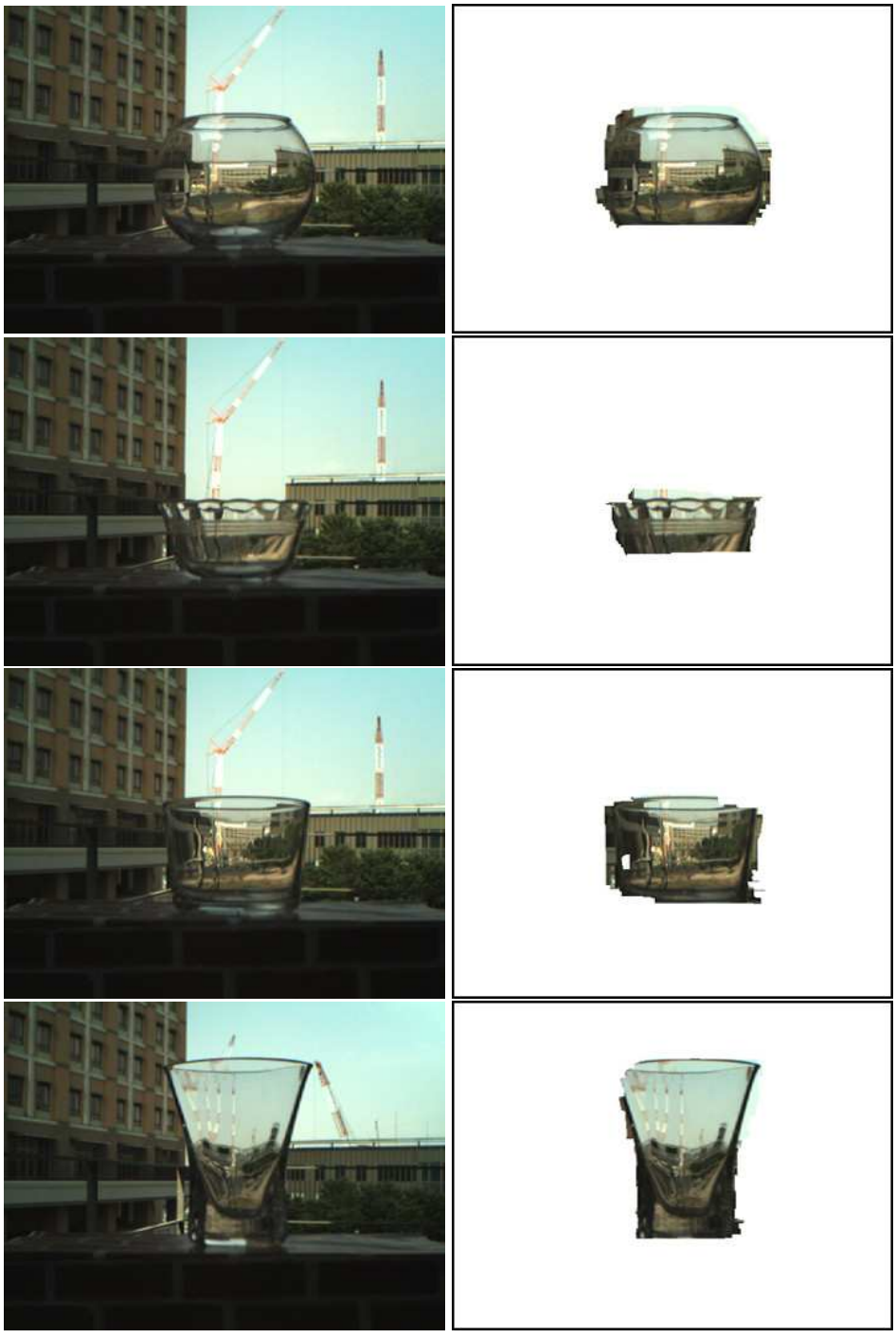


Fig. 6.10 Four different objects in the real scene. The left column shows the central view of the light field image, and the right column shows the segmentation result.

task in the beginning, and then described the proposed method. Finally, the experimental results show the proposed method using light field image can get stable and impressive results, which outperforms the conventional method using an image captured by a regular camera.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis, the author introduces that some computer vision problems can be solved by light field image, and this new technique is called light field vision.

The author presented a model of a camera array for light field acquisition, which includes a rigid constraint between viewpoints. A calibration algorithm with global optimization based on the rigid-constrained model was proposed for the light field camera. The proposed algorithm extends Zhang's well-known method, and consists of two steps: closed form initialization and non-linear refinement. It is rather faster than conventional calibration methods that do not have the rigid-constraint. The proposed method is not restricted with respect to the number of viewpoints or alignment of cameras. Simulation experiments show that the proposed algorithm yields better results than the linear solution, the method where each viewpoint is refined independently, and the Bundler. The proposed method was also applied to the calibration of a commercial light field camera, and the results show that all intrinsic and extrinsic parameters were optimized with a total re-projection error smaller than 0.4 pixels. The author also performed a digital refocusing experiment on a captured light field image. The calibrated light field image was effectively refocused on the required target, whereas this was not possible for the rough calibrated light field image.

Compared with conventional cameras, which capture 2D images from a single perspective, light field cameras obtain richer 4D images that include both the angular and positional information of the light ray space. The new feature extracted from the light field image also has more advantages than the features from the object appearance. The LFD feature models the distortion from differences in corresponding points between viewpoints in the 4D light field, whereas common features, such as gradients or edges, model the appearance. This is an entirely original concept for feature description with the advantage that LFD is less affect-

ed by background changes, as it uses patterns of ray distortions caused by the transparent objects, not patterns from the appearance of objects. LF-linearity is an important property of LFD feature, and the LF-linearity can describe the likelihood of a light ray comes from Lambertian object or not. The proposed occlusion detector is very useful to describe the light rays are occluded in the reference viewpoint or not.

LFD feature and LF-linearity are successfully used in the transparent object categorization application. The proposed transparent object categorization method achieved on average 84% accuracy with 18 objects in lab setting and 53.3% in the real scene setting, while standard SIFT is not working at all. The author discussed about parameter optimizations and limitation analysis of the method in some experiments, such as: threshold value of the LFD, number of clusters, density of the texture, camera and background positions, object rotations, lighting conditions and the number of cameras.

LF-linearity and occlusion detector have been applied to the segmentation of transparent objects. Unlike conventional methods, the proposed technique does not rely on color information to distinguish the foreground and background. The author has used LF-linearity and occlusion detector in 4D light field space for describing a transparent object, and designed an appropriate energy function utilizing the LF-linearity and occlusion for pixel labeling by graph-cut. The results show that this method produces stable results with various objects in different scenes.

7.2 Future Work

The work presented in this thesis still has many limitations, and there are several possible directions to improve the current results.

The datasets in this thesis are captured by camera array where the camera baseline is large and viewpoint number is few. An straightforward future step is to capture the light field image using a lenslet camera, such as Lytro Illum, and extend the proposed algorithms to be suitable for the new dataset. The current results are not yet perfect, as the assumptions produce some limitations. The feature descriptors can be designed in a more sophisticated way in order to overcome the limitation in scale and rotation change. Thereafter, the new feature descriptors can be used in more flexible environment. For the object categorization application, it is obvious that good segmentation can help get good categorization results, so combining the proposed segmentation method with categorization task is a meaningful future work.

The work presented in this thesis introduces light field vision, which opens a door to solve the problems that conventional computer vision techniques cannot deal with, and it

suggests several directions for new research. Here are few more possible directions are worthy to explore.

Light field vision for non-Lambertian object recognition and reconstruction.

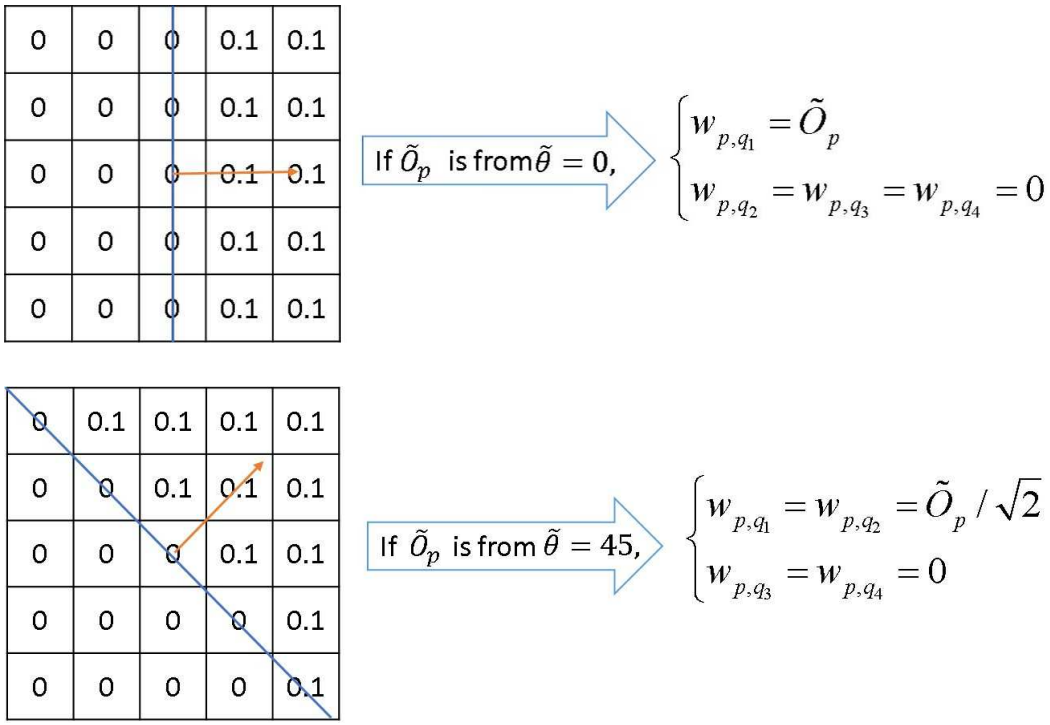
In this thesis, the author applies the light field vision to transparent object recognition, and it will be more interesting if the target objects are generalized to all the non-Lambertian object including the specular and glossy objects. Moreover, because the light field camera captures rich 4D information of the scene, the 3D object surface can be reconstructed from the a single-shot light field image. Object recognition can be performed for the reconstructed 3D object which will make the recognition more robust.

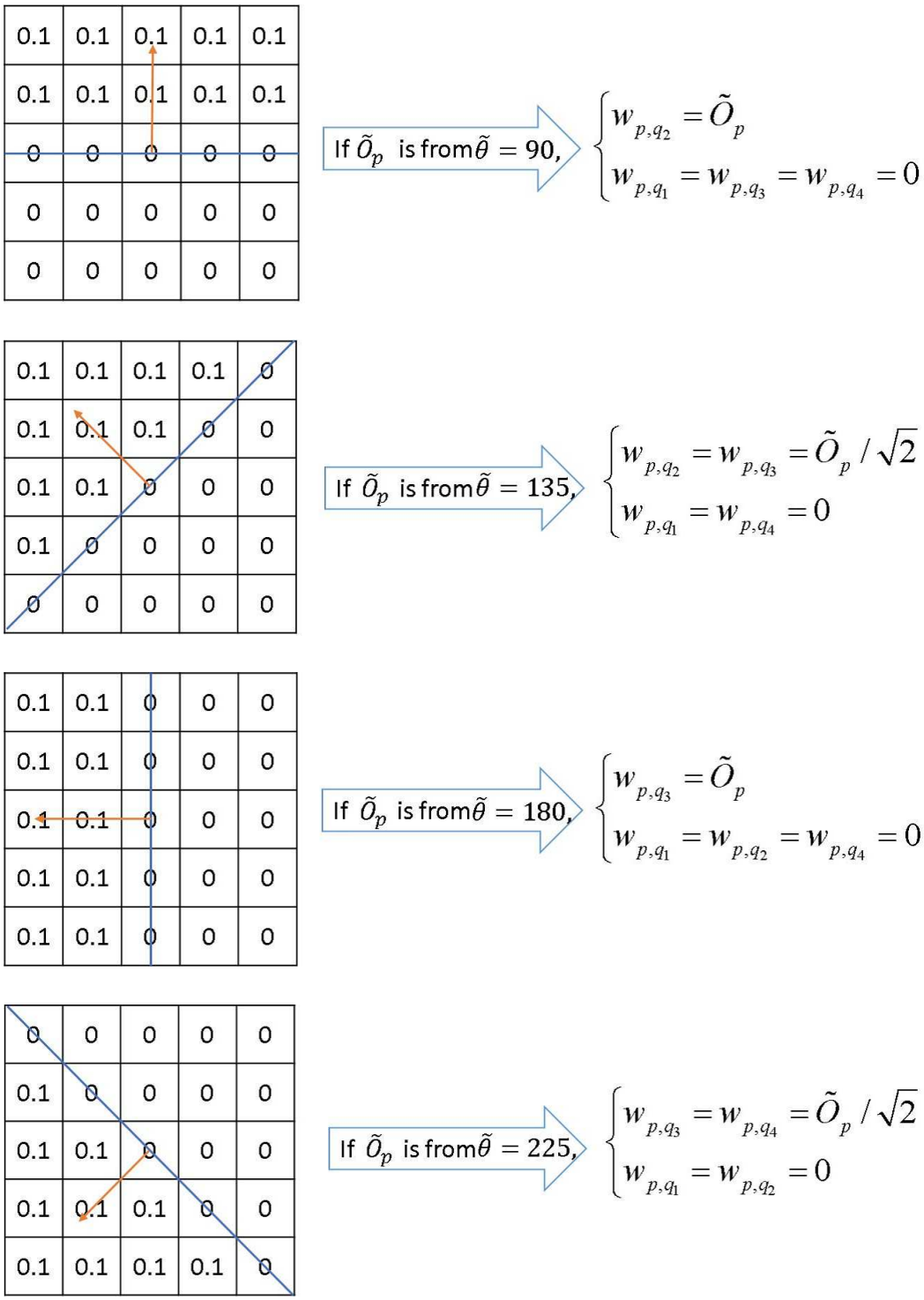
Machine learning for light field vision.

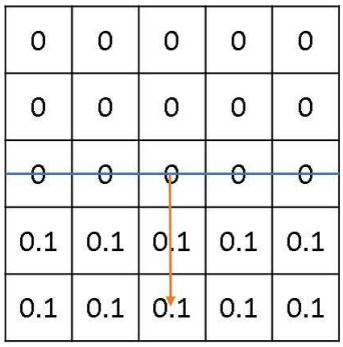
The author uses Bag-of-Feature method for transparent object categorization in this thesis. Other machine learning methods can also apply to the light field vision. The features extracted from the light field images have more information than the features from a single-view image. Machine learning methods can utilize the features from light field images to train a sophisticated model in order to deal with different types of applications.

Appendix A

Entire definition of the edge weight

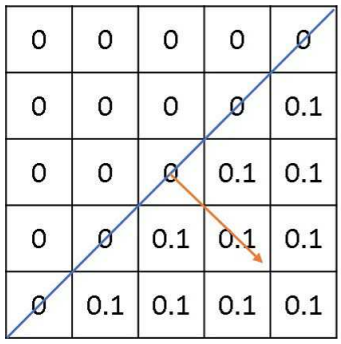



0.1 0.1 0.1 0.1 0.1
0.1 0.1 0.1 0.1 0.1
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
If $\tilde{O}_p$ is from $\tilde{\theta} = 90$,
$w_{p,q_2} = \tilde{O}_p$
$w_{p,q_1} = w_{p,q_3} = w_{p,q_4} = 0$
0.1 0.1 0.1 0.1 0
0.1 0.1 0.1 0 0
0.1 0.1 0 0 0
0.1 0 0 0 0
0 0 0 0 0
If $\tilde{O}_p$ is from $\tilde{\theta} = 135$,
$w_{p,q_2} = w_{p,q_3} = \tilde{O}_p / \sqrt{2}$
$w_{p,q_1} = w_{p,q_4} = 0$
0.1 0.1 0 0 0
0.1 0.1 0 0 0
0.1 0.1 0 0 0
0.1 0.1 0 0 0
0.1 0.1 0 0 0
If $\tilde{O}_p$ is from $\tilde{\theta} = 180$,
$w_{p,q_3} = \tilde{O}_p$
$w_{p,q_1} = w_{p,q_2} = w_{p,q_4} = 0$
0 0 0 0 0
0.1 0 0 0 0
0.1 0.1 0 0 0
0.1 0.1 0.1 0 0
0.1 0.1 0.1 0.1 0
If $\tilde{O}_p$ is from $\tilde{\theta} = 225$,
$w_{p,q_3} = w_{p,q_4} = \tilde{O}_p / \sqrt{2}$
$w_{p,q_1} = w_{p,q_2} = 0$



If $\tilde{O}_p$ is from $\tilde{\theta} = 270$,

$$\begin{cases} w_{p,q_4} = \tilde{O}_p \\ w_{p,q_1} = w_{p,q_2} = w_{p,q_3} = 0 \end{cases}$$



If $\tilde{O}_p$ is from $\tilde{\theta} = 315$,

$$\begin{cases} w_{p,q_1} = w_{p,q_4} = \tilde{O}_p / \sqrt{2} \\ w_{p,q_2} = w_{p,q_3} = 0 \end{cases}$$

Appendix B

Additional results for transparent object segmentation

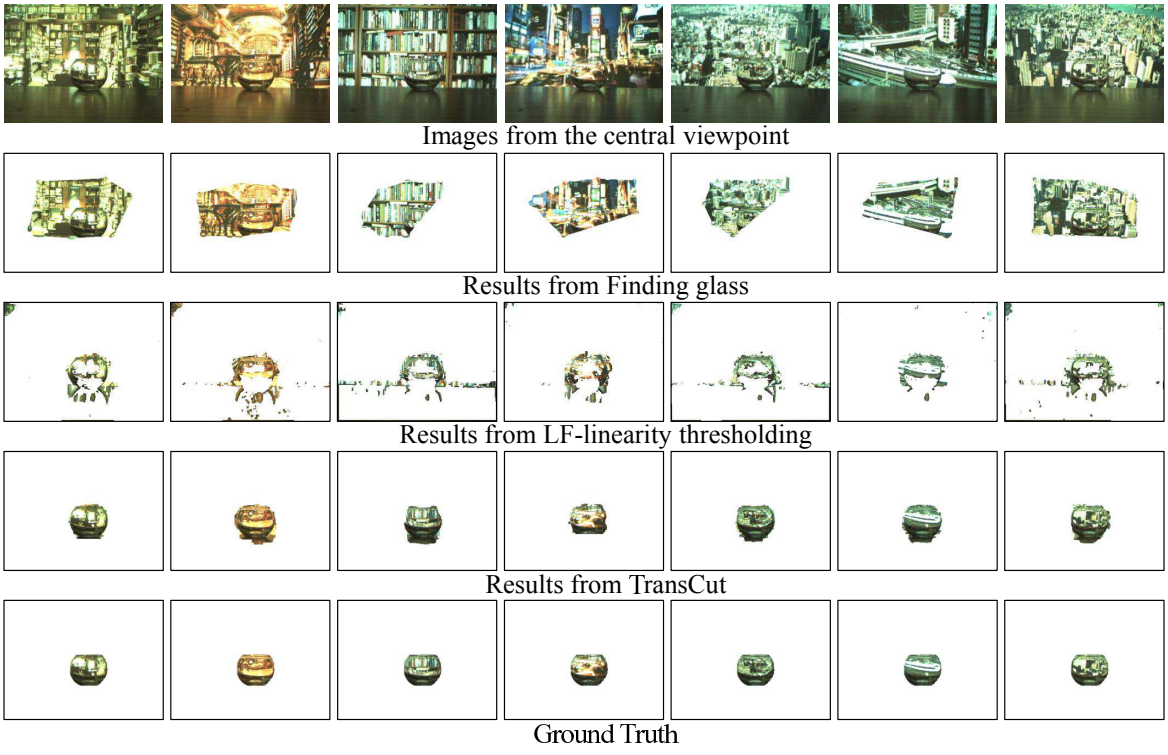


Fig. B.1 Object 1.

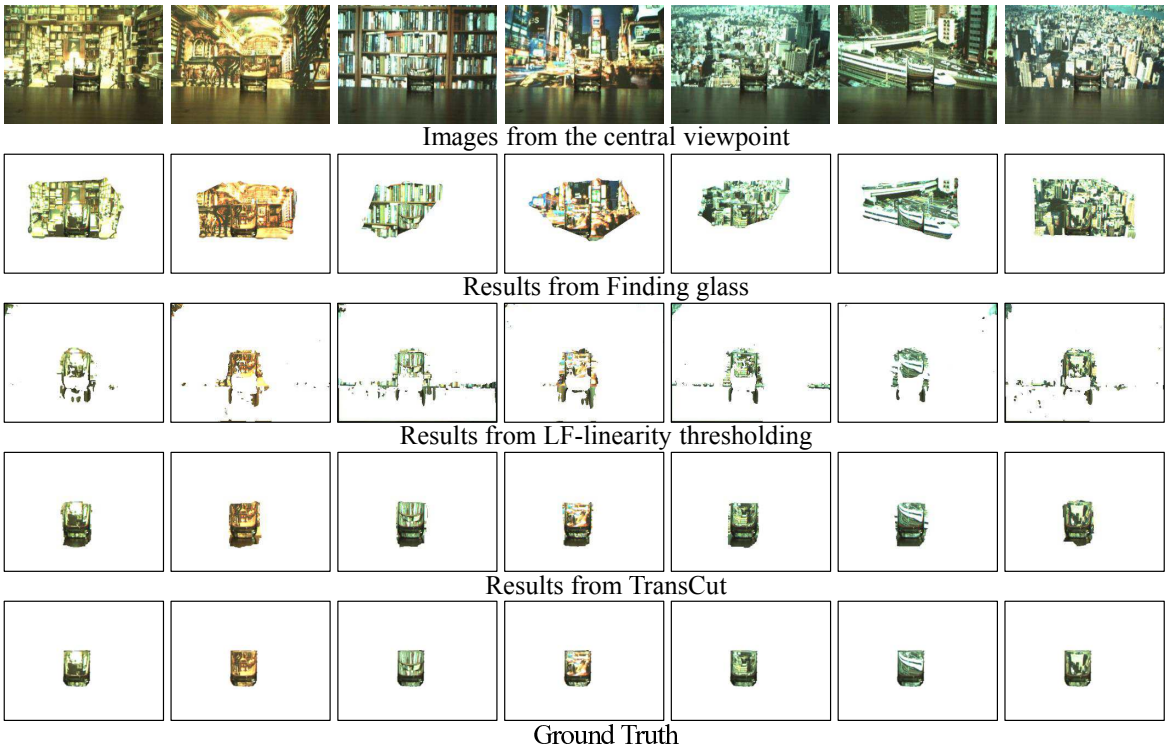


Fig. B.2 Object 2.

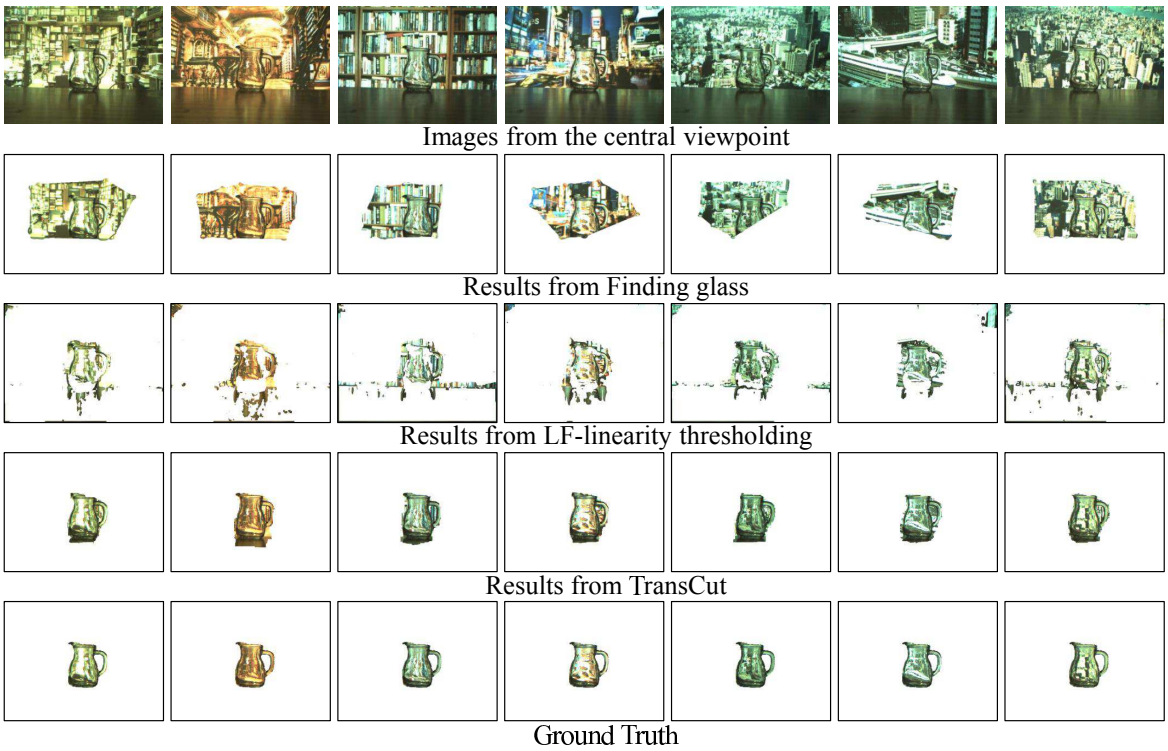


Fig. B.3 Object 3.

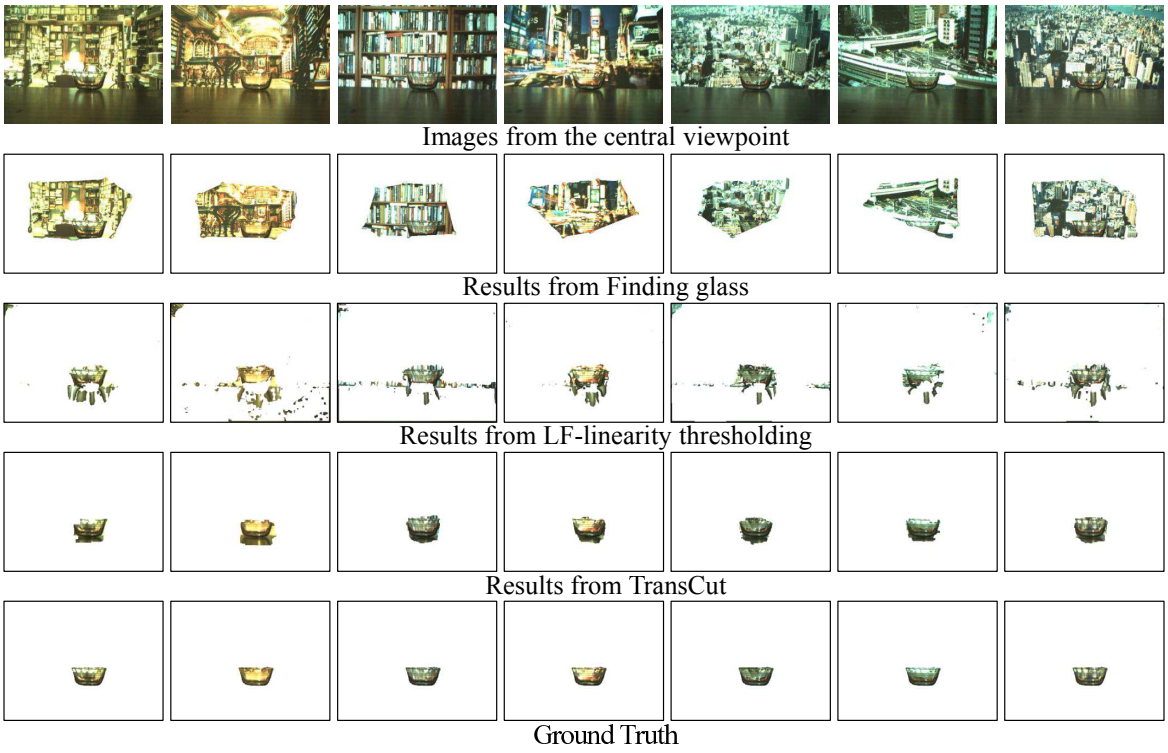


Fig. B.4 Object 4.

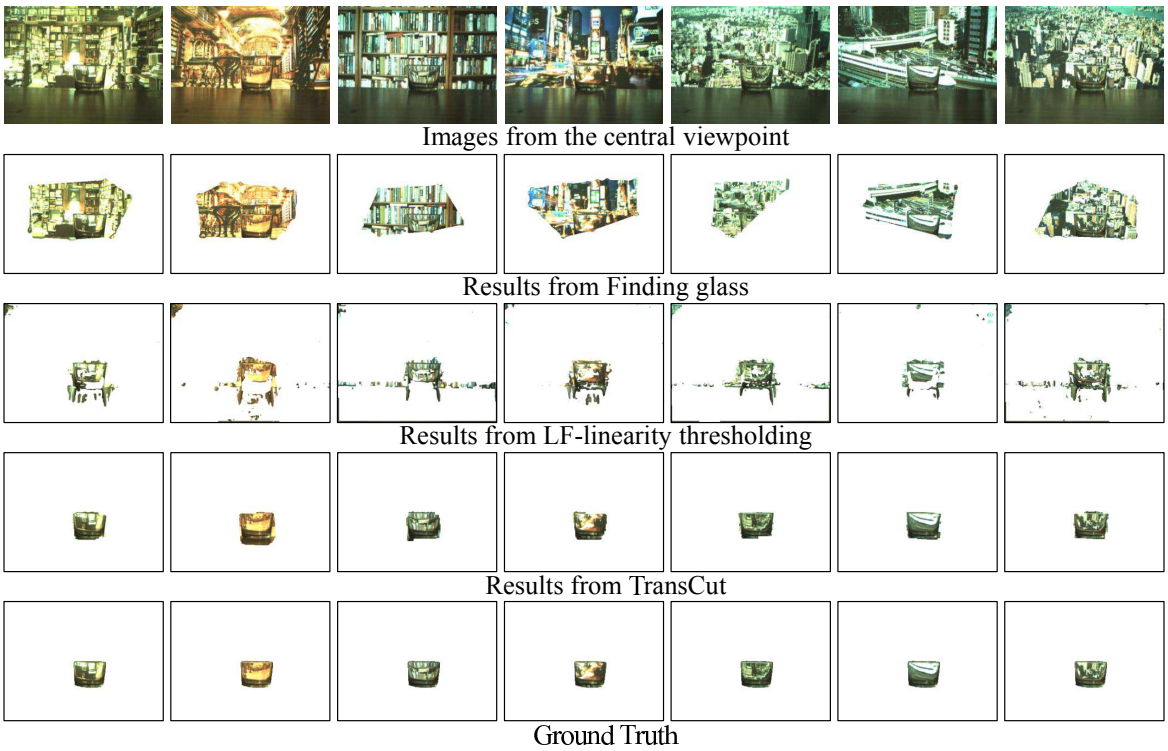


Fig. B.5 Object 5.

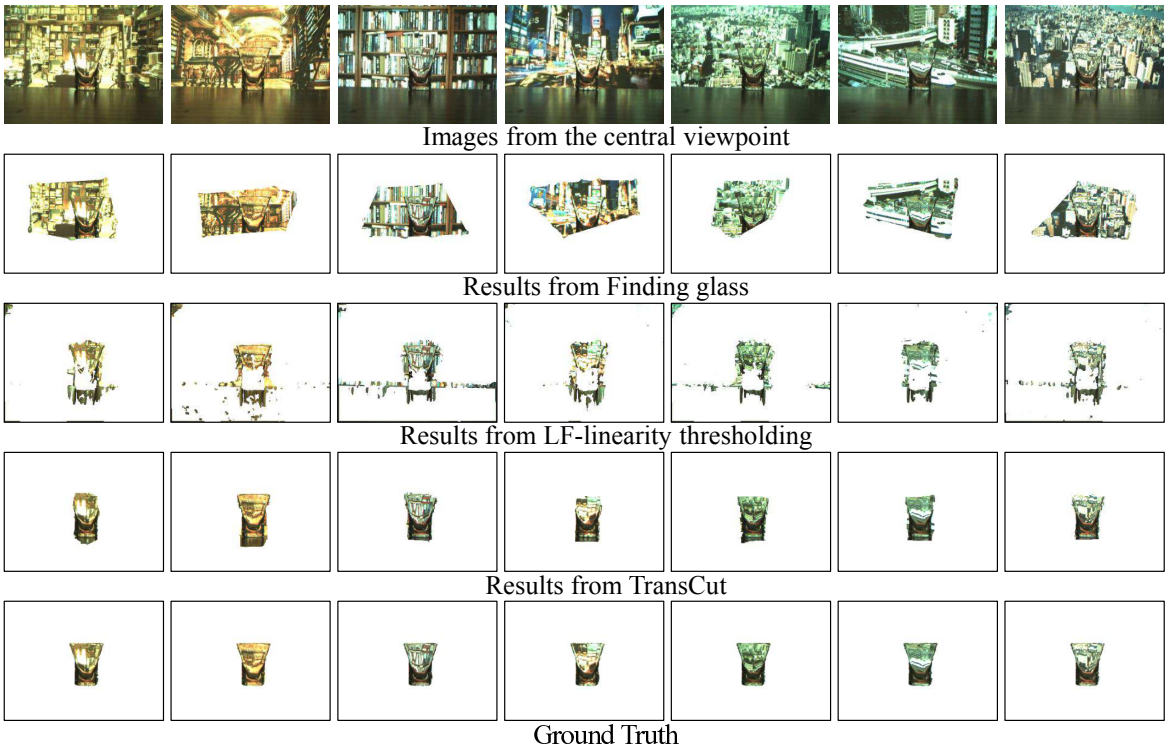


Fig. B.6 Object 6.

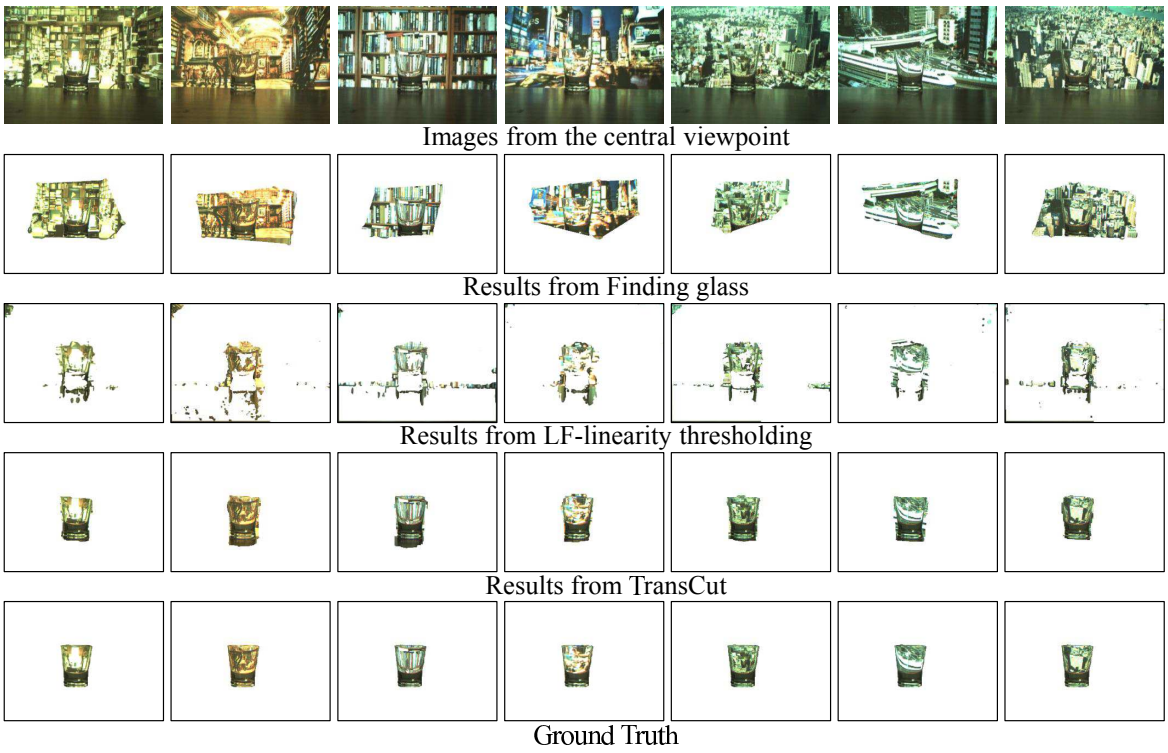


Fig. B.7 Object 7.

References

[1] D. Crevier, *AI: the tumultuous history of the search for artificial intelligence*. Basic Books, 1993.

[2] http://www.mobvis.org/index.htm.

[3] http://www.nvidia.com/object/drive px.html.

[4] O. Linde and T. Lindeberg, "Composed complex-cue histograms: An investigation of the information content in receptive field based image descriptors for object recognition," *Computer Vision and Image Understanding*, vol. 116, no. 4, pp. 538 – 560, 2012.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[6] E. H. Adelson and J. R. Bergen, *The plenoptic function and the elements of early vision*. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1991.

[7] T. Georgiev and A. Lumsdaine, "Focused plenoptic camera and rendering," *Journal of Electronic Imaging*, vol. 19, no. 2, pp. 021106–021106, 2010.

[8] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the ACM Conference on Computer Graphics*, pp. 31–42, 1996.

[9] I. Ihm, S. Park, and R. K. Lee, "Rendering of spherical light fields," in *Proceedings of the Fifth Pacific Conference On Computer Graphics And Applications*, Pacific Graphics '97, pp. 59–68, 1997.

[10] http://graphics.stanford.edu/projects/gantry/.

[11] http://www.viewplus.co.jp/product/camera/profusion25.html.

[12] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," *Computer Science Technical Report CSTR*, 2005.

[13] A. Lumsdaine and T. Georgiev, "The focused plenoptic camera," in *IEEE International Conference on Computational Photography (ICCP)*, pp. 1–8, 2009.

[14] http://www.lytro.com/.

[15] http://www.raytrix.de/.

[16] B. Wilburn, N. Joshi, V. Vaish, E.-V. E. Talvala, E. Antunez, A. Barth, A. Adams, M. Levoy, and M. Horowitz, "High performance imaging using large camera arrays," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 765–776, 2005.

[17] K. Venkataraman, D. Lelescu, J. Duparré, A. McMahon, G. Molina, P. Chatterjee, R. Mullis, and S. Nayar, "Picam: An ultra-thin high performance monolithic camera array," *ACM Trans. Graph.*, vol. 32, pp. 166:1–166:13, Nov. 2013.

[18] A. Shimada, H. Nagahara, and R. ichiro Taniguchi, "Object detection based on spatio-temporal light field sensing," *IPSJ Transactions on Computer Vision and Applications*, vol. 5, pp. 129–133, 2013.

[19] S. Wanner and B. Goldluecke, "Variational light field analysis for disparity estimation and super-resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[20] N. Li, J. Ye, Y. Ji, H. Ling, and J. Yu, "Saliency detection on light field," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2806–2813, IEEE, 2014.

[21] K. McHenry, J. Ponce, and D. Forsyth, "Finding glass," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 973–979, 2005.

[22] Y. Xu, K. Maeno, H. Nagahara, and R. Taniguchi, "Mobile camera array calibration for light field acquisition," in *International Conference on Quality Control by Artificial Vision (QCAV)*, pp. 283–290, 5 2013.

[23] Y. Xu, K. Maeno, H. Nagahara, A. Shimada, and R. Taniguchi, "Light field distortion feature for transparent object classification," *Computer Vision and Image Understanding*, 2015.

[24] Y. Xu, K. Maeno, H. Nagahara, and R. Taniguchi, "Camera array calibration for light field acquisition," *Frontiers of Computer Science*, pp. 1–12, 2015.

[25] http://cyberware.com/products/scanners/sphere.html.

[26] T. A. Harvey, K. S. Bostwick, and S. Marschner, "Measuring spatially-and directionally-varying light scattering from biological material," *Journal of visualized experiments: JoVE*, no. 75, 2013.

[27] M. Holroyd, J. Lawrence, and T. Zickler, "A coaxial optical scanner for synchronous acquisition of 3d geometry and surface reflectance," *ACM Trans. Graph.*, vol. 29, pp. 99:1–99:12, July 2010.

[28] http://lightfield.stanford.edu/acq.html.

[29] A. Isaksen, L. McMillan, and S. J. Gortler, "Dynamically reparameterized light fields," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 297–306, ACM Press/Addison-Wesley Publishing Co., 2000.

[30] C.-K. Liang, G. Liu, and H. H. Chen, "Light field acquisition using programmable aperture camera," in *IEEE International Conference on Image Processing*, vol. 5, pp. V–233, IEEE, 2007.

[31] H. Nagahara, C. Zhou, T. Watanabe, H. Ishiguro, and S. K. Nayar, "Programmable aperture camera using lcos," in *Computer Vision–ECCV 2010*, pp. 337–350, Springer, 2010.

[32] C.-K. Liang, T.-H. Lin, B.-Y. Wong, C. Liu, and H. H. Chen, "Programmable aperture photography: multiplexed light field acquisition," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, p. 55, 2008.

[33] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin, "Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing," *ACM Trans. Graph.*, vol. 26, July 2007.

[34] R. Ng, "Fourier slice photography," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 735–744, 2005.

[35] V. Vaish, M. Levoy, R. Szeliski, C. L. Zitnick, and S. B. Kang, "Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures," in *CVPR*, pp. 2331–2338, 2006.

[36] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. S. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *CVPR*, pp. 519–528, 2006.

[37] D. Miyazaki, M. Kagesawa, and K. Ikeuchi, "Transparent surface modeling from a pair of polarization images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 73–82, 2004.

[38] D. Miyazaki and K. Ikeuchi, "Inverse polarization raytracing: estimating surface shapes of transparent objects," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 910–917, 2005.

[39] G. Settles, *Schlieren and shadowgraph techniques: visualizing phenomena in transparent media*. Springer, 2001.

[40] G. S. Settles, "Important developments in schlieren and shadowgraph visualization during the last decade," in *International Symposium on Flow Visualization (ISFV)*, 2010.

[41] M. Alterman, Y. Y. Schechner, M. Vo, and S. G. Narasimhan, "Passive tomography of turbulence strength," in *European Conference on Computer Vision (ECCV)*, pp. 47–60, Springer, 2014.

[42] M. Alterman, Y. Swirski, Y. Schechner, *et al.*, "Stella maris: Stellar marine refractive imaging sensor," in *IEEE International Conference on Computational Photography (ICCP)*, pp. 1–10, IEEE, 2014.

[43] G. Wetzstein, R. Raskar, and W. Heidrich, "Hand-held schlieren photography with light field probes," in *IEEE International Conference on Computational Photography (ICCP)*, pp. 1–8, 2011.

[44] G. Wetzstein, D. Roodnick, W. Heidrich, and R. Raskar, "Refractive shape from light field distortion," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 1180–1186, 2011.

[45] Y. Ji, J. Ye, and J. Yu, "Reconstructing gas flows using light-path approximation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2507–2514, 2013.

[46] Y. Ding, F. Li, Y. Ji, and J. Yu, "Dynamic fluid surface acquisition using a camera array," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 2478–2485, 2011.

[47] J. Ye, Y. Ji, F. Li, and J. Yu, "Angular domain reconstruction of dynamic 3d fluid surfaces," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 310–317, 2012.

[48] K. McHenry and J. Ponce, "A geodesic active contour framework for finding glass," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1038–1044, 2006.

[49] M. Fritz, M. J. Black, G. R. Bradski, S. Karayev, and T. Darrell, "An additive latent feature model for transparent object recognition," in *Neural Information Processing Systems (NIPS)*, 2009.

[50] T. Wang, X. He, and N. Barnes, "Glass object localization by joint inference of boundary and depth," in *International Conference on Pattern Recognition (ICPR)*, pp. 3783–3786, 2012.

[51] T. Wang, X. He, and N. Barnes, "Glass object segmentation by label transfer on joint depth and appearance manifolds," in *IEEE International Conference on Image Processing (ICIP)*, pp. 2944–2948, 2013.

[52] A. Criminisi, S. B. Kang, R. Swaminathan, R. Szeliski, and P. Anandan, "Extracting layers and analyzing their specular properties using epipolar-plane-image analysis," *Computer Vision and Image Understanding*, vol. 97, no. 1, pp. 51 – 85, 2005.

[53] M. Ben-Ezra and S. K. Nayar, "What does motion reveal about transparency?," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 1025–1032, 2003.

[54] S. Agarwal, S. P. Mallick, D. Kriegman, and S. Belongie, "On refractive optical flow," in *European Conference on Computer Vision (ECCV)*, pp. 483–494, Springer, 2004.

[55] N. J. Morris and K. N. Kutulakos, "Dynamic refraction stereo," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 1573–1580, 2005.

[56] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 323–344, 1987.

[57] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[58] R. Horaud, G. Csurka, and D. Demirdijian, "Stereo calibration from rigid motions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1446–1452, 2000.

[59] H. Malm and A. Heyden, "Stereo head calibration from a planar object," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. II–657, 2001.

[60] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy, "Using plane + parallax for calibrating dense camera arrays," in *CVPR (1)*, pp. 2–9, 2004.

[61] T. Svoboda, D. Martinec, and T. Pajdla, "A convenient multi-camera self-calibration for virtual environments," *PRESENCE: Teleoperators and Virtual Environments*, vol. 14, no. 4, pp. 407–422, 2005.

[62] T. Ueshiba and F. Tomita, "Plane-based calibration algorithm for multi-camera systems via factorization of homography matrices," in *ICCV*, pp. 966–973, 2003.

[63] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge: Cambridge University Press, 2004.

[64] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," *International Journal of Computer Vision*, vol. 80, pp. 189–210, 2008.

[65] C. Loop and Z. Zhang, "Computing rectifying homographies for stereo vision," in *CVPR*, vol. 1, pp. 125–131, 1999.

[66] A. Fusiello, E. Trucco, and A. Verri, "A compact algorithm for rectification of stereo pairs," *Machine Vision and Applications*, vol. 12, no. 1, pp. 16–22, 2000.

[67] K. Deng, L. Wang, Z. Lin, T. Feng, and Z. Deng, "Correction and rectification of light fields," *Computers & Graphics*, vol. 27, no. 2, pp. 169 – 177, 2003.

[68] N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "A novel rectification method for two-dimensional camera array by parallelizing locus of feature points," in *International Workshop on Advanced Image Technology (IWAIT2008)*, pp. B5–1, 2008.

[69] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *CVPR*, pp. 1106–1112, 1997.

[70] G.-Q. Wei and S. D. Ma, "Implicit and explicit camera calibration: Theory and experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 469–480, 1994.

[71] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly Journal of Applied Mathmatics*, vol. II, no. 2, pp. 164–168, 1944.

[72] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM Journal on Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[73] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, 2011.

[74] N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by gpu-accelerated large displacement optical flow," in *European Conference on Computer Vision (ECCV)*, pp. 438–451, 2010.

[75] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 1470–1477, 2003.

[76] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, 2004.

[77] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2161–2168, 2006.

[78] L. Shapiro and G. Stockman, *Computer Vision*. Prentice Hall, 2001.

[79] G. Mori, X. Ren, A. A. Efros, and J. Malik, "Recovering human body configurations: Combining segmentation and recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[80] C. O. Conaire, N. E. O'Connor, E. Cooke, and A. F. Smeaton, "Multispectral object segmentation and retrieval in surveillance video," in *IEEE International Conference on Image Processing (ICIP)*, pp. 2381–2384, 2006.

[81] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[82] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[83] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[84] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient nd image segmentation," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, 2006.

[85] D. Greig, B. Porteous, and A. H. Seheult, "Exact maximum a posteriori estimation for binary images," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 271–279, 1989.

[86] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 309–314, 2004.

[87] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3241–3248, 2010.

[88] A. Papazoglou and V. Ferrari, "Fast object segmentation in unconstrained video," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 1777–1784, 2013.

[89] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sánchez, "Symmetrical dense optical flow estimation with occlusions detection," *International Journal of Computer Vision*, vol. 75, no. 3, pp. 371–385, 2007.

[90] A. Ayvaci, M. Raptis, and S. Soatto, "Sparse occlusion detection with optical flow," *International Journal of Computer Vision*, vol. 97, no. 3, pp. 322–338, 2012.

[91] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.