Layer Extraction and Image Compositing using a Moving-aperture Lens

Anbumani Subramanian

Dissertation submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

A. Lynn Abbott, Chair
Peter M. Athanas
Amy E. Bell
Lamine M. Mili
Roger W. Ehrich

May 2, 2005
Blacksburg, Virginia

Keywords: Image compositing, video matting, matte extraction,
image layers, layer extraction.

Layer Extraction and Image Compositing using a Moving-aperture Lens

Anbumani Subramanian

(ABSTRACT)

Image layers are two-dimensional planes, each comprised of objects extracted from a two-dimensional (2D) image of a scene. Multiple image layers together make up a given 2D image, similar to the way a stack of transparent sheets with drawings together make up a scene in an animation. Extracting layers from 2D images continues to be a difficult task. Image compositing is the process of superimposing two or more image layers to create a new image which often appears real, although it was made from one or more images. This technique is commonly used to create special visual effects in movies, videos and television broadcast. In the widely used "blue screen" method of compositing, a video of a person in front of a blue screen is first taken. Then the image of the person is extracted from the video by subtracting the blue portion in the video, and this image is then superimposed on to another image of a different scene, like a weather map. In the resulting image, the person appears to be in front of a weather map, although the image was digitally created. This technique, although popular, imposes constraints on the object color and reflectance properties and severely restricts the scene setup. Therefore layer extraction and image compositing remains a challenge in the field of computer vision and graphics. In this research, a novel method of layer extraction and image compositing is conceived using a moving-aperture lens, and a prototype of the system is developed. In an image sequence captured with this lens attached to a standard camera, stationary objects in a scene appear to move. The apparent motion in images is created due to planar parallax between objects in a scene. The parallax information is exploited in this research to extract objects from an image of a scene, as layers, to perform image compositing. The developed technique relaxes constraints on object color, properties and requires no special components in a scene to perform compositing. Results from various indoor and outdoor stationary scenes, convincingly demonstrate the efficacy of the developed technique. The knowledge of some basic information about the camera parameters also enables passive range estimation. Other potential uses of this method include surveillance, autonomous vehicle navigation, video content manipulation and video compression.

Dedicated to

my appa (father)

and

the memory of my amma (mother)

Acknowledgments

I gratefully acknowledge the valuable guidance of my advisor Prof. A. Lynn Abbott, all through my research and graduate studies. His thoughtful analyses, insightful suggestions and seemingly simple, yet intellectually stimulating questions during our discussions have enormously helped me to improve my work.

I am indebted to two of my best friends whose support enabled me to reach this far in my education. The main source of my motivation was Saraswathi Padmanabhan, whose incessant encouragement lead me in to graduate school. Joaquim Fernandes, supported me numerous times and I would not have crossed some major hardships, without his timely financial help.

I acknowledge the Physics Department at Virginia Tech, for supporting my graduate studies over a long period and for offering an opportunity to teach the students, which I really enjoyed. Of special mention is Ms. Christa Thomas, whose concern and kind help was a great comfort during my assignments in the Physics department.

I also acknowledge the support and interest from Vision III Imaging, Herndon, VA towards this research. The upbeat and enthusiastic Chris Mayhew was very helpful, and appreciative of the progress made during my work.

I thank Prof. Amy E. Bell, for various timely help and concerned suggestions during my graduate studies and most importantly for introducing me to the wonderful world of LaTeX. I also thank Prof. Lamine M. Mili, for opening up the new realm of statistical robust methods to me, a topic which is applied in this work and which continues to fascinate me. I acknowledge Prof. Peter M. Athanas and Prof. Roger W. Ehrich for devoting their time in being part of my committee.

I recollect the many thought-provoking discussions with Lakshmi R. Iyer on the moving-aperture lens

during the earlier image segmentation work. I acknowledge her for those discussions, as they helped me years later, when modeling the lens characteristics in this work. I thank Dr. Erol Sarigul, for the informative conversations we had in our lab – they helped me to refine some of my developments in this work. I appreciate the help of Dr. Sang-Mook Lee, Xiaojin Gong, and Joel. A. Donahue, who gladly volunteered their time to be the subjects in the image capture experiments used in my research.

I must acknowledge many of my friends – Roshan Bangera, E.N. Gururajan, Harish, Rajesh Jagannathan, Anjani Anant, S. Balaji, Karthik Subramanian, Gopinath Rangan, and Ayub Khan for their timely guidance which helped me to pursue graduate studies and research.

I thank my sister for taking very good care of my parents, while I was occupied with my studies, miles away from home. I attribute the success and credit, from this and other work to my father and mother, who have made immense personal sacrifices towards my education and well being.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

An image of a scene is a two-dimensional representation of the three-dimensional real world. The real world objects in the scene are projected onto the image plane in a camera by the various lens elements. For some situations, the camera image of the scene can be considered as made from a stack of 2D planar images located at different distances in the scene, similar to viewing a stack of drawings on a glass plate, placed one over the other. Therefore these planar images are like 'layers' which together form the image on the camera. The process of determining the various 'layers' which make up an image is called *layer extraction.*

Image compositing is the process of superimposing or overlaying two or more image layers to create a new image, which often appears real, although it was created from multiple image layers. This technique is commonly used to create special visual effects in movies, and videos where highly imaginative, sometimes unrealistic scenes with artists performing complex actions are created. It is also used in commercial television broadcast where a news reader appears to read weather forecasts in front of a map while in reality the broadcast image is a composite of two images – the news reader and the map, from two different cameras in the studio.

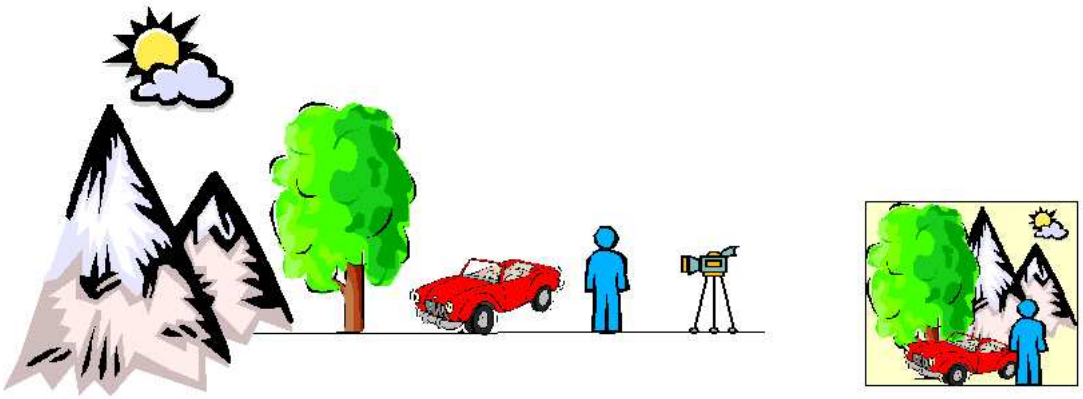
1.1 Image layers and compositing

Consider a simple, real world 3D scene as illustrated in Fig.1.1. This figure shows a scene with two inanimate objects and a person located at arbitrary distances, in front of a camera. The corresponding 2D image

captured by the camera in this setting is shown in Fig.1.1(b).

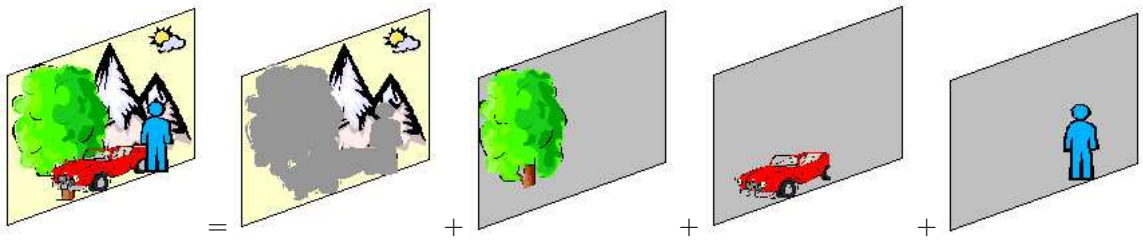
The concept of layers can be understood from Fig.1.2. The captured image from the example scene can be also thought of as being made of three different 2D image planes ("layers") shown in Fig.1.2(b)–(d). These three layers together comprise the original image. The layers shown in the illustration are extracted based on object distances in the scene. The two inanimate objects and the person are extracted in separate layers while the remaining regions in the camera image are extracted as a separate image layer.

This idea of image layers naturally allows manipulation of layers. Once the layers are extracted from an image as shown in Fig.1.2, a specific image layer can be chosen for overlaying on some random background image to create a new image. The resulting new image usually appears real, although it was artificially created. This idea is illustrated in Fig. 1.3. A more detailed description of image compositing and various existing techniques are given in Chapter 2.



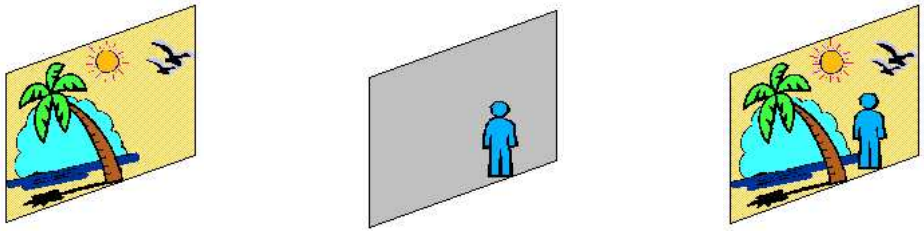
(a) (b)

Figure 1.1: Illustration of a real-world scene: (a) a camera in an example scene; (b) captured image of the scene.



(a) (b) (c) (d) (e)

Figure 1.2: Illustration of image layers concept: (a) 2D image captured, (b) – (e) component "layers" of the image.



(a) (b) (c)

Figure 1.3: Image compositing: (a) a random background image, (b) an image layer from Fig. 1.2, (c) composited result of overlaying the image layer on the new background.

1.2 Motivation

This research is concerned with the problem of layer extraction in 2D images. Using a pinhole camera model and optics, it is fairly straightforward to find the corresponding two-dimensional (2D) camera image from a three-dimensional (3D) scene or from a stack of image layers. But the inverse problem – given a 2D image, determining the 3D locations of objects in real world or its image layers – is underconstrained, due to insufficient information in the image. Therefore this problem is a difficult and challenging problem.

This research is primarily motivated by an earlier work on image segmentation using a moving-aperture lens [1], where parallax in an image sequence was utilized for segmentation. It offered an insight into the advantages of utilizing the moving-aperture lens and convinced of the feasibility of extracting layers in an image. Furthermore, the application of image compositing using the extracted layers led this research to develop a prototype of the compositing system.

This dissertation presents a methodology for layer extraction and image compositing – novel in its approach and advantageous in many aspects. The novelty lies in the utilization of a moving-aperture lens for image capture, which induces a constraint in the images and hence enables layer extraction. Some advantages of this method in compositing include multiple-object extraction, independence of object properties and relaxation of constrained scene settings.

1.3 Previous Work

Popular image editing software like Adobe Photoshop and GIMP allow a user to manipulate images and replace parts of an image with different content. These software packages use an interactive procedure based on techniques like Intelligent Scissors [2, 3], Magic Wand (by Adobe) and Knockout2 (plugin from Corel for Adobe Photoshop) to easily extract objects in an image. The idea of image layers is also used in these software packages to allow a user to arrange the order of synthetic image layers while creating new images. These software packages only allow the manipulation of individual images, and do not work on a sequence of images.

1.3.1 Layer extraction

In the field of computer vision, the idea of layers was pioneered by Wang and Adelson [4]. They formulated the layer extraction problem as a superposition of multiple, overlapping layers and this work sparked additional interest in the community due to its applications in image compression. Their approach extracted layers using affine motion analysis and k-means clustering of image flow vectors. In their sample video sequence, a camera translates parallel to the scene and so a relatively large motion is induced in the image sequence. This large motion information was exploited to extract layers in the image.

Baker et al. [5] introduced a framework for stereo reconstruction, exploiting the idea that each layer implicitly lies on a fixed plane in the 3D world to represent a scene as a set of planar layers. Each layer consists of a 3D plane equation, a colored opacity map (*sprite*), and a depth-offset map relative to the plane. This method requires a manual initialization of layers to estimate the parameters of the plane for each layer and a re-synthesis step refines the initial layer input.

Torr et al. [6] applied a Bayesian approach to segment a scene into layers. The number of layers are automatically determined from an Expectation-Conditional Maximization (ECM) algorithm, initialized from high-confidence areas in the image, where disparities are easy to estimate. The big problem in this approach is that layer segmentation is entirely based on the first image in the sequence and so the initialization determines the result.

Ke and Kanade [7] identified the major issues in layer extraction from images as follows: 1) determining the number of layers, 2) estimating spatial properties of each layer (such as motion or pose), and 3) the

assignment of individual pixels to layers. They formulated layer extraction as a clustering problem in a low dimensional subspace computed using a robust statistical method. The subspace constraints are derived from relative affine transformations in multiple local regions in a sequence of images and a mean shift based clustering is utilized to extract planar layers in real image sequences. As this approach is modeled layers with 2D parametric motion, it is not applicable to segment scenes with non-rigid or articulate motion and so will result in a multiple layer segmentation for scenes with human body.

Tao et al. [8] modeled a multi-object tracking problem as a 2D layer estimation problem to enable dynamic-layer representations. They apply constraints on motion, appearance and segmentation across multiple images and use a global parametric shape prior for segmentation. An Expectation-Maximization (EM) algorithm in a maximum a posteriori (MAP) estimation framework is used for layer tracking, assuming the number of layers in a scene and an initial layer representation are given. This approach is shown to work in tracking vehicles in airborne videos where background and objects compete using motion videos for layer estimation.

Wexler et al. [9] used constraint priors in a Bayesian framework to extract layers from multiple images. This method works without the knowledge of background by relying on sufficiently sampled background. More recently, Xiao and Shah [10] used a graph-cut approach to extract motion layers in presence of occlusion.

1.3.2 Compositing

Image compositing has been used in film production for many decades, since early 1920s. It was used in films like *King Kong* (1933), *Ben-Hur* (1959), and *Star Wars* (1977) [11]. It has evolved to become an essential component in present day film production. Its early use was based on the concept of 'cut-outs (or 'mattes') in an optical printing process. Later, with the evolution of digital devices, the concept was ported to the digital domain. The commonly used system for compositing in the film industry is the 'blue-screen' technique ('chroma-key' in the broadcast industry). This was created by Peter Vlahos, who later founded the company Ultimatte. Vlahos was recognized with a Scientific or Technical Oscar award for "the conception and perfection techniques for Color Traveling Matte Composite Cinematograph" in 1964. A detailed description of existing image compositing methods is given in Chapter 2.

The earliest work on image compositing in the field of computer vision is by Porter and Duff [12]. This work laid out in simple mathematical terms the process of image compositing using an 'over' operator. In

a later work, Blinn [13, 14] neatly summarized the various possible conditions for the over operator that arise in image compositing. The repeated addition of foregrounds to an image is called *2-1/2 D rendering* or *painter's algorithm,* as is explained in Chapter 2.

Mitsunaga et al. [15] developed a human-assisted matte (key) extraction system with a three step approach: boundary estimation of foreground object, computation of gradient along the boundary and construction of matte surface.

Smith and Blinn [16] presented a mathematical approach to constant-color matting based on the expired patent by Vlahos [17]. In this approach, two images of the foreground object are taken with two different background colors. Although the background colors change in the two images, the foreground object remains the same in the two images. A simple image difference operation can extract the foreground object and later it can be composited on a new image. This method is not suitable for live action scenes, where an object cannot be expected to perfectly repeat a sequence of action, after changing the background in the scene.

Brostow and Essa [18] demonstrated that using a single camera, the motion of objects in a static scene can be used to estimate the occlusion in the scene. This occlusion information can then be used to decompose a video sequence into different layers based on the spatial information obtained from edge detections. The method relies entirely on the object motion and occlusion of objects in the scene. Therefore it does not work in a stationary scene or when there is motion but no occlusion in a scene.

To overcome the limitations in the popular blue screen method, novel techniques have also been attempted. Ben-Ezra [19] used an invisible keying signal in which the foreground object is imaged using a polarized light source. The camera used a polarization beam splitter to produce two images – an in-phase and an out-of-phase image from the single incident image. The center of projection of the two images are same owing to the use of beam splitter and so a difference between the two images can produce a matte for use in compositing.

Color mixture models [20] is another method attempted to replace the blue screen method. In this technique, panoramic images of a scene are captured using a fixed camera and the foreground is segmented from the panoramic scene using Gaussian mixture models in an Expectation-Maximization (EM) procedure to model the various colors in the object. However this method is restricted to panoramic images with horizontal camera panning and cannot handle multiple foreground objects.

Chuang et al. [21, 22, 23] developed a Bayesian approach to video matting and obtained good composited

results. In this work, a user-specified foreground mask (*garbage matte*) in a few important frames are taken as an initialization to estimate the background. Then bidirectional optical flow in image sequences is used for background estimation along with user specified *trimaps*: a segmentation of the scene into three regions namely "definitely foreground", "definitely background", and "unknown". The trimaps are used to extract the matte of a scene. The trimaps are calculated for every frame using hand-drawn trimaps in selected key-frames. This method has been shown to work for stationary scenes and also with images involving smoke in a scene. Some problems in this approach are requirement of user-specified trimaps, dependence of key-frames on sequence and requirement for more key-frames in complicated scenes.

Recently, wavelets have also been used to generate the matte of a scene (*environment matte*), which also contains the reflection and refraction effects of the backdrop through the scene [24]. In this method, many 2D wavelets are emitted as illumination patterns in a scene by a regular CRT monitor and images of the scene are captured by a camera. An approximation of light transport from the scene into the camera, through a feedback of the emitted patterns, is used to extract the matte of the scene. Although this method is shown to handle diffuse and transparent surfaces, the setup is complicated with the need for high dynamic range photographs, large storage requirement for each matte (on an average 2.5 GB, for each matte) and is extremely time consuming (12 hour time limits were used).

Apostoloff and Fitzgibbon [25] presented a Bayesian video matting technique using statistics of natural images. Priors learned from spatiotemporal gradients in training image sequences are used with a learned foreground color model to get a matte of the scene using a regularization approach. This approach is heavily dependent on building a good prior, which is obtained from several blue-screen sequences with reliable mattes.

The problem of image compositing can also be considered in relation to the MPEG-4 standards [26]. The MPEG-4 standard is concerned with the multiplexing and decoding of audiovisual data. At the coder end, it allows various, independent audio and video objects to be multiplexed to produce a media stream. The various objects in the video called '*video objects*' (VO) can be arbitrarily shaped (unlike rectangular shapes in MPEG-1), each with its own semantics. These video objects are coded on a video object plane (VOP). This process of video coding is somewhat similar to the image compositing technique.

1.4 Problem Statement

This research is concerned with the extraction of 'image layers' for a stationary scene, and then superimposing an image layer from the scene onto an arbitrary background image. The goal is to develop a new, automatic image compositing system.

Consider a scene in a 3D world XYZ (as in Fig. 1.1), with the origin of a reference coordinate system centered at the camera and a 2D image coordinate system in xy. Let various stationary objects in the scene be located at different (parallel) distance planes $Z_1, Z_2, Z_3, ...$ from the camera. As shown in Fig. 1.2, the 2D image of the scene formed in the camera $I(x, y)$ can be thought of as a combination of the *planar images (layers)* of objects at different distances in the scene, $f(X, Y, Z_k)$, $k = 1, 2, 3, ...$ A planar image $f(X, Y, Z_k)$ is formed by some function of the distance of an object from the camera, interaction of light in the scene, optics of the camera and many other factors. An image of the scene is considered to be made of multiple $(L-1)$ planar images and a layer comprising the background objects. For a scene made from L layers (sets), the 2D camera image can be approximated as combination (superset) of the layers located at distances $\{Z_k | \, k = 1, 2, 3, ..., L\}$ from the camera and therefore can be written as

$$I(x,\, y) = \bigcup_{k=1}^{L} f(X,\, Y,\, Z_k)$$

The problem of layer extraction is to determine the planar layers $f(X, Y, Z_k)$, given the image $I(x,\, y)$. As mentioned earlier, this is an underconstrained problem and therefore some additional information or constraint must be used to solve the problem.

In this research, a camera whose aperture moves on its two-dimensional (2D) aperture plane is utilized to address the layer extraction problem. As the aperture of the lens moves, the camera captures images of the scene at different positions of the aperture. This *moving-aperture* induces parallax in the images, whose magnitude is proportional to the location of an object in the scene, from the plane of focus. The parallax information inherently present in the image sequence is highly valuable, and is exploited in this research to extract the layers in an image. The extracted layers can then be used to composite a layer with an arbitrary background image.

The research problem can be summarized as follows: using a moving-aperture lens, identify the various layers planes in an image, arrange these layers by the distance of the corresponding planes in the scene from

the camera, and then superimpose a layer from the image onto an arbitrary background image. The analysis is limited to stationary scenes imaged using a stationary camera with a moving-aperture lens.

1.5 Significance of this Research

This research presents a novel approach for layer extraction that differs from the methods described above in several ways. It is not dependent on a constant color background, and does not depend on color or reflection of objects in the scene. Whereas most previous systems rely on a substantial level of motion, this system utilizes a novel lens in which the aperture moves by a small amount off the nominal optical axis under motor control. This moving, off-center aperture introduces a small image flow field that facilitates layer extraction for stationary scenes, even though the camera itself can remain stationary.

An alternative approach of using inexpensive, multiple cameras in a stereo configuration is often cited, is not a valid argument as it ignores some necessary details. An intended application of this research is image compositing in the film industry, where the cost of cameras is exorbitantly high. Therefore the use of multi-camera stereo for layer extraction is prohibitively expensive. Using only a camera with a moving-aperture lens in this technique is highly cost effective.The significance of this work is that it performs layer extraction, with no need for additional requirements (like a blue screen or extra lighting) in the scene and uses only one camera to perform compositing.

The algorithms involved are suited for implementation on specialized hardware or as plugin modules for existing compositing systems and hence the viability of an implementation. While most existing methods for image compositing use controlled scene settings and are limited by the object properties, this method poses no such constraints. In addition, this method has significant advantages of a possible reduction in cost and the ease of a portable setup.

Also unlike most previous systems, this approach automatically determines the number of layers, and can identify layers as located in focal or non-focal planes. With no manual input necessary for the system, it typically takes less than 10 minutes to extract a stationary matte for a stationary scene.

1.6 Contributions of this Research

The following contributions are made to the field of computer vision from this research:

1. a new framework for automatic extraction of image layers using a moving-aperture lens is developed,
2. a prototype of a novel image compositing system is developed for stationary scenes,
3. a mathematical model of the moving-aperture lens is formulated and verified using experimental data,
4. a new method of passive range estimation using the moving-aperture lens is developed and
5. new approaches to circle and ellipse fitting using robust statistical methods are proposed.

The motivation for this research is based on an earlier work [1], which

1. developed a novel method of image segmentation using a moving-aperture lens, and
2. demonstrated the feasibility of range estimation using the same lens.

1.7 Overview of the Chapters

Chapter 2 explains some basic concepts in image analysis. The earlier work on image segmentation using a moving-aperture lens including its results are presented in Chapter 3. Chapter 4 presents the new methodology developed for layer extraction and image compositing using the moving-aperture lens. It also discusses the mathematical model of the moving-aperture lens. An introduction to robust statistical methods and a discussion on the new robust circle fitting method is given in Chapter 5. The related study on robust ellipse fitting is presented in Chapter 6. The results from the developed layer extraction and image compositing methodology along with the passive range estimation method are given in Chapter 7. The final Chapter concludes the dissertation, discussing the applications of this research and outlining some possible future directions.

Chapter 2

Image Analysis - A Review

This chapter presents a review of some topics in image analysis that are helpful and necessary to follow the later discussions on the developed layer extraction and image compositing method. Some good references for a detailed discussion on these topics in computer vision are [27, 28, 29].

2.1 Imaging Geometry

In image analysis, it is often convenient to represent the three-dimensional (3D) world using a camera-centered coordinate system in X, Y and Z. In this convention, width is represented along the X-axis, height along the Y-axis and distance (or depth) along the positive going Z-axis, away from the viewer. Thus, if an object is in view (located in front of the observer), then its Z value of the location is positive. The farther the object is from the viewer, the higher is the value of Z and when the object may be located behind the observer, Z is a negative value.

The illustration in Fig. 2.1 shows a left-handed, 3D coordinate system by the X, Y and Z axes, whose origin O can represent either the aperture center of a camera or the iris of a human eye. Similarly the two-dimensional (2D) image plane can either represent the film in a camera or the retina of a human eye. The image plane coordinate system is represented by x and y axes. Thus, a point P in the three-dimensional system is represented as (X, Y, Z) and a point on the image plane is represented as (x, y). The image plane is at a distance f from the aperture, called the *focal length* and in most vision systems, the focal length can

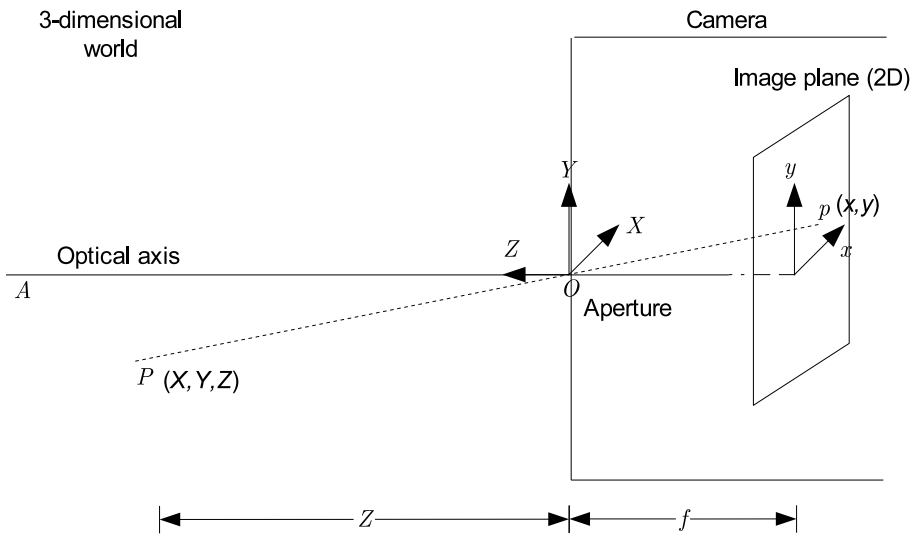


Figure 2.1: Image plane and coordinate geometry.

be adjusted to get a clear, sharp view of an object in a scene, on the image plane. In this imaging geometry, each point in the 3D scene (X, Y, Z) is mapped to a point (x, y) on the 2D plane. Using the relation of similar triangles from Fig. 2.1, we can write

$$\frac{x}{X} = \frac{y}{Y} = \frac{f}{Z} \tag{2.1}$$

$$\Rightarrow \qquad x = \frac{f}{Z}X \qquad \text{and} \qquad y = \frac{f}{Z}Y \tag{2.2}$$

This mapping of the 3D world onto the 2D image plane in 2.2 is a many-to-one relation. But the reverse mapping of a 2D point on an image plane to a unique point in the 3D world is one-to-many relation. Therefore a unique solution is not possible without additional constraints.

2.2 Image Segmentation

Image segmentation is the process of partitioning an image into meaningful groups (or regions). Despite this seemingly simple definition, segmentation is a challenging task. A significant amount of work has been

(a) (b) (c)

Figure 2.2: Example images where various cues such as edges, texture and color are used by the human visual system for segmentation.[2]

devoted to image segmentation in the computer vision and image processing areas.

The human visual system utilizes many cues such as color, shape, shade and reflectivity in an image to identify different objects. For example, consider the images in Fig. 2.2. In Fig. 2.2(a), the observer is likely to identify the edges of a roof using the different grayscale (or intensity) information while in Fig. 2.2(b) the dominant cue used is probably texture. Figure 2.2(c) is a good example where the human eye excels in identifying the various jelly beans perhaps using grayscale and reflectivity cues.

Segmentation has been studied in great detail for at least three decades and so numerous techniques have been developed, many of them specific to certain applications. These techniques have been used in computer vision, machine vision, and biomedical applications for a long time. The expectations of the results vary widely between applications. In most vision related applications, a coarse approximation of object boundaries is usually sufficient to distinguish objects, whereas in biomedical applications, every pixel represents some valuable information and so any misclassification is considered detrimental. Therefore, no one segmentation method can be recommended for all applications. A good survey of image segmentation techniques is given in [31, 32, 33].

In general, segmentation methods can be classified into three categories based on the approach used:

- edge-based methods,
- region-based methods and
- model-based methods.

The first category relies on edge information in images, while the region based methods often use intensity

in a chosen local region for segmentation. The third category employs higher level models to approximate a shape or object in an image using edge based or region-based methods or both for segmentation. Apart from the approach used, segmentation methods can also be classified into two types, namely (i) supervised and (ii) unsupervised methods. In the former category, the number of regions expected in the image is specified and the segmentation procedure groups various portions of the given image into one of the possible regions. In the unsupervised category, the segmentation procedure itself will determine the number of possible regions in the image and then will segment areas of the image into one of these regions. The latter method is more challenging, because the number of regions in an image is itself difficult to determine and so segmentation becomes a two-fold problem. In supervised methods, the user or a semi-automatic procedure initializes the regions, and then the segmentation procedure attempts to refine the initial input.

Motion is also an important clue widely used in image segmentation. Consider an image sequence in which an object moves in a scene. The apparent change in location of the object between successive frames in the sequence can be utilized to determine boundaries of the object from its surroundings. In this case, even simple operations such as a difference of the images can help to estimate the boundaries. If the surface of the object is uniform and the apparent shift is small, then image differencing can give a good estimate of the boundaries and this can assist in segmentation. However, if the shift is large, the segmentation is more difficult.

2.3 Stereo

A standard camera records a stationary scene (3D world) seen as a 2D image. An image sequence of this scene adds another dimension - time, but still no distance information is included. Therefore depth of a scene is not perceived when observing a sequence of 2D images. In contrast, the human visual system is able to approximately perceive relative locations of objects in a scene in many situations.

Two images of a scene taken by a small distance offset comprise a *stereo* pair. Stereo images are commonly used to enable the perception of 3D using special glasses. Stereo is a well known topic of research in computer vision.

A general approach in stereo is to use two identical cameras held in a rigid configuration. Stereo obtained using two cameras is called *binocular stereo.* An inexpensive alternative is to use a single camera, but move it carefully between different positions to capture images of a scene. At the other extreme, it is possible to

use more than two cameras. This approach is called *N-camera* or *multiview stereo*. Some good references on this topic are [34, 35].

2.3.1 Epipolar Geometry

Consider two cameras at arbitrary orientations in a scene and separated by a small distance. It is often convenient to consider an image plane (imaginary) at a distance of f in front of the camera center instead of an image plane at a distance of f behind the camera center (represented by O in Fig. 2.1).

For the two cameras in our discussion, their corresponding image planes, imaginary in nature, can be considered at a distance f from the respective camera centers. Fig. 2.3 shows an illustration of this setup in an arbitrary orientation. Let P be a point in the 3D scene which is projected onto point p in the left image. Now the same point P appears at a different location p' in the right image, the offset distance proportional to the distance of camera separation. The image points p and p' correspond to P in the scene and establishing this relation is called the *correspondence problem*.

The 2D plane given by the two camera centers C and C', the scene point P is called the *epipolar plane*. The line formed on an image plane by intersection with an epipolar plane is called the *epipolar line*. A careful analysis of image geometry in stereo reveals that the point p' always lies on the corresponding epipolar line in the right image. This information is extremely useful as the knowledge of epipolar lines significantly narrows the search area required to establish correspondences.

Correspondences between two images are typically established by a search for best match of a pixel (or image region) in one image, near a neighborhood of its expected location in the other image. This is a cumbersome process and is a computationally expensive task. However, by knowing the epipolar line in an image, the correspondence problem is reduced to a one-dimensional search for the best match instead of a wider neighborhood search.

2.3.2 Range Estimation

A simple model of stereo is binocular stereo, in which two cameras are separated by a fixed distance (*baseline*) with an assumption that most of the image content is very similar. Consider an example with two cameras positioned almost parallel and separated by some distance in a scene with one object, as shown in Fig. 2.4. Here the image planes are considered coplanar. Due to the baseline separation b, point p appears at

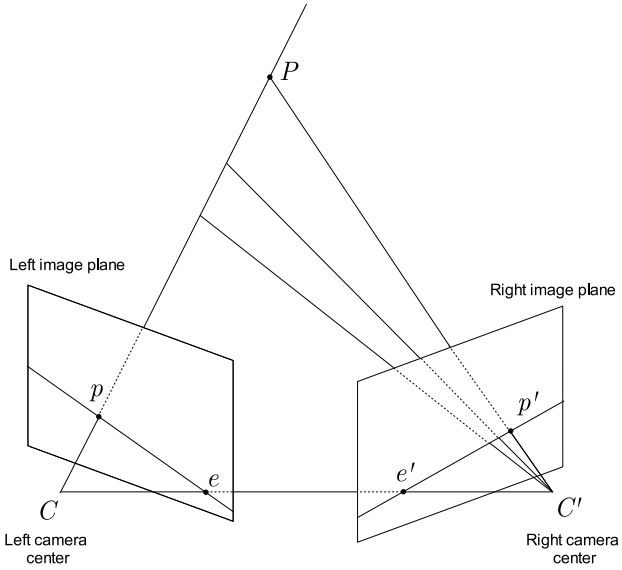


Figure 2.3: Stereo and epipolar geometry in an arbitrary camera configuration.

different image locations, x and x' in left and right images respectively as shown in the figure. The magnitude difference in these image distances or the image offset is known as the *disparity* d. This disparity is directly proportional to the baseline distance as given by the relation

$$d = \frac{b\,f}{Z} \tag{2.3}$$

where f is the focal length of the lens and Z is the distance of the object from the camera. The image disparity can be calculated by analyzing the images.

If the baseline separation and focal lengths of the lenses are known, then it is possible to calculate the distance of the object using (2.3). A more detailed description of range estimation techniques can be found in [36, 37, 38].

2.3.3 Stereo Configurations

The possibilities of camera configurations in stereo are limitless. But the big challenge arises in image analysis, when the correspondence problem is to be solved.

Consider the simple case of two similar cameras in a stereo setup – parallel to each other, separated by some baseline distance, and observing an object in a scene as shown in Fig. 2.5(a). The images captured by the cameras will appear as given in Fig. 2.5(b). An image of the object appearing in the two images will only

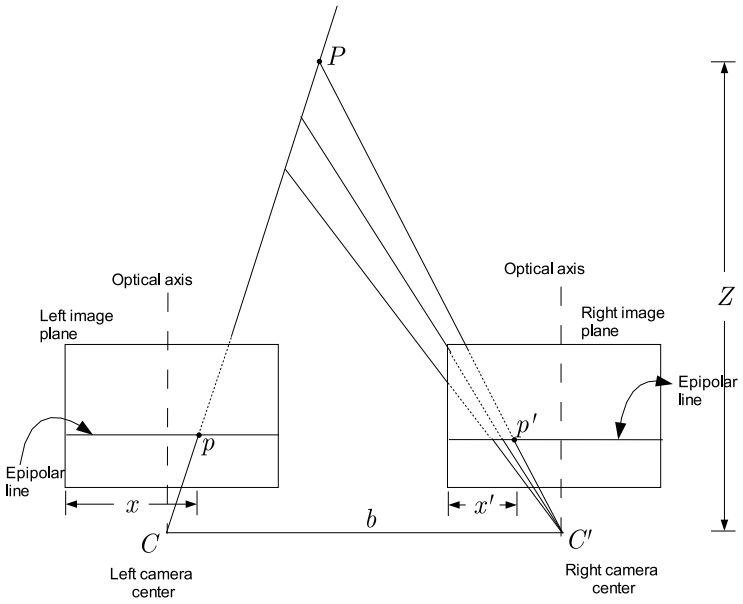


Figure 2.4: Stereo geometry with coplanar image planes.

be displaced by some horizontal distance. Because the cameras are separated by a horizontal baseline, the epipolar lines corresponding to this configuration are also horizontal in the images. Therefore, for a point p on the left image, its corresponding image point on the right image p' will appear on the horizontal epipolar line in the image. By searching for an image match along this epipolar line, the correspondence of $p - p'$ can be established.

Now consider another simple stereo configuration, with cameras displaced by a baseline at an angle θ from the horizontal X axis as shown in Fig. 2.6(a). In this case, the camera images will appear as in Fig. 2.6(b), with disparity corresponding to the object in an angular direction. The epipolar line in this stereo configuration will also be at an angle θ to the cameras. As earlier, the correspondence $p - p'$ can be identified by searching along the angular epipolar line.

A good review of topics in stereo, its problems and challenges can be found in [39, 40, 41] and in books such as [42, 34, 35].

2.4 Optical Flow

Inspired by the idea of fluid flow, Horn and Schunk applied a similar idea of flow in image sequences. They called this technique *optical flow* (or *image flow*) and laid the foundations of a new area of research. Since

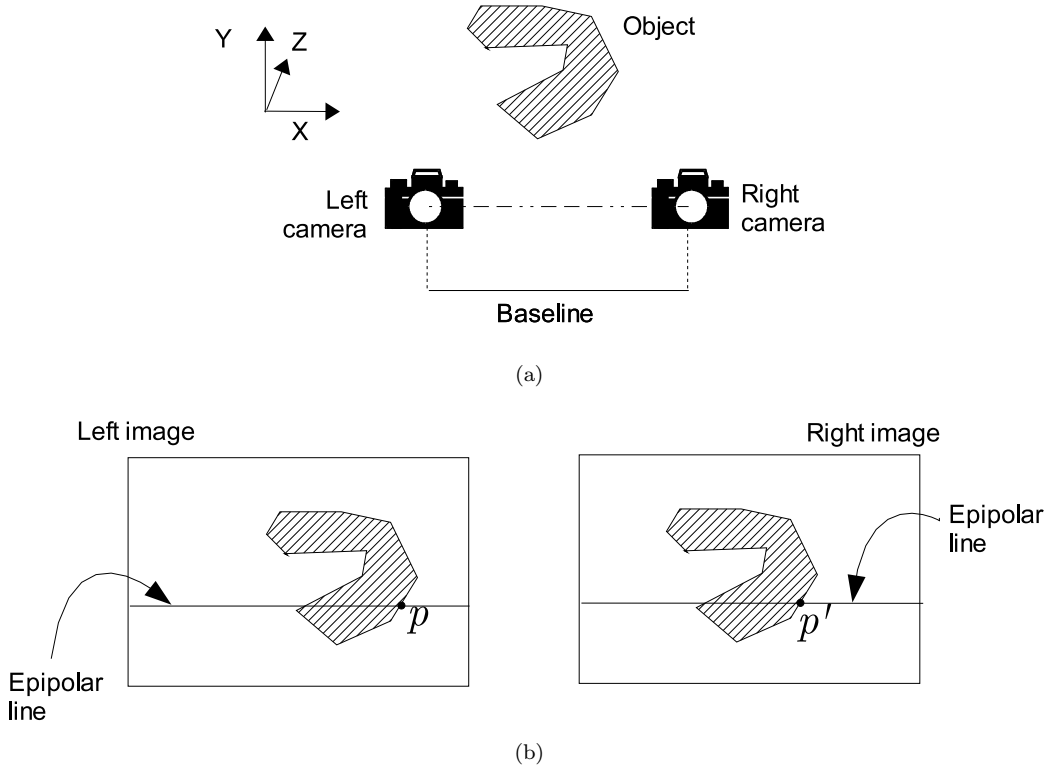


Figure 2.5: Simple binocular stereo: (a) camera configuration in a scene, (b) images from two cameras.

their first publication [43], various vision applications have exploited this idea, many of which are used in image segmentation.

Consider a sequence of images taken over time, where objects in the images appear to move. This object motion apparent in images can be thought of as a "flow" of the object in the image sequence. Let I represent a 2D image of a scene and $I(x, y, t_1)$ be the intensity of a point (x, y) in the image at time t_1. Let $I(x, y, t_2)$ be the intensity at the same position in another image of the scene, at time t_2. With the assumption that the image intensity at a position (x, y) remains constant in small interval of time, the following relation can be written:

$$\frac{d}{dt}I(x, y, t) = 0 \tag{2.4}$$

Expanding the above equation for the entire image gives

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \tag{2.5}$$

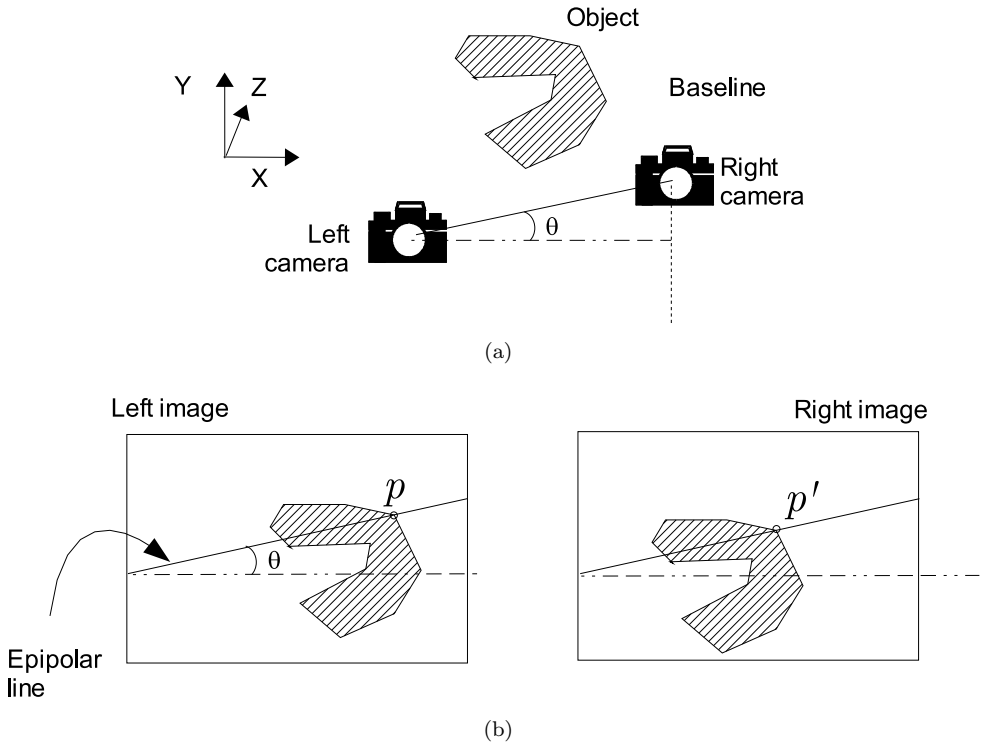


Figure 2.6: Simple binocular stereo with an inclined baseline: (a) cameras displaced by an arbitrary angle in a scene, (b) images from two cameras showing object displacement with same angle of inclination in disparity.

which can be simplified as

$$I_x\, v_x + I_y\, v_y + I_t = 0 \tag{2.6}$$

by substituting

$$I_x = \frac{\partial I}{\partial x} \qquad I_y = \frac{\partial I}{\partial y} \qquad v_x = \frac{\partial x}{\partial t} \qquad v_y = \frac{\partial y}{\partial t} \qquad I_t = \frac{\partial I}{\partial t} \tag{2.7}$$

Equation (2.6) is the well-known *optical flow constraint* [27], where I_x and I_y represent the image flow components along the x and y directions, respectively, for each pixel in the image. If an object appears to translate horizontally in an image sequence, then only the I_x component is present (with $I_y = 0$). Similarly, when an object translates vertically in an image, then only the I_y component is present (or non-zero) in the sequence. This approach of using differentiation information to estimate optical flow is known as the *gradient-based* method [44].

Another popular approach to calculate optical flow is the *region-based* or *area based matching* method [44]. In this, a small rectangular region of size $m \times n$ in the neighborhood of an image point is used to find

the optical flow. The local region (*template* or *reference window)* is matched with several possible image windows of same size about the image point. The match is evaluated using a quantitative measure and the position which yields the best measure is taken as the next position of the original point in the image sequence. The difference in two positions give the optical flow at the image point. Let $I_r(x, y)$ be a reference image window about the point (x, y) in image I_1. Let $I_w(x + \Delta x,\ y + \Delta y)$ be an image window of the same size in the image I_2, offset by some distance $(\Delta x,\ \Delta y)$. One of the widely used matching measure is the *normalized cross-correlation*, which is defined as

$$\rho = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} I_r(x, y)\ I_w(x + \Delta x,\ y + \Delta y)}{\sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} [I_r(x, y)]^2 \quad \sum_{i=1}^{m} \sum_{j=1}^{n} [I_w(x + \Delta x,\ y + \Delta y)]^2}} \tag{2.8}$$

where $-1 \leq \rho \leq 1$. The value of $|\rho| = 1$ indicates a best match, while $\rho = 0$ indicates a poor match between the two windows.

The research on optical flow dates back at least two decades and continues to be an active area. Some good references on optical flow include [45, 46, 47, 48, 44, 49].

2.5 Image Layers

An animation (cel) artist, arranges a stack of transparent, celluloid ('cel') sheets with drawings, one over the other to create a desired scene in an animation or cartoon sequence. Various characters and objects in a scene drawn on separate sheets are placed under an animation camera to capture the frame of the animation. By independently moving or changing the stacks, various actions or motion of characters in the animation are created. This idea of using stacks of drawings is very well known and was used by visual artists for many decades, before computer animation became popular.

Layers in images can also be understood from the same principle. The concept of a stack of drawings overlaid to create a visual scene can be reversed to consider an image made of multiple image layers, with objects of the image being components in one of the multiple layers. Therefore given an image of a scene, it is possible in principle, to decompose the image into many layers and extract the objects of the scene in different layers (*layer extraction*). Popular image editing programs like Adobe PhotoShop, GIMP, etc. allow a user to create synthetic image 'layers' and manipulate them independently.

The concept of layers in images was introduced in computer vision by Wang and Adelson, who formulated the problem in mathematical terms [4]. They defined the problem of extracting layers from images, where

layers are ordered by depths and slide one over the other, and combine obeying the rules of transparency and occlusion. In this work, each image layer consisted of three property maps – an intensity map, an alpha map and a velocity map. An intensity map represents the intensity information of objects in the image, an alpha map defined the opacity (or transparency) of a layer, and a velocity map described the motion or warping of a layer over time. In addition to these property maps, a layer representation can contain more maps such as – bump map, depth (Z) map, and delta map [50]. A bump map is used to encode the surface normal of a layer, a depth map encodes the Z-coordinate, and the delta map describes the rate of change of intensity in a layer.

The layers are usually extracted using motion analysis and ordered based on distance (or depth). In the MPEG *Flower Garden* video sequence used in [4], a camera translated in the scene creating a large image motion. This motion produced varying magnitudes of optical flow in the image sequence, for each object in the scene. Because the magnitudes were inversely proportional to the object distance from the camera, an analysis of optical flow was utilized to segment the image sequence into its component layers.

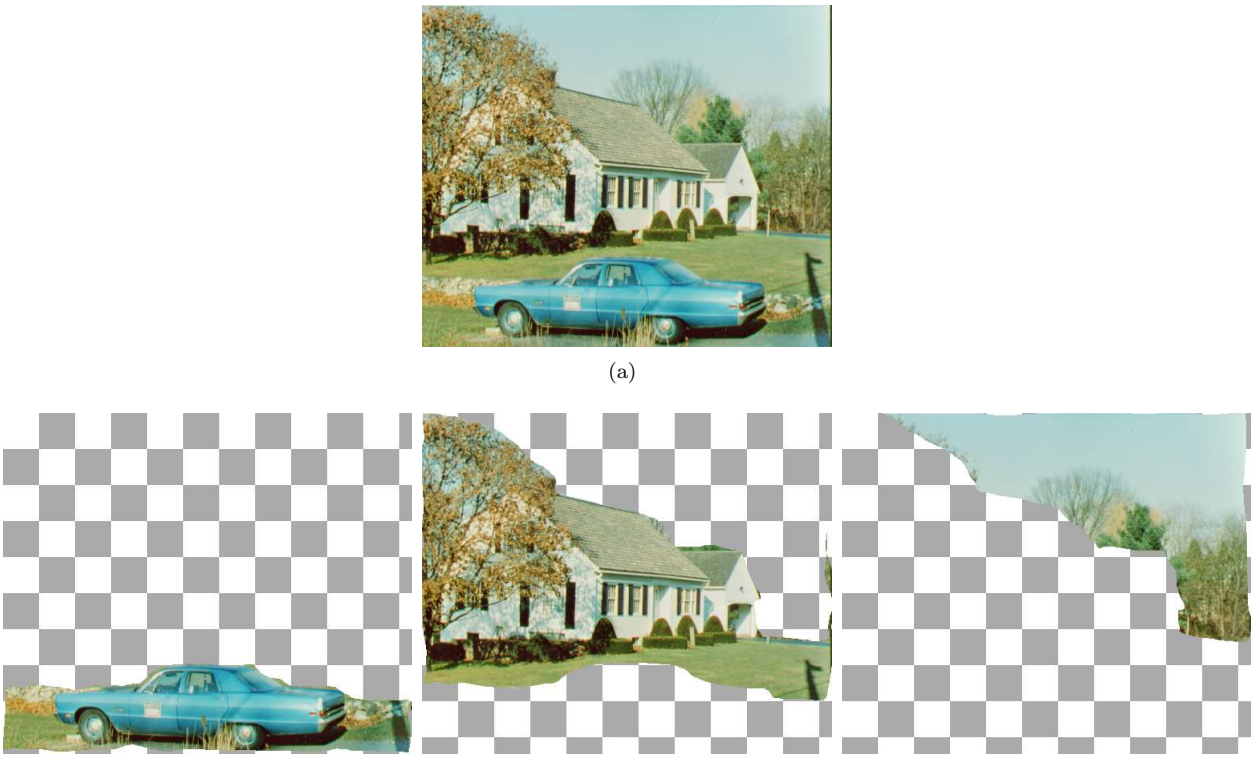
Layers are extracted depending on the segmentation procedure and are typically arranged by depth. There can be many possible layer representations for an image. For example, in [5], each layer consisted of a 3D plane equation, a colored opacity map (*sprite*) and a dense, depth-offset map relative to a nominal plane considered in the scene. Therefore layer representations and so a solution to the layer extraction problem is generally non-unique.

2.6 Compositing

Image compositing can be considered as a linear, mathematical combination of two images. Porter and Duff [12] proposed this as a foreground image $I_f$superimposed or laid *over* a background image I_b and hence called image compositing an 'over' operation (denoted by the *over* operator, $\backslash$). The mathematical relation can be written as

$$\begin{aligned} I_{comp} &= I_f \backslash I_b \\ \rightarrow I_{comp} &= (1-\alpha)\, I_b + \alpha\, I_F \end{aligned} \tag{2.9}$$

where I_{comp} represents the resulting composited image, α is the *opacity map* of the same size as the image. As α represents the opacity map, the value $(1-\alpha)$ therefore corresponds to the *transparency map*. The



(a) (b) (c) (d)

Figure 2.7: Image layers: (a) an image from [30], (b)-(d) possible layers in the image in (a). The layers shown here were manually extracted for illustration.

values in an opacity map can range from 0 to 1. An α value of 1 at a location in the image implies that the foreground image (intensity or color value) is opaque in that location and so the background image (beneath the foreground image) is not seen at that location. But an α value of 0 at an image location indicates transparency and so the background image is visible at that location ('transparent' or 'see-through' effect). An intermediate value of α indicates a visibility which is combination of the two images.

Equation (2.9) is illustrated in the example shown in Fig. 2.8, which shows an arbitrary image chosen and the result of compositing a layer from Fig. 2.7 with the chosen image. Following the notation in 2.9, I_f is the layer with the car (α value is zero, wherever the checker pattern appears), I_b is the chosen background shown in Fig. 2.8(a) and the resulting composited image is shown in Fig. 2.8(b).

The relation in (2.9) helps to understand compositing as an image overlaying operation. The compositing operation can be applied continuously with the composited result I_{comp} as a new background and another

(a) (b)

Figure 2.8: Illustration of image compositing. (a) Chosen background image [30], (b) a layer from Fig. 2.7 composited on (a).

foreground image for the next step [13]. In mathematical terms, this is equivalent to

$$I_{comp2} = (\,1 - \alpha_2\,)\, I_{comp} + \alpha_2\, I_{f2} \tag{2.10}$$

where I_{comp2} is the new composited result, I_{f2} is a new foreground and α_2 represents its opacity value and I_{comp} is the composited result from the previous step.

The compositing operation has an interesting property of associativity [13]. Extending the discussion with two foreground images I_f and I_{f2}, this property implies that the the final composited result I_{comp2} remains the same even when the sequence of operations change. Then the associativity property holds true:

$$I_{comp2} = I_{f2} \backslash (I_f \backslash I_b) = (I_{f2} \backslash I_f) \backslash I_B \tag{2.11}$$

2.6.1 Mattes

The opacity map is commonly known as the *alpha layer* or *key* or *matte* (hence image compositing is also called "matte extraction" problem). An easy way to understand a matte is to imagine a 'mask' held in front of a camera such that the image is captured only in the areas where the mask has a hole or allows light to enter the camera. The inverse of a matte is the transparency map, also known as a *counter-matte.*

Image compositing is commonly done using luminance or chrominance information in an image. In each technique, the important step is to extract a matte. Once a matte corresponding to a foreground image is extracted, then its transparency map can be obtained to perform image compositing.

The matte extracted for a stationary scene remains static and is called a *static matte.* In a scene with moving objects or moving camera, the matte needs to follow the motion of the object in a scene and this type of matte is called a *traveling matte.* In early days of film production, the static mattes were used either externally, in front of a camera lens or internally, behind a lens. The traveling mattes were always physical or photographic mattes. These were extracted using various methods – contrast-difference, color-difference, motion-control, self-matting and multifilm [51].

In the luminance (contrast-difference) based compositing method, the matte is extracted based on the difference in luminance (intensity) information in an image. The matte (luma key) extracted from this method will not be of high quality. So this method is used to extract the matte, mostly from scenes with text or graphic contents where the intensity variation between text and non-text areas of an image is often very good.

In chrominance (color-difference) based compositing technique, a matte (chroma key) is extracted using the difference in chrominance of an image. The chrominance corresponds to the color information and so allows more freedom in matte extraction than a luminance based method. A commonly used chroma-key method is the *"blue screen"* process.

A related technique is widely in television broadcasts of the (American) football games [52]. In this, a yellow first-down line is drawn on the grass of the stadium to show the television viewers how far the ball was carred in the game. This technology is based on an advanced color keying method, in which colors are represented in various formats. Before the game begins, the entire stadium is surveyed using a laser system and each camera to be used in the broadcast is calibrated. The various cameras at different locations are connected to a central computer. Depending on the available light, colors in the field, and colors in the uniforms of players, a human operator selects a suitable key for use in drawing the yellow line in the broadcast. When there is a touchdown in the game, the operator instructs the system to draw the yellow line on the screen. The system replaces the pixels corresponding to the stadium ground in the image with those of yellow color and thus creates an appearance of a virtual yellow line on the field. This real-time processing is possible because of a short time delay in the video feed between the cameras and the broadcast. However, unlike a typical compositing system, the yellow-line system does not use a matte for compositing. Although the line is drawn in near real-time, the system has various advantages compared to other compositing systems such as, the use of highly informative feedback from (calibrated) cameras. Moreover, the line is drawn automatically based on the positional information obtained from the cameras,

which convey the angle of view of the camera in the field.

2.6.2 Blue Screen Compositing

An illustration of this compositing method is shown in Fig. 2.9. An image of an object is captured in front of a stationary blue screen or wall. Then, the blue background area behind the object in the image is removed by a color subtraction process or simply by dropping the blue component in the color image. This results in an image which contains only the object of interest in the scene and therefore a corresponding matte for this scene can be obtained. Using this matte, the color removed portion of the object image can be composited on an arbitrary image. In essence, the blue color of the original image is replaced by another image and thus compositing is achieved.

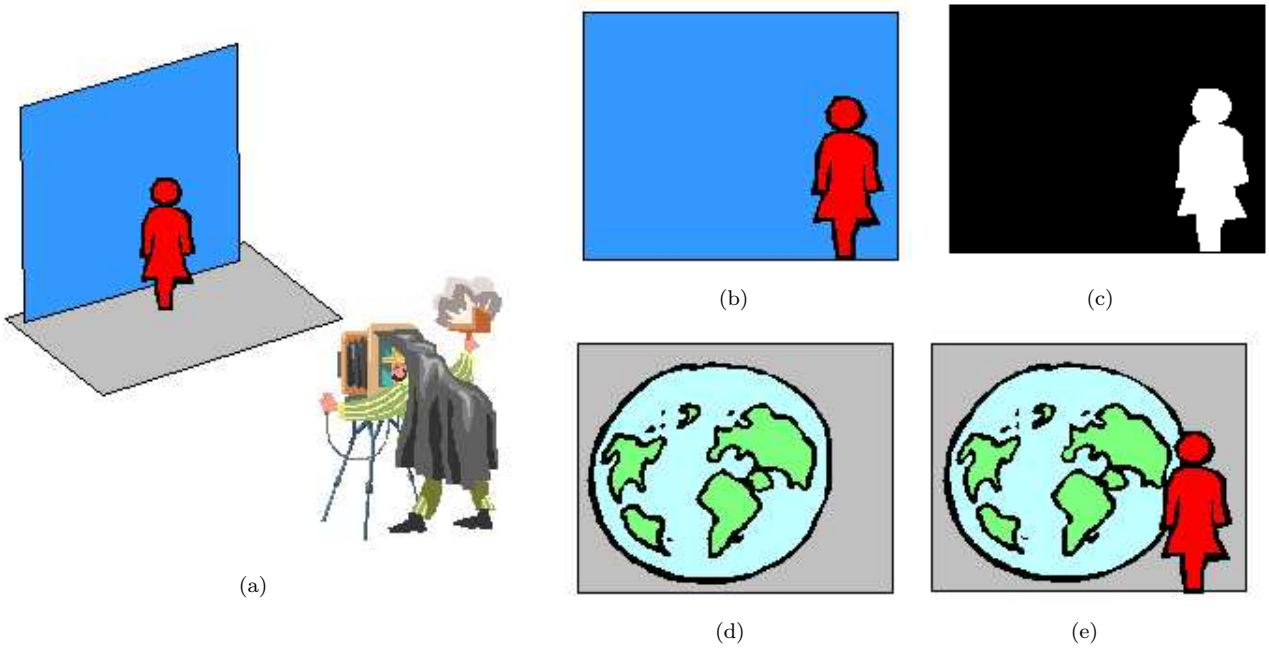


Figure 2.9: Illustration of the blue screen method of image compositing: (a) a person in front of a blue screen in a studio, (b) image captured by a camera, (c) matte extracted from the camera image, (d) an arbitrary image of a map chosen for compositing, and (e) person in the studio composited on to the arbitrary image, creating a visual effect that the person is in front of the map.

Many commercial compositing systems are available to perform blue screen methods in real time. Ultimatte, a company founded in 1970s by Peter Vlahos - the pioneer in compositing, produces a blue screen compositing system. Years of research and development by Ultimatte has led to compositing in scenes with difficult elements like glass and smoke effects.

Image compositing is called *video matting* due to the use of matte in compositing for videos. It is also referred to as *chroma keying,* signifying the use of color (chroma) as a 'key' signal in compositing. Occasionally, a green color background is used instead of blue and hence the use of the term "chroma" in chroma keying. The popularity of chroma-keying technique is evident by the existence of many commercially available systems such as Ultimatte and Corel's Knockout package.

Although the blue screen method is relatively simple, it has disadvantages such as (1) the requirement of a stationary background screen or wall, (2) the object cannot have a color similar to that of the background screen used, and (3) the object cannot have a lustrous surface which may reflect the background color. These disadvantages drive the continuing research for better image compositing systems. Some good references on image compositing include [53, 51].

2.7 Summary

This chapter presented an overview of the topics coordinate geometry, image segmentation, stereo analysis, optical flow, image layers and compositing. These topics form the basic building blocks of the research work and our later discussion.ounded in 1970s by Peter Vlahos, the pioneer in compositing, produces a blue screen compositing system. Years of research and development by Ultimatte has led to compositing in scenes with difficult elements like glass and smoke effects.

Chapter 3

Image Segmentation using a Moving-aperture Lens

This chapter explains the principle of a moving-aperture lens and describes an image segmentation method from the motivating, previous work, which inspired the present work on image layers and compositing. The results obtained from this segmentation procedure convincingly demonstrated the feasibility of extracting layers of a scene using the moving-aperture lens. The discussion in this chapter is based on the previous work [1, 54, 55].

3.1 Auto-stereoscope

In Chapter 2, we learned that disparity refers to the displacement of an image feature between two images. The disparity between the images in a stereo pair is typically not apparent when observed directly. Yet the image pair can invoke a perception of 3D, when viewed using a special viewer called the *stereoscope*, in which two images placed side by side are viewed through a binocular viewer. Other popular techniques to perceive 3D from images encode the disparity information in a single image. An *anaglyph* is a single image with disparity information of a stereo pair encoded in two color channels (generally red and blue)of an image. To perceive 3D, the anaglyphs are viewed using special eye-glasses, with a red and blue filter for the right and left eyes respectively. The two different color inputs received by the eyes are fused to form a single image in the human brain and the disparity between these two inputs creates a perception of depth in the image. Anaglyphs are not generally suitable to display high quality color images, due to their use of color channels and so are limited to inexpensive display units. Another common method employed for 3D perception is

by using polarizers in eye-glasses. In this method, the disparity information is encoded in two orthogonal polarizing directions and the two inputs are projected onto a display device, such as a screen. With the use of orthogonal polarizing filters for each eye, two different inputs are given to the visual system which fuses them into one image and here again, the inherent disparity produces a perception of depth in the image. This technique is not limited by color and so is used in high quality images such as 3D movies.

The development of computer graphics and virtual reality applications fueled the need for 3D viewing. But the major problems with popular 3D perception techniques were that they either require special display equipment or they required custom viewers to perceive depth. Moreover, capturing stereo image pairs was a cumbersome process, which often required two or multiple cameras calibrated on a bulky rig. Such serious limitations curbed the widespread use and applicability of 3D imaging. So naturally there existed a necessity to find better alternatives for 3D perception using normal display devices. Various alternative techniques were proposed which came to be known as "auto-stereoscopic" methods. Auto-stereoscopic display devices neither require special eye-glasses nor custom display equipment to perceive 3D [56].

3.2 Moving-aperture Lens

3.2.1 Introduction

Mayhew and Bacs Jr. [57] proposed an auto-stereoscopic technique for 3D perception using a moving optical element (MOE) for image capture. The MOE consisted of an aperture moving on the aperture plane and included an assembly of lens components. The aperture in a MOE moved in a circular path about the optical axis and as the aperture moved to different positions, the camera captured images. Although the aperture moves, the camera itself remains stationary [58, 59].

The use of a MOE is equivalent to taking multiple images of a scene, with different off-center aperture cameras. The lens elements in the assembly aid in maintaining the focus of the camera. So the image of an object at a focussed distance (on a plane of focus) continues to be projected on the same location in the image plane while objects not at the focal distance tend to be projected at different locations on the image plane, as the aperture moves. In other words, the MOE introduces *planar parallax* in the images. When this image sequence is played back on a normal display device, the induced parallax reappears and the human visual system perceives it as though the objects in the scene were moving (even if they were stationary). This effect creates a perception of depth in the scene (3D), although the displayed images were only 2D.

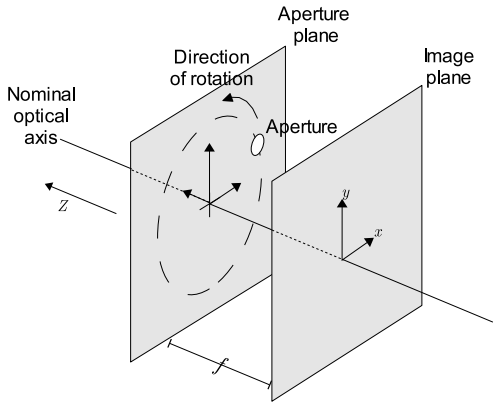


Figure 3.1: An off-axis aperture moving on the 2D aperture plane. For simplicity, the lens components in the assembly are omitted from the figure.

Therefore this technique obviates the need for additional display devices commonly used for 3D perception and hence is an "auto-stereoscope". The MOE can also be called a *moving-aperture lens* and this term will be used in the discussion henceforth. An illustration of the moving-aperture lens tracing a circular path about the optical axes is shown in Fig. 3.1.

Changing the focus setting of a lens alters the focal distance from the camera. The lens produces a well defined image of an object at the focal distance. But objects located at other distances do not appear sharply defined but appear slightly blurred in the image. This effect is well known as *depth of field* in photography. With the moving aperture lens, in addition to the depth of field effect, an image sequence captured also exhibits a unique, interesting visual effect. In the image sequence, a stationary object at the focal distance appears stationary in the sequence whereas stationary objects located at other distances in the scene appear to move in the images. This effect is explained more in detail later.

The moving aperture can be attached to cameras using a standard mount. It has a custom controller to vary the settings such as aperture size (f-stop), radius (amplitude) of aperture motion and frequency of the aperture rotation. A detailed description on the construction and working of the moving-aperture camera can be found in [58].

3.2.2 Principle

The principle behind the moving-aperture design is illustrated in Fig. 3.2. In a standard camera, the aperture is stationary and is located at the center of the aperture plane. But in a moving-aperture lens, the aperture

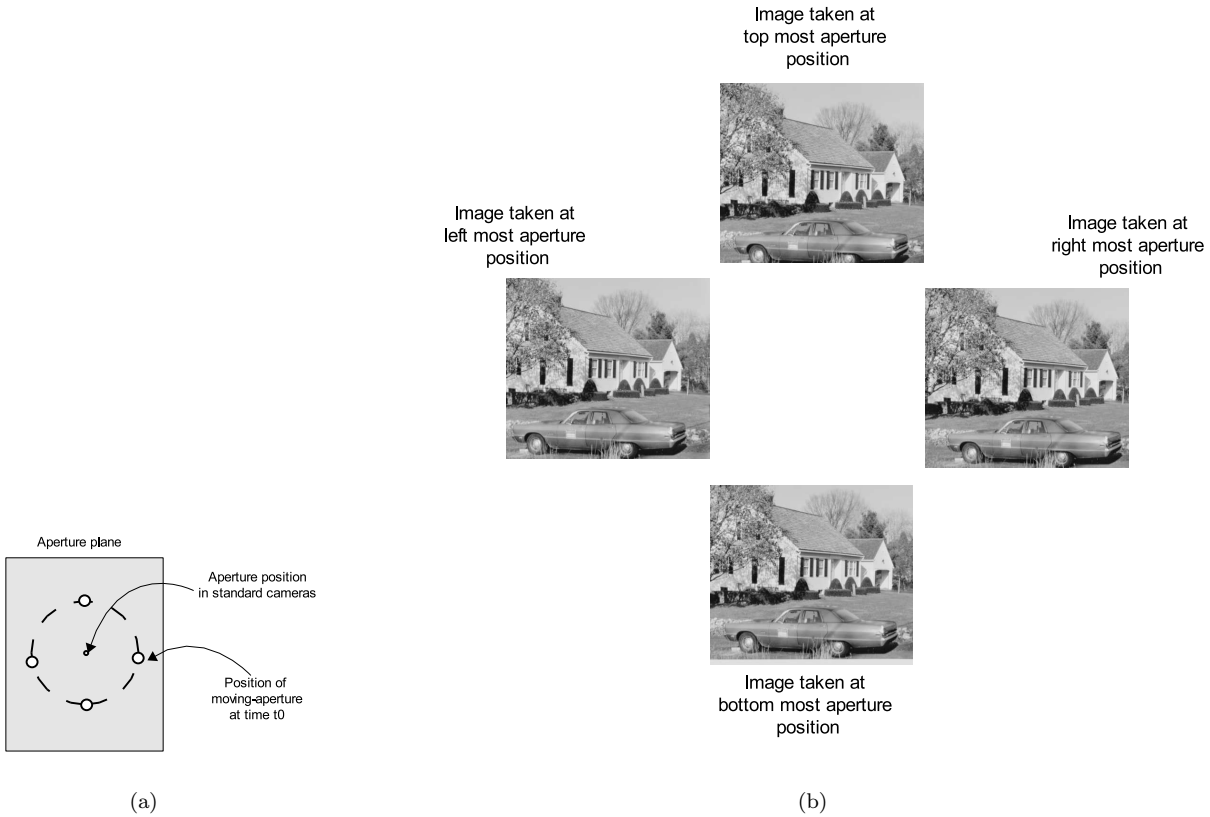


Figure 3.2: Illustration of images captured by a moving-aperture. (a) Different possible positions of a moving aperture, on the aperture plane, as it traces a circular path and (b) the corresponding images captured of a scene. (The images are created manually to illustrate the moving-aperture effect.)

moves over time and traces a circle of some fixed radius, about the center of the aperture plane. The radius of aperture motion can be varied with a knob in the control unit attached to the lens. Although images of the scene viewed statically may look very similar, the inherent parallax effect becomes apparent when the image sequence is observed in real time.

As an example, for the four aperture positions shown in Fig. 3.2(a), the corresponding images captured at these positions are shown in Fig. 3.2(b). It can be seen that the positions of various objects in the scene are displaced by different amounts in the different images.

Consider a stationary scene with three occluding objects located at different distances from the camera, as illustrated in Figure 3.3. Let the scene be imaged using the moving-aperture lens with the pyramid object in the focal plane. The aperture displacement introduces parallax, and therefore image displacement, for

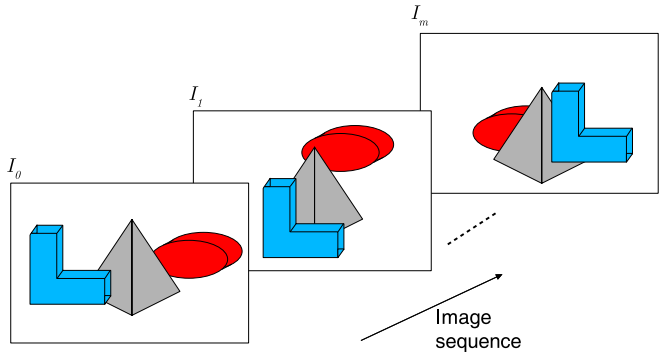


Figure 3.3: Illustration of an MOE image sequence. The camera is focussed on the pyramid shaped object in the middle. The pyramid will remain stationary in the images, while the two other objects in the scene will appear to move (out of phase) in circular paths. The apparent image motion is exaggerated to illustrate the effect.

objects not on the focal plane. As the aperture travels in a circular path, with constant displacement about the nominal optical axis, objects not on the focal plane will appear to travel in elliptical (nearly circular) paths. The displacement amount, which we model as the radius of a circular path, is proportional to distance of the object from the focal plane. In the image, the 'L'-shaped object in front of the pyramid will appear to travel in a circle by a small amount. The ellipse-shaped object beyond the pyramid will also appear to travel in a circular path. Because the 'L' and the ellipse are (respectively) closer and farther than the focal plane, their travel will be anti-phase (out of phase by 180 degrees) along their respective circular paths. With several images acquired during one rotation, the moving aperture lens *simulates* a multi-camera stereo setup, albeit with a very small baseline.

An image sequence made with this camera shows the objects at the plane of focus remain stationary while objects on other distance planes appear to move in a circular fashion in the images. The magnitude of this apparent translation of other objects in images is directly proportional to the radius of aperture motion, as will be explained later.

A change in the amplitude of aperture rotation alters the radius of the circular path traced by the aperture about the center O. Small amplitudes imply a small baseline between aperture positions and therefore a small displacement of objects (or parallax) in the image sequence. A small parallax causes a 3D perception for many observers with a comfortable viewing experience. However, a large amplitude of aperture motion means a large baseline between the off-centered aperture positions and this induces a large planar parallax in the image sequences. A large parallax becomes an annoying visual experience for a human

viewer during playback. But for analyzing image sequences of a scene, large parallax produces a valuable source of information which can be exploited. The main advantage of the moving-aperture lens in image analysis is due to the convenient, small, portable, controllable assembly attached to a single camera, which simulates a multi-camera stereo setup.

The frequency parmaeter set in the controller determines the number of aperture rotations about the plane center and this is also a main factor for comfortable viewing. Empirical results have shown that a frequency of 4.3 Hz gives the strongest depth and image enhancement [60], with a pleasing 3D perception to a human viewer and hence is recommended for videos made for human viewers.

3.2.3 Simple Model

Consider a 3D world coordinate system in XYZ, centered at the aperture plane of a camera O as shown in Fig. 3.4. In a standard camera, the aperture is located at O, the center of the aperture plane and an image plane is located at a focal distance f, from the aperture. The optical axis is indicated by OA. Let C be the point of focus in the 3D world, at a distance Z, from the aperture plane. Thus the coordinates of C in three dimensions are $(0, 0, Z)$. Consider a point P located at $(X, Y, Z+\Delta Z)$. Following the discussion in Section 2.1, the point P corresponds to the point p located at (x, y) on the image plane and is given by

$$x = f\,\frac{X}{(Z+\Delta Z)} \quad \text{and} \quad y = f\,\frac{Y}{(Z+\Delta Z)} \tag{3.1}$$

Now consider that the aperture is moving and traces a circle of radius R, about the center of the aperture plane O, as illustrated in Fig. 3.4. Assume that R is very small (or negligible) when compared with the distance Z in the scene. Let O_m be a position of the moving-aperture at some instance. Then, we can define a coordinate system $X_mY_mZ_m$ with respect to this position.

Consider the same point of focus C, at a distance Z from the aperture plane center O. The moving aperture, with constant focus, forms an imaginary cone with the point C at its apex. Let θ be the half angle of this cone and be called the *scan angle* or *parallax angle*. (In practice, θ is usually small due to implementation reasons and by the assumption stated earlier.) With respect to O_m, the point P is now located at $(X_m, Y_m, Z_m+\Delta Z_m)$ where X_m, Y_m, Z_m and ΔZ_m represent the coordinates with respect to the aperture position at O_m. The corresponding point p_m on the image plane is located at (x_m, y_m), which is slightly different from (x, y) due to the aperture offset. This displacement of the image point represents

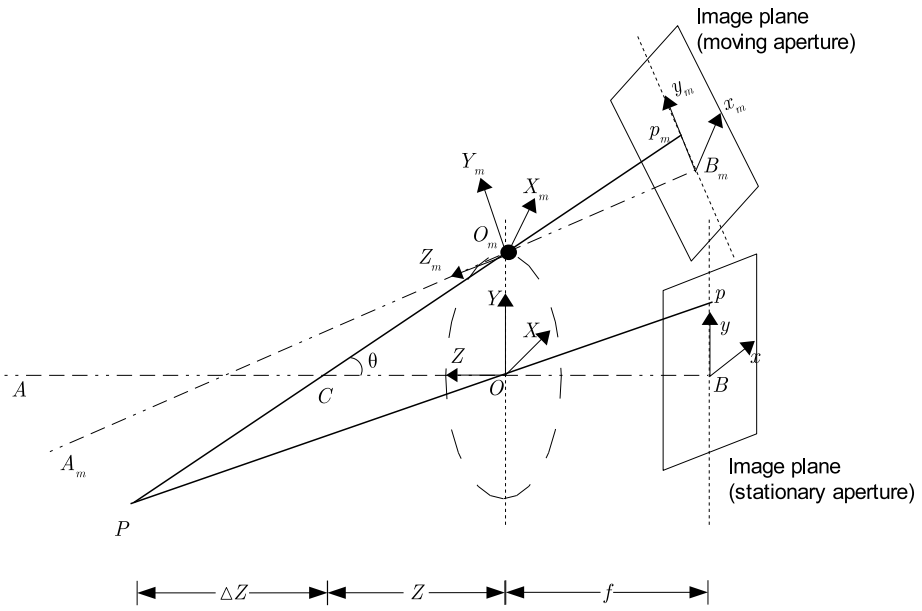


Figure 3.4: Coordinate system and image plane corresponding to a moving-aperture. The radius of aperture motion is shown larger than the image plane for illustration. The lens elements in the assembly are omitted for clarity.

parallax in the image. The displacement (or parallax) along the x axis on the image plane can be written as,

$$\begin{aligned} \Delta x \;\; = \;\; x_m - x \;\; &= \; f\,\frac{X_m}{(Z_m + \Delta Z_m)} - f\,\frac{X}{(Z + \Delta Z)} \\ &= \; f\,\frac{\Delta Z \sin\theta + X\cos\theta}{(Z_m + \Delta Z\cos\theta - X\sin\theta)} - f\,\frac{X}{(Z + \Delta Z)} \end{aligned} \tag{3.2}$$

Dividing the first numerator and denominator by $\cos\theta$, and using the approximations $Z \;\approx\; \frac{Z_m}{\cos\theta}$ and $X\tan\theta \approx 0$, the above equation can be written as,

$$\Delta x \approx \frac{\Delta Z}{(Z + \Delta Z)}\, f\tan\theta \tag{3.3}$$

Therefore for a given focal length and scan angle, the resulting image displacement Δx is a function of the focal length f, distance to the point of focus Z and the relative object distance ΔZ. Objects at the plane of focus correspond to the case $\Delta Z = 0$, and so there is no image displacement (parallax), as mentioned earlier. If an object is located beyond the plane of focus, the ΔZ is positive and so the corresponding image displacement Δx is also positive. But if an object is located in front of the plane of focus, then ΔZ is

negative. So its image displacement will be opposite to that of an object beyond the plane of focus. Thus, the moving aperture induces an anti-phase relation in image parallax, for objects in front and beyond the focal distance. The sign of ΔZ therefore indicates the relative distance of an object as nearer or farther than the plane of focus. In other words, Δx3.3) gives the relative location of an object in the image from the plane of focus, depending on the sign of ΔZ.

A corollary can be derived from the relation in (3.3). The displacement Δx is dependent on the distance ΔZ from the plane of focus and the parallax on a two-dimensional $X - Y$ plane at a given distance Z is constant for a corresponding region in the image. Because the distance Z (and ΔZ) is defined along the optical axis, perpendicular to the image plane, the induced parallax is planar, where the planes are 2D $(X - Y)$ at various distances Z and so are parallel to the image plane. These planes are commonly referred to as 'fronto-parallel planes'. Therefore, a moving-aperture induces fronto-parallel, planar parallax, with respect to the plane of focus in a scene. This is an important characteristic of the moving-aperture lens.

Rearranging (3.3), we get

$$\Delta Z = \frac{Z\,\Delta x}{f \tan\theta - \Delta x} \tag{3.4}$$

This relation gives the relative distance ΔZ in a scene when other parameters in an image are known. The image displacement (parallax) Δx, can be estimated from some image-analysis technique. With knowledge of the focal length and distance to the plane of focus, the distance of any object in the scene ΔZ, can be easily estimated. It should be noted that this type of range estimation is a passive-range technique as it does not involve any emission of energy in the scene. Therefore this approach of range estimation using a moving-aperture lens is a valuable alternative where other active range estimation approaches are not preferred.

The pattern of motion by the aperture on its plane is not restricted to circular design but can also be set to elliptical patterns. (A later version of the moving-aperture allows more complex patterns that can be programmed through a separate controller.) The circular pattern of motion was chosen for this research work mainly for convenience and simplicity in analysis.

3.3 Methodology

From the discussion in Section 3.2, it is understood that a moving-aperture lens induces image parallax based on the fronto-parallel planes in a scene. This parallax corresponds to the displacement Δx on the image plane, which is dependent on the distance from the plane of focus ΔZ (see (3.3)).

The apparent motion of each point in an image constitutes 'optical flow' (or 'image flow') between successive image pairs in the image sequence. Over the entire sequence, the vectors of image flow (or displacement of image points between two images) for each image point, will also follow a circular pattern of movement as the aperture traces a circular path. In the following discussion, we refer to image *disparity* as the *radius* of circular motion at an image point. For objects at different depth planes, the magnitude of optical flow varies and hence their disparity.[1]

3.3.1 Optical Flow

The estimation of optical flow in a moving-aperture image sequence is the most important part for image segmentation. The optical flow can be estimated as follows. Consider a sequence of N images $(I_0, I_1, ..., I_{N-1})$ from one rotation of the the moving-aperture lens. Choose an image pair $I_0$and I_1 from this sequence. Select an image window I_{w0} centered at a point (x, y) in the first image I_0. Using a classical pattern matching technique like block matching with normalized cross-correlation, we can find the location of a best match for I_{w0} in the other image I_1 (see Section 2.4). Let the best matching image window in I_1 be located at $(x + \Delta x, y + \Delta y)$ and be represented as I_{w1}. The difference between the two positions of the image windows I_{w0} and I_{w1} therefore gives the optical flow $(v^x, v^y) = (\Delta x, \Delta y)$, at the particular image point (x, y) along the respective directions in I_0. In an alternate representation, we can write the optical flow at a point simply as v, whose magnitude and angle are given as $|v| = \sqrt{(v^x)^2 + (v^x)^2}$ and $\angle v = \arctan\left(\frac{v^y}{v^x}\right)$.

Repeat the estimation for every point in the image I_0 and find the corresponding optical flow from the $I_0 - I_1$ pair. The result is a dense estimate of the optical flow for the entire image I_0, and this is represented as V^x_{01}, and V^y_{01}, where the subscript indicate the image pair and the superscript indicates the component direction. This two-component optical flow map can alternately represented as V_{01}, where each point in the map is a vector with a magnitude, and direction.

[1] In our discussion, we shall refer to the image motion between any two images as "optical flow" and the net displacement of image points in an image sequence as "disparity". This subtle difference in terms is essential to distinguish the various image motions that arise in analyzing a moving-aperture image sequence.

Now, repeat the optical flow estimation procedure for the next image pair $I_1$and I_2 in the sequence, to get another dense optical flow map, V_{12}^x, and V_{12}^y corresponding to the image flow for I_1. Similarly obtain dense flow maps V_{23}^x,V_{34}^x, ...$V_{N-2,N-1}^x$ and V_{23}^y,V_{34}^y, ...$V_{N-2,N-1}^y$ from the entire image sequence.

3.3.2 Disparity Estimation

The dense optical flow maps describe the motion of individual image points in a sequence. Say p is an image point at (x_{p0}, y_{p0}) in the image I_0. The corresponding flow vectors of p to I_1 are $v^x = \Delta x$ and $v^y = \Delta y$, which represent a point $(x_{p0} + \Delta x, y_{p0} + \Delta y)$, which can be denoted as (x_{p1}, y_{p1}). Similarly the locations of (x_{p2}, y_{p2}), (x_{p3}, y_{p3}), (x_{p4}, y_{p4}), ..., $(x_{p(N-1)}, y_{p(N-1)})$ from images $I_2, I_3, I_4, ..., I_{N-1}$, respectively, can also be found as the point moves in the image sequence. The locations of points in the image sequence are a function of the aperture motion. This is shown in Fig. 3.5.

Because the aperture traces a circular path, these points will also (ideally) tend to lie on a circle. The radius r of the circle corresponding to this image point can be calculated using a least-squares based circle fitting procedure (Appendix A). The result of circle fitting is the center of a best fitting circle for the given data and also the radius of the circle. The radius corresponds to the disparity of a point in the image sequence.

It should be emphasized that the number of images $(I_0, I_1, ..., I_{N-1})$ chosen for this analysis should be sufficient to fit a circle, i.e., $N \geq 3$, as otherwise a unique circle cannot be fit and so the fitting procedure will result in large errors. Also, it is redundant to select a large N, because using overlapping data points in circle fitting may not improve the result significantly.

3.3.3 Segmentation

Once the disparities (i.e., radii of motion) for each image point in the image sequence are calculated, a segmentation step is necessary to group pixels based on these disparity values. The result of segmentation is a set of non-overlapping image regions which corresponds to various objects at various distances in the 2D image of a scene.

The algorithm for segmentation from an image sequence of a stationary scene can be summarized as follows.

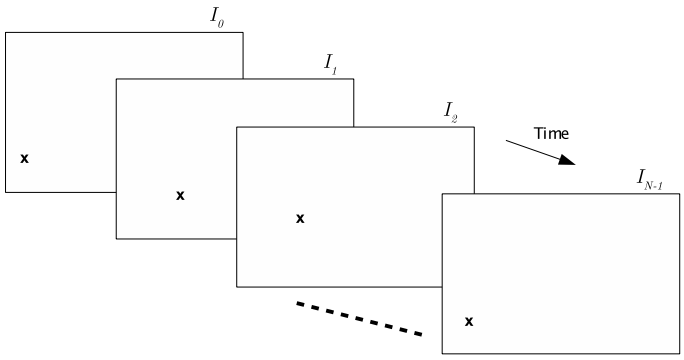


Figure 3.5: Illustration of an image point moving in a moving-aperture image sequence. The point shown in Fig. 3.5, when tracked in the image sequence traces a path similar to the path traced by the aperture.

1. Threshold the estimated disparities using measure of fit values. Large value for measure of fit indicates a poor circle fit at an image point, and so a threshold must be used to identify invalid, large disparity values. These are set to zero in our implementation.

2. Next threshold using the maximum radius allowed (or possible) in an image sequence and set all those radius values greater than this threshold to be the maximum possible radius. The anticipated maximum radius can be determined from the scan angle or empirically from the video sequence.

3. Smooth the thresholded disparity data using a 3×3 averaging filter.

4. Group the disparities into distinct regions (based on similar values) using 8-neighbor connectivity and morphological erosion technique. The regions grouped represent various objects at different distances in the scene.

5. Assign unique labels to each identified region. The labelled regions are non-overlapping segments (partitions) of the image of a scene.

3.3.4 Range Estimation

In addition to image segmentation, the moving-aperture image sequence can be used to estimate the location of the objects (or depth plane) in the scene as discussed in Section 3.2.3.

The circle fitting algorithm estimates the center of the circular motion and the radius. If the difference between the initial point in the sequence and the calculated center is considered, the direction of shift along the x axis offers highly valuable information for range sensing.

Figure 3.6 shows the motion of a scene point in the image sequence. Although a point in the scene is assumed to be stationary, the illustration in the figure is only shown to demonstrate the moving-aperture lens, in a three object scene. Consider a point in focus C, as shown in Fig. 3.6. As the aperture moves, the point C remains at the apex of an imaginary cone and continues to be in focus. The radius of motion at this image point is zero.

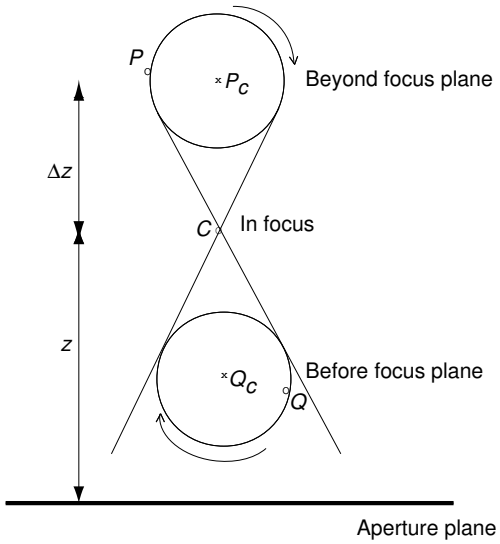


Figure 3.6: Illustration of location estimation based on image parallax.

For an object (or distance plane) in front of the focus plane, there exists some motion in the images. In this case, the difference between an initial point Q from I_1 and the calculated center Q_c indicates a shift. In the figure, the shift from Q to Q_c is to the left and hence can be considered toward the negative X axis. Considering the same image sequence, an initial point P, on a plane beyond the focus will exhibit a positive shift from the calculated center P_c. This shift information can be used to identify the location of objects relative to the plane of focus.

The results of location estimation, when combined with segmentation result, gives the range values in the scene. The algorithm for range estimation can be summarized as follows.

1. Apply the segmentation algorithm to identify image regions that are related to object distance.
2. For each point in the image, calculate the direction of shift between the initial point and the calculated center of flow.
3. Use principal component analysis (PCA) to find the principal axes in a scatter diagram of flow directions

from Step 2.

4. Discard outliers that correspond to very large radius values.

5. With respect to a coordinate system that is aligned with the calculated principal axes, transform the points.

6. Data points at the origin (or in focus) correspond to no flow; data points on the left half-plane correspond to image points on one side of the plane of focus and the data on the right half-plane correspond to image points on the other side of the plane of focus.

7. From the disparities calculated during image segmentation, the distances of regions can be determined and so the regions can be identified as 'near' and 'far' regions.

3.4 Study Setup

A video sequence was acquired with the moving-aperture lens attached to a camera looking at a real world scene – a line of trees as shown in Fig. 3.7. The trunks of the trees in the image appear to be regularly spaced and are at different distances from the camera. For convenience in the following discussion, the trees are labeled $T1$, $T2$, $T3$ and $T4$ in Fig. 3.7. Let the region beyond $T4$ be referred as the *background* for this image.

Figure 3.7: Example image obtained using a moving-aperture camera. The trees are labeled $T1 - T4$ on this image, for convenience. $T1$ is the nearest tree and $T4$ is farthest tree from the camera and observer. Trees $T2$ and $T3$ are in between the other two trees in the scene. (Thanks to Vision III Imaging, Herndon, VA for providing the video sequence used in this research.)

The original video sequence in S-VHS video format was converted to a digital image sequence at 640×480

resolution. The color images of this sequence were converted to grayscale images for segmentation.

Because the video included a demonstration of the controller for moving-aperture lens capabilities, the irrelevant portions of the image were discarded. So only a portion of images in the sequence, in which the trees were present were taken for analysis. The dimensions of the cropped portion of the image as shown in Fig. 3.7 was of size 350×300 (350 pixels length by 300 pixels height), extracted from the 640×480 size original image.

The single image shown in Fig. 3.7 is sufficient for a human viewer to conclude that there are four trees with bushes between the trees in the scene. It is also easy to infer that $T1$ is close to the observer while T4 is far from the observer. This simple, yet exceedingly complex feat by the human visual system is possible only because the visual system utilizes clues like relative size, intensity and texture strongly augmented by acquired knowledge over generations. But unlike the human visual system, computer vision systems have to rely on various complex algorithms to acquire the similar information.

When the camera focus changes, some dramatic visual changes occur in the image sequence. Fig. 3.8 shows the images captured at different focus settings – when $T1, T2$ and $T4$ are in focus respectively. It can be observed that in Fig. 3.8(a), $T1$ is sharply defined while the background is blurry. In Fig. 3.8(b), where $T2$ is in focus, the knot in the tree is clearly visible which was not seen when $T1$ was in focus. When the background is in focus as shown in Fig. 3.8(c), the background and the bushes between trees are clearly visible. [2]

(a) (b) (c)

Figure 3.8: Effect of changing focus: (a) $T1$ in focus; (b) $T2$ in focus; (c) background in focus. The object or region in focus appears clearly defined in the image while other areas of the image do not appear so sharp.

Consider the case when the camera is focused on tree $T2$. As ΔZ is approximately (or effectively) zero for

[2] Although the images are carefully chosen to differentiate the focus settings, the details may not be easily visible in print.

all points on tree $T2$, the optical flow Δx is also zero for all image points corresponding to $T2$. We know that the disparity (or radius of image motion) increases when the distance from the point of focus ΔZ increases. Therefore the optical flow magnitude in the image points corresponding to $T4$ will be more than that for $T3$. When ΔZ is negative as with tree $T1$, Δx also becomes negative (refer to (3.3)). But a negative Δx makes no difference in the disparity except to induce an anti-phase nature. In short, the induced parallax in the image sequence can be considered as an imaginary double cone, whose apex is at the point of focus $T2$ and whose circles of cross sectional represent the increasing disparity with increasing distance from the apex of the cone as shown in Fig. 3.9.

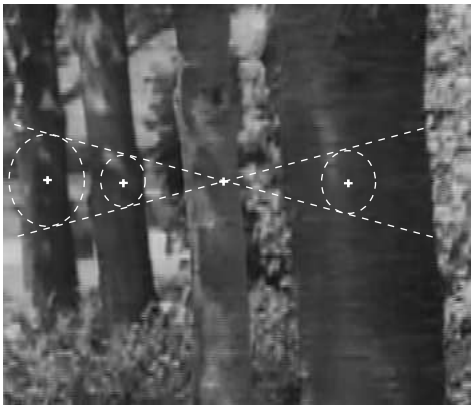


Figure 3.9: Illustration of increasing disparity (radius of circular motion) in moving-aperture images. At the point of focus, there is no perceived image motion. As the distance from the plane of focus increases in a scene, the disparity also increases.

3.5 Results - Far Focus

The accuracy of optical flow estimation, disparity calculation and hence the later image segmentation depend on the quality and resolution of images used in the analysis. To aid in subpixel resolution analysis, the 350×300 size images were interpolated to 700×600 size using a bilinear interpolation method.

For this discussion, the analysis for the case when the camera was focused beyond $T4$ (i.e., far focus) is chosen. In this case, we expect to find zero optical flow for the image points beyond $T4$. Following the earlier discussion in Section 3.4, we expect an increasing disparity for regions in image corresponding to increasing distance from tree $T4$. It should be noted that the low-lying bushes in the scene are not located at distinct depth planes but instead lie in the ground plane. So the results obtained in this region of the image can create discrepancies in analysis and therefore deserve a closer examination.

In this analysis, images captured in one rotation of the moving-aperture were taken for analysis. The demonstration video was shot at 4.3 Hz and the image sequence was digitized at 30 frames per second from the S-VHS video format. This gives approximately 7 images per rotation of the aperture.

3.5.1 Optical Flow and Disparity

The optical flow vectors in adjacent image pairs of the sequence were estimated using the block-matching method described earlier. The image window used was of size 21×21. From the optical flow estimates for each block in the image, a circle was fit to the flow vector at each location in the image to calculate the disparity in the image at that location as described earlier.

Figure 3.10(a) shows a disparity map (in pixels) calculated for the entire image. The color scale adjacent to this map shows the disparity (parallax) values at various points in the image. Blue, at the bottom of the color scale indicates a small disparity (or image motion) while red, at the top of the scale implies a large disparity. From this map, it can be inferred that disparity increases in the image regions corresponding to increasing distances toward $T1$, from the plane of focus in the scene.

We notice some noise present in the estimates of Fig. 3.10(b) and this noise needs to be removed to achieve a good image segmentation. The first step is to use threshold based on the measure of fit from circle fitting (Appendix A) and set all disparities with poor measure of fit to be zero. The second step in noise removal is to threshold the parallax (circle radius) values based on the maximum expected (in pixels), which can be approximately obtained by a human observer viewing the image sequence or numerically from (3.3) with the knowledge of some parameters in the scene. In the chosen example, the maximum parallax was visually approximated as 6 pixels in the image and any parallax estimated as more than this value was set to zero. The result obtained after thresholding is shown in Fig. 3.10(b).

3.5.2 Segmentation

After noise removal, the disparity map is smoothed by applying a 3×3 low-pass filter, whose result is shown in Fig. 3.10(c). This map shows four, nearly distinct image regions which approximately correspond to the four trees in the image of the real scene and hence an encouraging sign towards the goal of image segmentation. We now apply a morphological erosion operation [29] to isolate the image regions and then use a region-labeling procedure to uniquely label each image region. The result of this operation shown in

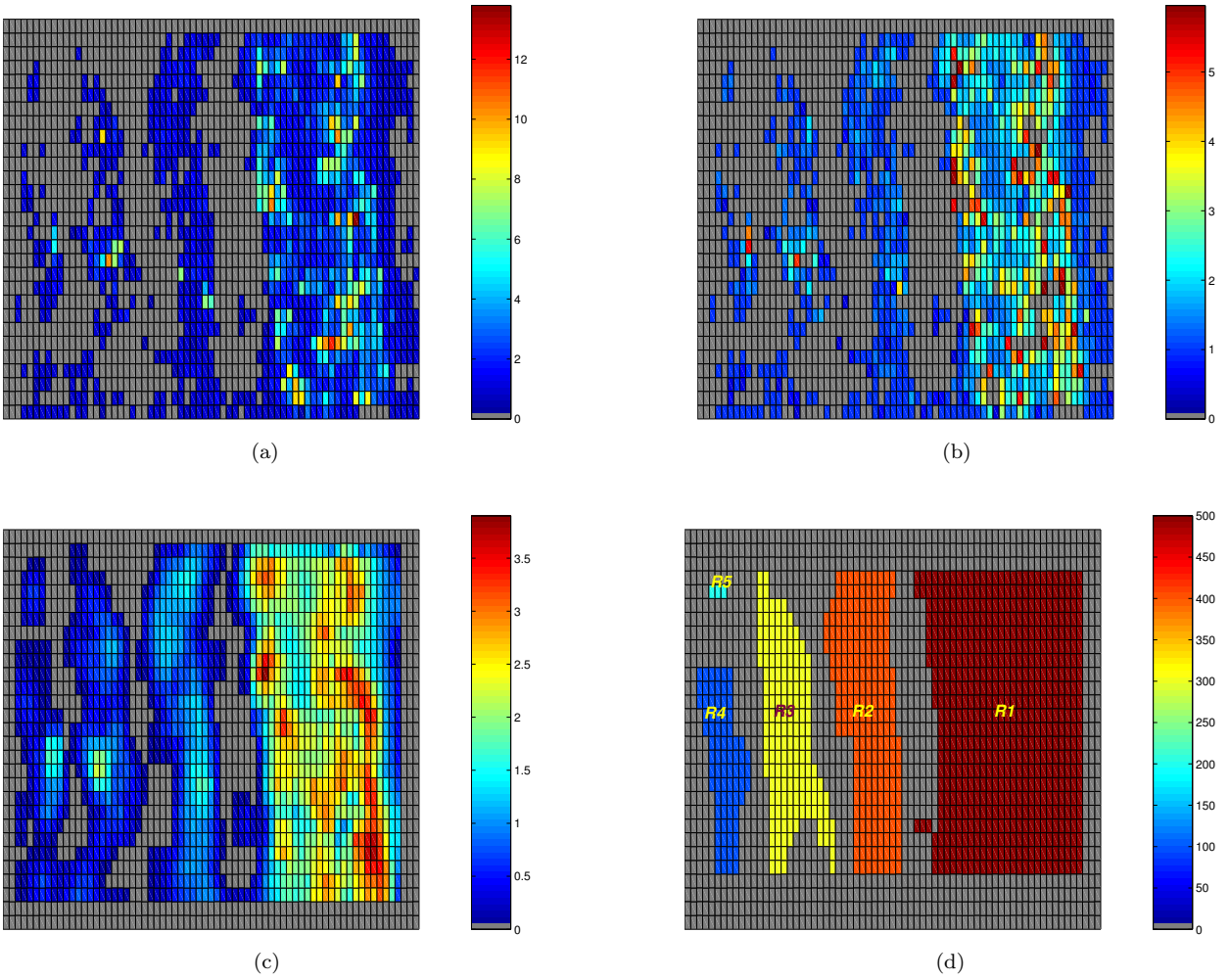


Figure 3.10: Results when the camera focus is beyond $T4$: (a) calculated disparities; (b) disparities after thresholding based on measure-of-fit and maximum radius; (c) smoothed result using a 3×3 low-pass filter; (d) segmented result after morphological erosion and region labeling, showing regions corresponding to the tree trunks in the scene.

Fig. 3.10(d) clearly shows the distinct trees $T1 - T4$. (The region labels were manually added in the result shown here for convenience.)

As the background was in focus, the region between the trees which can also be considered as background also are in focus. Therefore the trees in the scene are the only image regions which are not in focus and so exhibited apparent image motion (optical flow, hence disparity) in the image sequence. The segmentation result obtained clearly represents this expected observation and hence the success of the image segmentation approach using a moving-aperture lens. Table 3.1 shows the average parallax (circle radius) values corre-

sponding to each image region. It can be seen that as the distance from the plane of focus in the scene increases the disparity of the corresponding image region also increases.

Table 3.1: Average disparity of image regions, with a far focus setting.

Image region	Average disparity (pixels)
R1	1.969
R2	0.829
R3	0.605
R4	0.554
R5	0.522

3.5.3 Range Estimation

The computation of numerical values of object distances in the scene using (3.3), requires that the camera parameters are known. Unfortunately, these parameters were not available for the images used and so the exact numerical values of range could not be calculated. But the table demonstrates the feasibility of such numerical calculations with knowledge of more parameters and most importantly, the proportionality relation of parallax and real world distance, as expected.

Having segmented the trees at different depth planes, the location of trees in the scene can be estimated. In the absence of an exact information on the position of the first image in the aperture motion, only location estimation relative to the plane of focus is possible. Figure 3.11(a) shows a plot of the differences between an initial point in the sequence and its corresponding circle center estimated from the circle fitting procedure. The radial distance from the origin indicate the magnitude of parallax and the angular location from the axes indicate the direction of the estimated circle center in the analysis.

Principal component analysis is used to identify the principal axes of this data and then groups the data into three classes - zero shift, positive shift and negative shift with respect to an initial point from the first image. Zero shift in the data indicate that the point is in focus. A positive shift implies positive disparity at an image point and it corresponds to an object located on one side of the focus - either in front, or beyond the plane of focus. In contrast, a negative shift indicates the presence of an object at the other side of focus. A threshold value is used to remove the noisy estimates, based on the anticipated maximum radius. The circle in Fig. 3.11(a) indicates the disparity (or radius) threshold of 5 pixels used as the threshold in this analysis.

Figure 3.11(b) shows the result after identifying the axes of the data, using principal component analysis

(PCA). There exists multiple image points with same (or approximately same) disparity magnitudes and so some data in this plot represent multiple images, although they are not apparent in the plot.

After grouping the data into three classes - in focus (zero shift), on one-side of focus (say, positive shift) and the other-side of focus (negative shift), the corresponding image points appear as shown in Fig. 3.11(c). This plot represents the relative location of objects in the scene. This plot shows image points in three different colors, as given by the color scale - red, corresponding to points with zero shift and hence in focus, blue and light-green corresponding to points on either sides of the focus (in front of or beyond the focus).

Figure 3.11(d) shows the location estimation results merged with the segmentation results (from Fig. 3.10(d)). The region shown in red indicates the focus while blue and light green regions indicate the other regions not in focus. This result shows that all the trees $T1 - T4$ lie on one side of the focus, which is true - they are in front of the focus. One possible reason the location estimation resulted in $T3$ and $T4$ as in-focus objects although they were not, is the various post-processing operations such as, thresholding and filtering operations performed on the noisy data. We need to recall that the quality of images used in this analysis was not very good - the image sequence was obtained after multiple image conversion steps and hence the loss in quality. Nevertheless, it is convincingly clear that better, more accurate results can be obtained using better image quality and with more knowledge of camera parameters.

3.6 Summary

This chapter demonstrated that image segmentation and range estimation are possible using a moving-aperture lens. The disparity in a moving-aperture image sequence increases with increasing distance from the plane of focus. Therefore image segmentation based on the planar, image parallax is shown to work using an example sequence. With some knowledge of camera parameters, the actual distances of objects in the scene could also be estimated. The results provide a convincing proof for the feasibility of extracting image layers based on distance in a scene, using the moving-aperture lens.

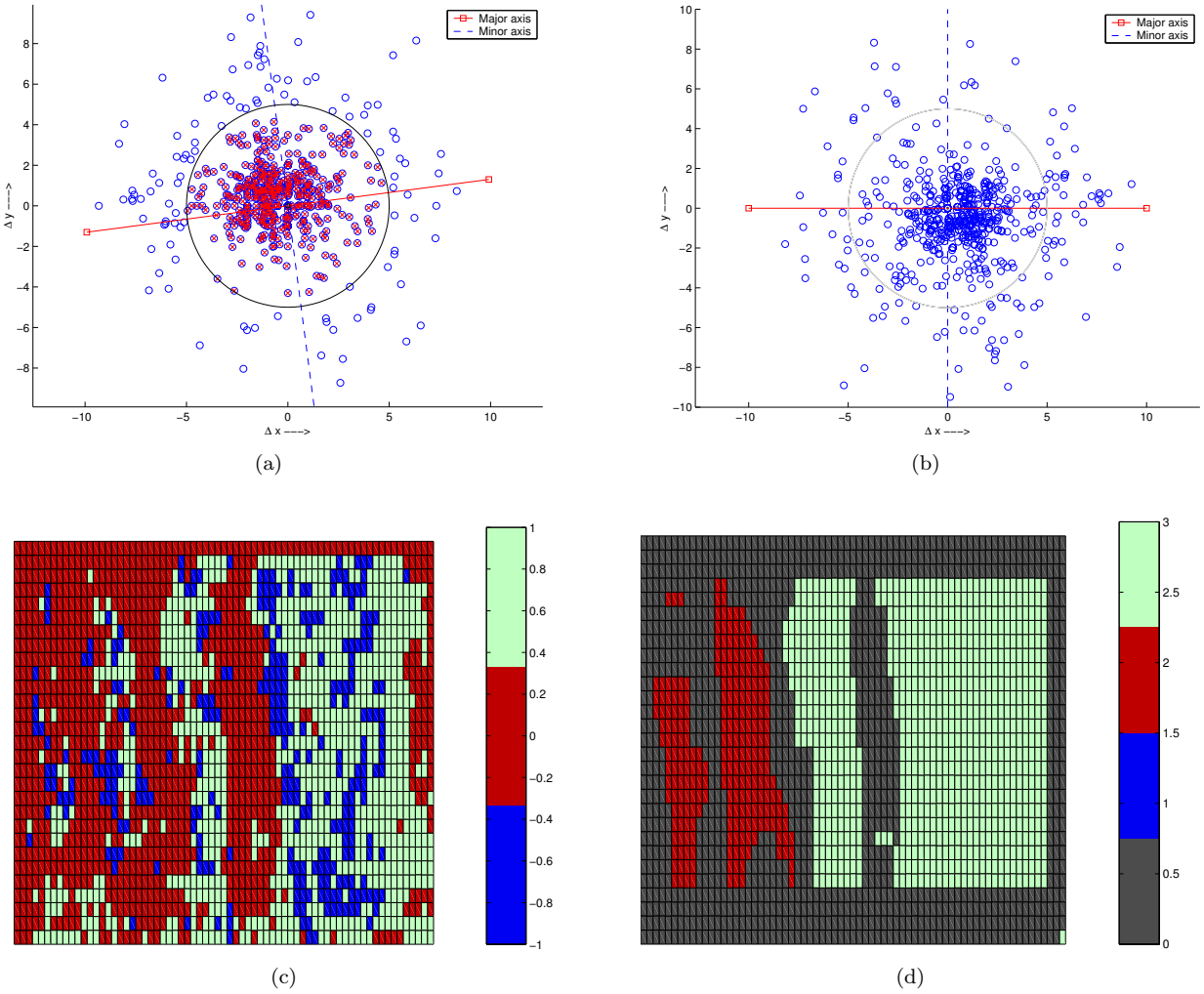


Figure 3.11: Location estimation: (a) plot of differences between an initial point in the first image of the sequence and its corresponding calculated center; (b) data transformed using principal component analysis (PCA); (c) near-far locations estimated using PCA; (d) near-far estimates merged with the segmentation result.

Chapter 4

Layer Extraction and Image Compositing

This chapter presents results from a test for image compositing, using the image segmentation results from the previous work discussed in Chapter 3. The composited output served as a proof-of-concept for automatic matte extraction and image compositing using the moving-aperture lens. This chapter outlines the methodology developed to address the research problem and describes in detail, the various steps involved in the process of layer extraction and image compositing using the moving-aperture lens.

4.1 Image Compositing - Proof of Concept

The image segmentation technique using a moving-aperture lens described in Chapter 3 automatically identified different regions in the image corresponding to real objects in a scene, and it partitioned the image into representative image regions or 'image segments'. The obtained results were reasonably good, although the input images were noisy and relatively low in quality.

As a simple test for image compositing, the segmentation results from this technique were used as a preliminary check to test the possibility of compositing using the moving-aperture lens. For this test, the image segmentation result from Fig. 3.10(d) was used. The camera image corresponding to the segmented result is shown in Fig. 4.1(a) and an arbitrary image chosen as a background for compositing is shown in Fig. 4.1(b).

In this test, a binary matte for compositing was derived from the segmentation result. In the binary

matte, regions with $\alpha = 1$ correspond to the tree trunks from the camera image (foreground) and regions with $\alpha = 0$ correspond to all other regions in the camera image. Therefore, in compositing by superimposing the camera image over the arbitrary background, only image regions corresponding to the tree trunks are visible in the composited output, while the remaining regions of the output are obtained from the background using an inverse matte.

Because the technique in Chapter 2 used block-matching with an image window of size 21×21 in the optical flow estimation step, the image segmentation result appears "blocky". Therefore, the matte extracted from the segmented output will also contain this artifact, which will appear on the composited output. To mitigate this artifact, a sequence of additional morphological operations [29] was used. Applying a morphological closing operation, dilation followed by erosion, removed the sharp inner edges present in the segmented result. However, the outer edges were not affected by this operation. So compositing with a matte from this post-processed result of segmentation produced a composited output, which is shown in Fig. 4.1(c). The artifacts of block-matching and morphological operation are clearly visible in this output. Using a morphological opening operation (erosion followed by dilation) instead of a closing operation helped to smooth the sharp outer edges of image regions but did not affect the inner edges of image regions. The composited image obtained using this post-processed result of image segmentation, is shown in Fig. 4.1(d).

It should be noted that compositing in this test was achieved from a small number of low quality images digitized from a video cassette, and also no information about the scene or camera parameters was available. Yet the result clearly shows the feasibility of the concept.

From these two images, it is clearly evident that a moving-aperture lens can be used for matte extraction and image compositing. The highlight of this preliminary test for compositing is that no manual tweaking was performed to extract the tree trunks from the original images, and that the entire process was automatic. With a moving-aperture lens and knowledge of more information about a scene, it should be possible to obtain better, more satisfying composited images. Thus an automatic image compositing method using the moving-aperture lens is feasible and viable for some scenes.

4.2 Planar Parallax and Moving-aperture Lens

As discussed in Chapter 2, the displacement of corresponding points in stereo images varies depending on the position of cameras and the distance of objects in the scene. When the disparity in two images is constant

(a) (b)

(c) (d)

Figure 4.1: Proof of concept for image compositing using the moving-aperture lens. (a) An image captured using a moving-aperture lens; (b) an *arbitrary* background image chosen for compositing. The matte for compositing was obtained from a morphological post-processing on the segmented result in Fig. 3.10(d). The composited output images obtained: (c) after a morphological closing operation in post-processing; (d) after a morphological open operation on the segmented result.

over an entire image region and varies only based on the distance of the plane from the cameras, then the parallax is planar in nature (*planar parallax*). Sawhney[61], Kumar et al. [62], and Irani and Anandan [63] have investigated planar parallax in images to analyze and recover the structure of objects in a 3D scene.

In an image sequence from a moving-aperture lens, the magnitude of parallax is proportional to the object distance in 3D world and therefore the parallax induced is planar. But the main difference from other works on planar parallax is that the parallax here is residual – *with respect to the plane of focus*, i.e., – the parallax is zero at the plane of focus and it increases with increasing distances on either side of the plane of focus.

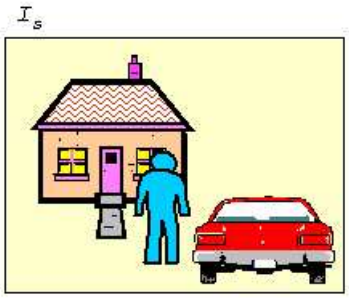


Figure 4.2: Illustration of an example stationary scene. Image I_s, captured with a standard camera.

4.3 General Setup

Consider a stationary real-world scene with multiple objects and let this scene be imaged using a standard camera with a stationary aperture. In the image I_s illustrated in Fig. 4.2, there are three distinct objects: a car close to the camera, a person, and a house far behind the person. There is also a background corresponding to the remaining portion of the image. Let the camera be focussed on the person for our discussion here.

Now consider the same scene imaged using a camera with a moving-aperture lens. As we already know, the aperture moves on its 2D plane and traces a circle on the aperture plane, centered about the Z axis. The camera captures a sequence of images at a constant rate (in frames per second) as the aperture moves. Therefore, the images in the sequence are captured at different positions of the off-center aperture, as it traced a circular path on its plane.

The controller for the moving-aperture lens allows variation of the frequency of aperture rotation (cycles per second), radius of aperture motion and f-stop of the aperture. We assume that the speed of aperture motion (angular velocity) is a constant and ignore any minor mechanical inaccuracies in operation of the moving aperture. We also assume that motion blur introduced in the images is negligible.

Let the frequency of aperture rotation be γ (number of cycles per second) and remain constant in an image sequence. Consider that the camera capture rate κ (number of images pers second) also remains fixed during the image capture. Therefore with this setting, the number of images captured in one complete cycle of the aperture is

$$N = \frac{\kappa}{\gamma} \tag{4.1}$$

When N is an integer, the images tend to repeat in a sequence. This indicates that the images are captured

(ideally) at the same positions of aperture on its path, in each cycle. In other words, the N positions represent N sample points on a circle of some constant radius. But, when N is not an integer, it implies that the aperture positions do not overlap between cycles and so this case is equivalent to sampling a circle with non-uniform spacing. However in both cases, the N positions can be considered to represent the vertices of an N-sided polygon. Let us refer this as an 'aperture-polygon', to indicate that it is on the aperture plane ($X-Y$ plane at Z=0).

For a stationary scene and camera, a sequence where images are captured with an integer N, at repeating, off-center positions of the aperture in every cycle of aperture rotation, ignoring noise in the images, we can write

$$I_{j+N} = I_j \qquad j = 0, 1, 2, ..., N-1$$

Therefore to analyze a scene that is stationary and captured using a moving-aperture lens, a set of any N consecutive images from an image sequence will be sufficient.

Consider an image sequence captured with the moving-aperture lens, where $N = 4$ images per cycle. Let the radius of the circle of aperture motion be a constant, represented by R (in mm). There are four positions on an imaginary circle in the aperture plane, corresponding to the aperture positions where the images are captured. The constant angular velocity assumption of the moving aperture ensures that these four positions are angularly equidistant and so overlap in every cycle. A possible configuration of these aperture positions is a square (aperture-polygon) as shown in Fig. 4.3. The four positions of the aperture as it moves on its path over time, at equal time intervals can be represented as a_0, a_1, a_2 and a_3 on the 2D plane.

For the example scene in Fig. 4.2, the images I_0, I_1, I_2 and I_3 are captured at aperture positions a_0, a_1, a_2 and a_3 respectively. When the camera is focussed on the person, the plane of focus is at a distance equal to the distance of the person from the camera. Therefore, the car in front of the person is located in a distance plane at a shorter distance from the camera whereas the house behind the person is located at a farther distance from the camera (on the opposite side of the distance plane of the car). Therefore, in a moving-aperture image sequence, the person will appear stationary, while the car, the house and the background will appear to move (although stationary in the scene) – in an anti-phase relation. The captured images for this imaginary scene corresponding to the four aperture positions a_0, a_1, a_2 and a_3 will be as shown in Fig. 4.4. The relative displacements of objects in the scene are sufficiently illustrated – with the person in focus being stationary and the positions of the car and house at different locations compared to every other image.

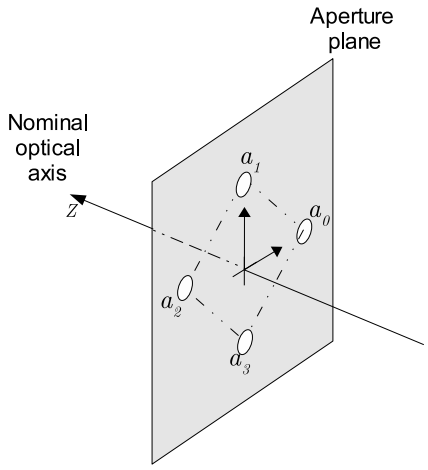


Figure 4.3: Illustration of the moving-aperture: a set of four different, off-center aperture positions in the path traced on the aperture plane. The aperture positions are at constant radius R from the aperture center O, which is also the origin of the global coordinate system.

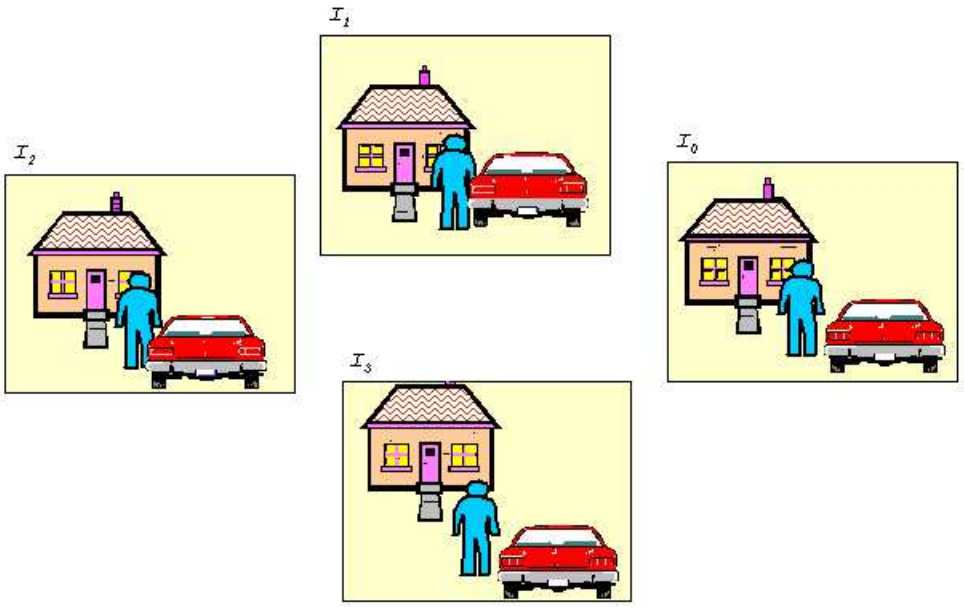


Figure 4.4: Illustration of a moving-aperture image sequence. Images I_0, I_1, I_2 and I_3 captured at the aperture positions a_0, a_1, a_2, and a_3 respectively, for the example scene. In the sequence, notice the displacement of image regions corresponding to objects not on the plane of focus.

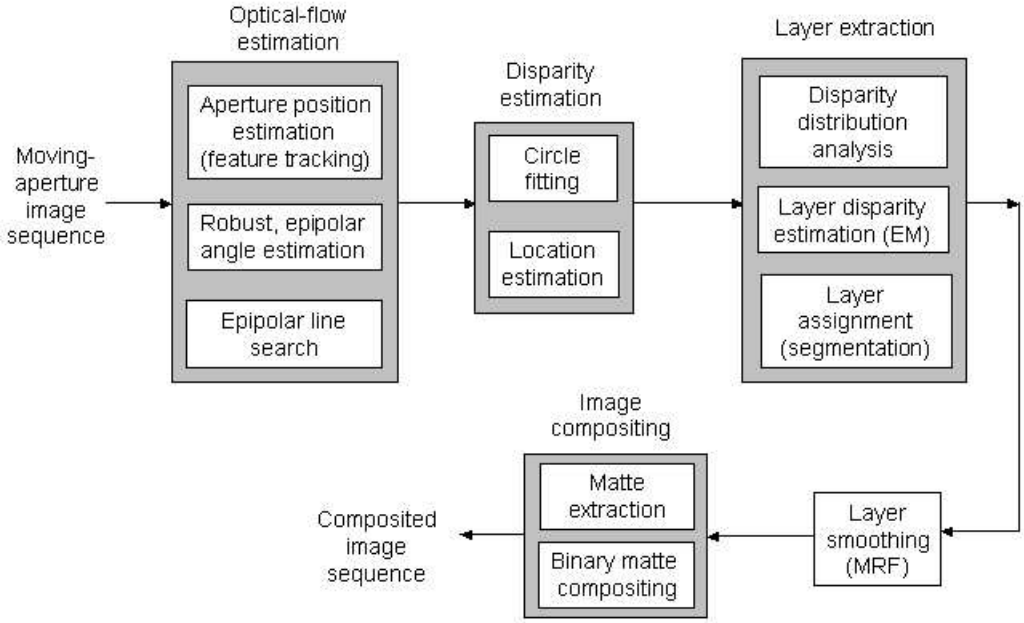


Figure 4.5: Methodology of layer extraction and image compositing using a moving-aperture lens.

4.4 Methodology

The layer extraction and compositing procedure developed here takes in a sequence of images from a camera using a moving-aperture lens and then analyzes the images to extract the objects in the scene based on their distances in scene, and assigns them to different planar 2D image layers. The entire process is unsupervised, i.e., requires no manual intervention, although the parameters in the analysis are configurable. The main stages in the layer extraction and image compositing method are

1. optical flow estimation,
2. disparity estimation,
3. layer extraction, and
4. image compositing.

This procedure identifying the key steps in each stage is summarized in the block diagram shown in Fig. 4.5.

4.5 Optical Flow Estimation

The moving-aperture lens induces parallax in various image regions depending on the 3D distance of an object corresponding to the image region. In our discussion, we will use the terms 'optical flow' and 'image motion' interchangeably.

4.5.1 General Approach

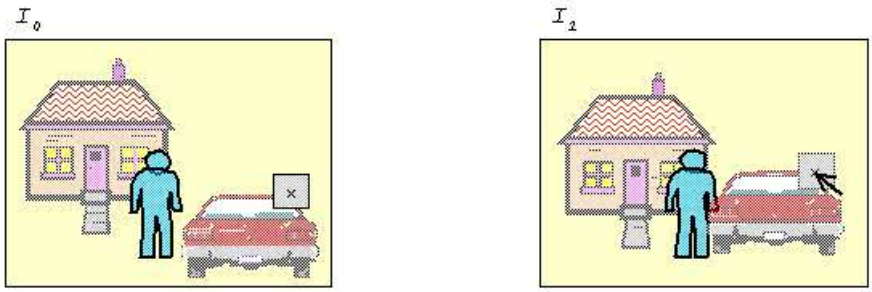
The optical flow constraint (2.7), introduced in Chapter 2, attempts to estimate two parameters of the optical flow, with an assumption that intensity in an image region where the optical flow is calculated does not change. But this is an under-constrained problem, because two parameters are to be estimated at each point and there is only one constraint equation. Therefore it is necessary to add another constraint to solve for the two parameters. Many estimation methods assume that gradients in the image are preserved. The gradient -based methods assume constant image gradients in a region of two images and use some local optimization method for tracking the displacement of a region from one image to the other. This tracking information gives the optical flow in the image windows.

A commonly used method for flow estimation, tracking, and image registration applications is the Lucas-Kanade technique [64]. It uses a finite, local neighborhood of an image point with a weighted, linear least squares based approach. The spatial support (or size of the window) is typically chosen as a square (5×5 or 9×9) for computational ease. The temporal support of the image window is limited to only one image. For tracking any point p, from image I_0 to image I_1, an image window I_{w0} is chosen from I_0 and a window I_{w1} of the same size from a neighborhood of p in I_1. The Lucas-Kanade algorithm minimizes the solution of

$$\begin{bmatrix} \sum I^x_{w0} I^x_{w0} & \sum I^x_{w0} I^y_{w0} \\ \sum I^x_{w0} I^y_{w0} & \sum I^y_{w0} I^y_{w0} \end{bmatrix} \begin{bmatrix} v^x \\ v^y \end{bmatrix} = \begin{bmatrix} I^t_{w0} I^x_{w0} \\ I^t_{w0} I^y_{w0} \end{bmatrix}$$

where $I^t_{w0} = (I_{w0} - I_{w1})$ is the difference in intensities between image windows, the superscripts of image windows indicate the directions of gradients or partial derivatives of the image window (not the location of a point in the image), and the parameters v^x and v^y represent the displacement (optical flow) of the image point p, along the x and y axes of the image respectively.

Estimating the components of optical flow at a position p, in an image gives motion vector components v^x and v^y for that point. The magnitude of flow (motion) can then be calculated as $|v| = \sqrt{(v^x)^2 + (v^x)^2}$



(a) (b)

Figure 4.6: Illustration of optical flow at an image point using a local neighborhood: (a) a local window in image I_0 captured at position a_0 of the aperture, (b) tracked position of the window in the image I_1 from adjacent aperture position.

and its direction $\angle v = \arctan\left(\frac{v^y}{v^x}\right)$, to represent the flow as a vector at the point.

In the example scene, an image window I_{w0} at a point p_0 in the image I_0 is shown in Fig. 4.6(a). This window is tracked to a point p_1 in image I_1, as shown in Fig. 4.6(b). The magnitude and direction of the flow vector u from the position in image I_0 is also indicated. The vector u indicates the displacement of point p_0 to p_1 from image I_0 to I_1 respectively.

The Lucas-Kanade method can be applied at every point in the image to estimate the optical flow in the entire image. This gives a dense, optical flow vector map V, which is of the same size as the original image I_0. It contains two components of the flow along x and y directions for every point in the image. (Note that v represents the optical flow at one point in the dense flow map V.)

The flow for every image point in I_0 can be estimated using the image pair $I_0 - I_1$, and this gives a dense map of the optical flow, V_{01}, where subscript indicates the image pair in sequence. A sparse version of this optical flow map for the example scene is shown in Fig. 4.7. The person in the scene is in focus and so the corresponding image points have zero displacement, while regions corresponding to objects not on focus plane have non-zero optical flow.

Following the same approach for every adjacent image pair in the sequence, optical flow maps for the entire image sequence can be obtained. Thus, we totally have a set of $N-1$ flow maps namely – $V_{01}, V_{12}, V_{23}, ..., V_{N-2,N-1}$ from the image pairs $I_0 - I_1, I_1 - I_2, I_2 - I_3, ..., I_{N-2} - I_{N-1}$ respectively. From these optical flow maps, it is possible to calculate the disparity of each point in the image. For the example scene, Fig. 4.8 shows the sparse optical flow maps of V_{12} and V_{23}, corresponding to the sparse points in Fig.

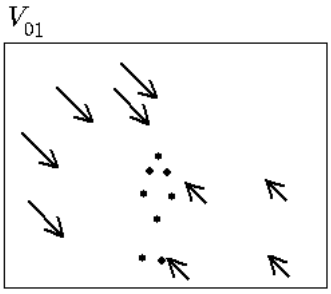
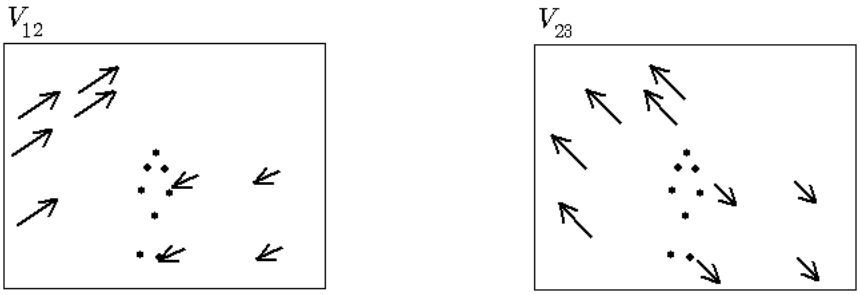


Figure 4.7: Illustration of sparse optical flow in V_{01}, estimated from images I_0 and I_1 in the example sequence. The vector magnitudes are not shown to scale.



(a) (b)

Figure 4.8: Illustration of sparse optical flow maps corresponding to Fig. 4.7 for the example scene. The vector magnitudes are not shown to scale.

4.7. These three maps V_{01}, V_{12} and V_{23} together give the displacement of image points in the entire sequence I_0, I_1, I_2 and I_3.

A main disadvantage of the Lucas-Kanade algorithm is its assumption of small pixel displacement in images (due to the first order Taylor series expansion in its formulation) [65]; therefore, it is not directly suitable for estimating large motion or displacement. This deficiency of the method is eliminated by using an image pyramid approach.

An image pyramid is stack made of multi-resolution images, with the bottom of the pyramid (*level* 0) being the original (raw) image. Let the size of this original image be M ($= M_x \times M_y$, i.e., M_x columns and M_y rows). The next image in the stack at *level* 1 is one-fourth the size of the image on *level* 0 (i.e, $\frac{1}{2}M_x \times \frac{1}{2}M_y$) and is a sub-sampled version of *level* 0, created using a Gaussian anti-aliasing filter. The other higher levels of the pyramid are constructed similarly. The height refers to the number of levels in a pyramid. As the height of a pyramid increases, the resolution of the images in each level decreases. The image size generally becomes small after level-2 and offers little advantage beyond this height. Figure 4.9 shows an

illustration of a two-level image pyramid corresponding to the example in our discussion. The original work by Burt and Adelson on Gaussian pyramids [66] explains more on the construction of image pyramids, and the anti-aliasing filter.

With the use of image pyramids, the Lucas-Kanade method can be applied to estimate large image motion. In the image pyramid based Lucas-Kanade method, with an initialization of zero flow, the optical flow at an image point is first calculated at the lowest image resolution (top most level). Then this estimate is used as an initial value in the next higher resolution, which gives a better estimate of motion than that from the previous level. Such a progressive approach eventually leads to the estimation of accurate optical flow at a point. With the addition of an interpolation operation in each step, subpixel level displacements can be estimated with high precision. When a dense optical flow map is required, the pyramid is traversed in a downward fashion - i.e., dense maps are estimated starting at the low resolution and these are then propagated to the next higher resolution, until the original image resolution is reached. In this research, the pyramidal implementation of Lucas-Kanade method in the OpenCV library [65] is used for optical flow estimation.

4.5.2 Epipolar Line Approach

From the optical flow maps, it can be observed that the direction of flow vectors correspond to the direction of aperture motion between adjacent positions on the aperture plane. If we compare the flow map v^{01} in Fig. 4.7 with the direction of aperture motion from a_0 to a_1, we can note that the directions are related to each other. Also, in Fig. 4.7, note the flow vectors in regions corresponding to the objects at opposite sides of the plane of focus. Although the flow vectors are in opposite direction with different magnitudes, the 180° difference in their directions is preserved in every flow map estimated in the image sequence, as can be seen in Fig. 4.8. These directions directly correspond to the direction of aperture motion between the positions where the images were captured. Therefore this information is highly valuable and so must be exploited during the optical flow estimation.

The knowledge of the aperture motion is not used in the general, pyramid implementation of the Lucas-Kanade method described earlier. Therefore exploiting this information in estimating optical flow will reduce the computational time taken and so can lead to a fast implementation.

Let us again use the example from Fig. 4.4 to discuss this approach. For $N = 4$, a possible set of four

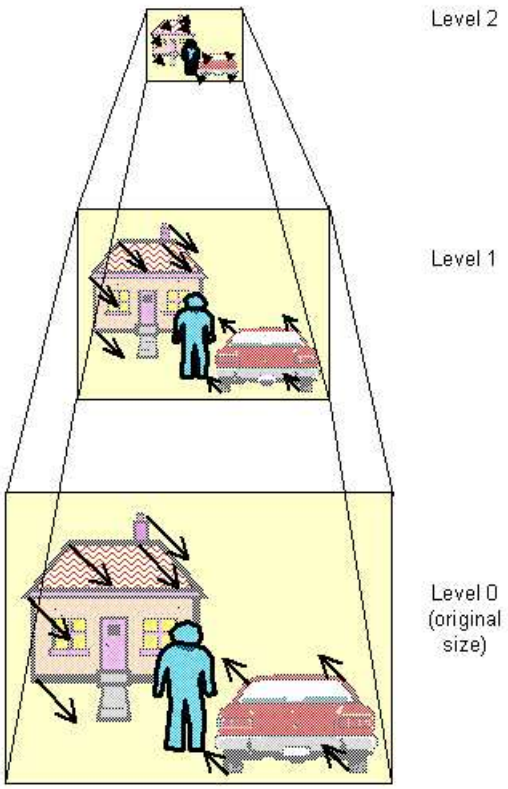


Figure 4.9: Illustration of a two-level image pyramid. The bottom of the pyramid stack is the original image. The upper levels in the pyramid are lower resolution images of their corresponding lower level.

positions of the aperture in the $X-Y$ plane are shown in Fig. 4.10 and this set forms a square on the aperture plane. The direction of aperture motion from a_0 to a_1 corresponds to the direction of the flow vectors in V_{01}. The vectors indicate that an image window (at an arbitrary position) in one image moved to a different position in the other image, *along the direction of the aperture motion.* This direction (or line) of aperture motion corresponds to the epipolar line in stereo (see Chapter 2). Therefore instead of estimating the optical flow (tracking) at each point from one image in the other image using no information of the aperture motion, it is wise to estimate the flow only in a specified direction along the epipolar line - especially, for a stationary scene, where the image motion is only due to the moving-aperture. The search process in the flow estimation can be limited to a short range of possible flow magnitude and so the computational time and processing can be drastically reduced.

Given only a sequence of images, it is not possible to guess the aperture positions corresponding to the sequence. The position of the aperture at rest, when it is not moving is the center of the aperture plane

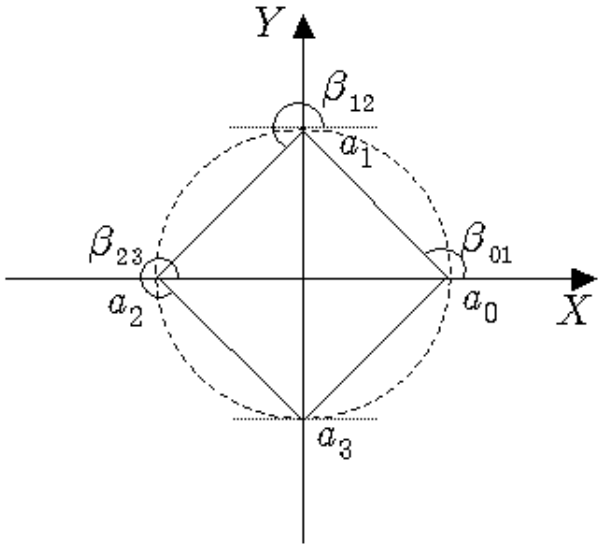


Figure 4.10: Illustration of an aperture-polygon, whose vertices represent the positions of the aperture a_0, a_1, a_2 and a_3 in the example. The angles $\beta_{01}, \beta_{12}$and $\beta_{23}$are the external angles of the edges with respect to the x axis.

and moreover, because the camera and moving aperture are independently controlled, the image capture can occur at any position of the aperture. Therefore the aperture positions are not exactly repeated in a sequence and so must somehow be inferred from the images. (The lens and camera used in this research operated indepentaly and hence the limitation in obtaining desired aperture positions existed.)

Estimating the aperture positions is possible by analyzing every adjacent image pair. In an image pair (say $I_0 - I_1$), we select few key points (say K; generally $K = 100$ and $K << M$) in the first image and find their corresponding optical flow by tracking these in the other image. These key points are chosen using the Shi-Tomasi features [67], which determines if an image point is a sufficiently good feature to track in images. An image point is chosen as a 'good feature', using the eigenvalues of the symmetric matrix

$$\begin{bmatrix} \sum I^x_{w0} I^x_{w0} & \sum I^x_{w0} I^y_{w0} \\ \sum I^x_{w0} I^y_{w0} & \sum I^y_{w0} I^y_{w0} \end{bmatrix}$$

calculated using an image window (typically 5×5) about the point. Large eigenvalues in the matrix represent corners or some other textured pattern, small eigenvalue pair indicate constant intensity windows, and a large and small eigenvalues represent a unidirectional texture pattern [67]. To track a point (or a window) in images, the ratio of eigenvalues of the matrix must be large and exceed a predefined threshold. This threshold is user-defined (typically chosen as 0.1 in our procedure). This threshold is only used for a preliminary setup

in the optical flow estimation step and does not determine the layers in the image later. Care has to be taken that the key points are chosen all over the image to represent every possible region in the image and are not crowded in a small area. It is possible to include a criterion on the Euclidean distance in the images between two good features such that the points are spread all over the image and are not concentrated in one specific area of the image. The chosen K image points are then tracked using an image window following the general approach described earlier, and thus giving the optical flow (hence displacement) for these points.

Consider the optical flow of the K points in one image pair. Each point has a displacement $\Delta x = v_x$ and $\Delta y = v_y$, along the x and y axes of the image respectively. These displacements, when plotted on the $\Delta x - \Delta y$ plane will appear as shown in the Fig. 4.11. In our example scene with a person in focus, two different objects on either side of the focus plane and a background, there will be distinct clusters of optical flow magnitudes and hence the plot will also contain at least three distinct cluster of points. The cluster of displacement points near the origin represents the displacement of image points corresponding to the image region of in-focus layer and the other clusters represent the optical flow image regions of the objects not on the plane of focus in the scene. Now, a line through these clusters can be fit using the data and it must pass through the origin. This line on the $\Delta x - \Delta y$ plane must pass through the origin, because a different result indicates a uniform global motion of objects in the scene, which contradicts our stationary scene assumption. Therefore the equation of this line representing the displacement direction of image points in the image pair is

$$\Delta y = m \Delta x \tag{4.2}$$

where m represents the slope of the line. The slope can be also written as

$$m = \tan \phi \tag{4.3}$$

where ϕ represents the angle epipolar line for the image pair.

A simple method to fit a line for given displacement data is to use a least-squares (LS) based linear regression. The LS approach for estimating the slope of the line is

$$m = \frac{\sum_{j=0}^{K-1} \Delta x_j \Delta y_j}{\sum_{j=0}^{K-1} (\Delta x_j)^2} \tag{4.4}$$

But least-squares based line fitting is known for its poor performance with outliers. Therefore, a robust statistical procedure for regression is preferred when the noise in given data is high. There exist various robust methods for linear regression [68, 69]. (Chapter 6 gives an introduction and a detailed discussion

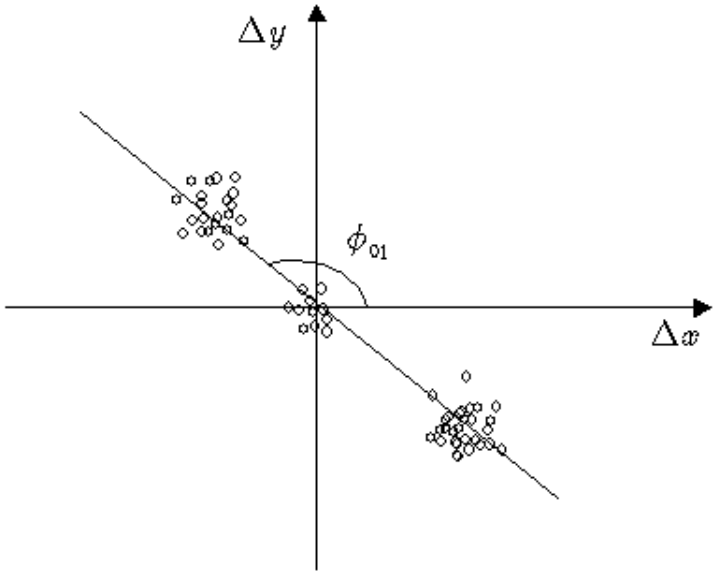


Figure 4.11: Graph illustrating the optical flow (displacement) of K points in the image pair $I_0 - I_1$ corresponding to the aperture positions a_0 and a_1 in the example.

on robust methods.) For our problem, we will use a maximum likelihood estimation (M-estimator) with an iteratively reweighted least squares (IRLS) approach for fitting the line. The IRLS procedure for fitting the line is as follows:

1. initialize the slope m with the LS estimate from (4.4)

2. for the line fit in this iteration using the current value of m, find residuals

$$e_j = \Delta y_j - m\Delta x_j, \qquad j = 0, 1, 2, ..., K-1$$

3. find robust standard deviation of the residuals, given by

$$\sigma = 1.4826\,\text{median}\,(|e_j - \text{median}(e_j)|)$$

   if $s = 0$, then $\sigma = \frac{1}{2}\,\text{mean}(e_j)$

4. *standardize* the residuals using

$$s_j = \frac{e_j}{\sigma}$$

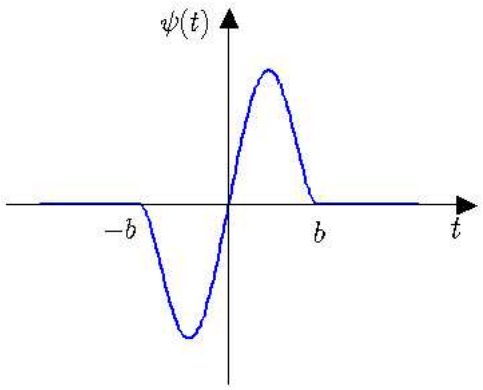


Figure 4.12: Tukey's influence function used in deriving weights for data.

5. find the robust weights to associate with each data,

$$w_j = \frac{\psi(s_j)}{s_j}$$

where $\psi(s_j)$ is the Tukey biweight estimator defined for any value t as

$$\psi(t) = \begin{cases} t\left[1 - (\frac{t}{b})^2\right]^2, & |t| < b \\ 0, & |t| > b \end{cases}$$

and b is a constant cutoff value (set to 4.6825, following the implementation for robust estimator in the Statistical Toolbox of Matlab). The influence function of the Tukey estimator, as seen in Fig. 4.12, sets the weight to zero for any *standardized residual* greater than the chosen cutoff.

6. recalculate the slope using the derived weights as,

$$m = \frac{\sum_{j=0}^{K-1} w_j \Delta x_j \Delta y_j}{\sum_{j=0}^{K-1} w_j (\Delta x_j)^2} \tag{4.5}$$

7. repeat steps (2) – (6) until the value of m converges (typically in less than five iterations).

From the estimated slope m, we can find the angle of the epipolar line using (4.3), which corresponds to an image pair.

Using the above IRLS procedure, the angles of the epipolar lines between every adjacent image pair are calculated. These estimated angles represent the external angle of the edges of the aperture-polygon, with respect to the x axis. When an initial position of the aperture is approximately chosen, the vertices

of a corresponding aperture-polygon can be estimated using these external angles. Ideally, this estimated aperture-polygon would be the ideal aperture-polygon (unknown) for N aperture positions. But due to noise in images, error in optical flow estimation and error in fitting a line, the estimated angles are not exactly the ideal external angles of the aperture-polygon. Therefore, a better result is obtained by optimizing the error in angles, identifying the best possible vertices and modifying the angle estimates which deviate from the expected value. For this, we use a global optimization procedure in which all the estimated angles are considered to find an aperture-polygon from every position, using the relative external angles of the edges.

The ideal external angles β, in a polygon, are shown in Fig. 4.13. For a regular polygon with N vertices, the *difference* in adjacent external angles between the adjacent edges ($\Delta\beta$) can be written as

$$\Delta\beta = \beta_{jk} - \beta_{ij} = \frac{2\pi}{N} \qquad i = 0, 1, 2, ...N-2; j = i+1; k = j+1$$

For example, in our example using $N = 4$, Fig. 4.3 shows the ideal aperture-polygon and its ideal external angles are represented as β_{01}, β_{12} and β_{23}. These three angles define the 4 vertices of the polygon. Let us say that the angles estimated using the IRLS procedure for each image pair are ϕ_{01}, ϕ_{12} and ϕ_{23}. Then the ideal difference in the external angles will be $\frac{\pi}{2}$ radians. Therefore the difference in estimated external angles should also be approximately $\frac{\pi}{2}$ radians and if this is not true, then there exists some error in the aperture positions which needs to be optimized. This optimization is performed as follows:

1. from the estimated external angles, choose one value, say $\phi_{k,k+1}$ (for some k, $0 \leq k \leq N-2$) and fix a vertex at a random point in the $x-y$ plane,

2. calculate the new external angles of the vertices in this polygon

$$\theta_{ij,k} = (j-i)\frac{2\pi}{N} + \phi_{k,k+1} \qquad i = 0, 1, 2, ..., N-2;\ j = i+1 \tag{4.6}$$

3. estimate the error in vertices of the N-sided polygon, with respect to the chosen vertex

$$e_k = \sum_{i=0;j=i+1}^{i=N-2} [|\tan\theta_{ij,k} - \tan\phi_{ij}| - \tan\beta_e] \tag{4.7}$$

   where β_e is an angular error 20% of $\frac{2\pi}{N}$ allowed to account for mechanical inaccuracies in the moving-aperture lens

4. find the sum of error e_k, for every $k = 0, 1, 2, ..., N-1$ which indicates the error in fitting an N-polygon

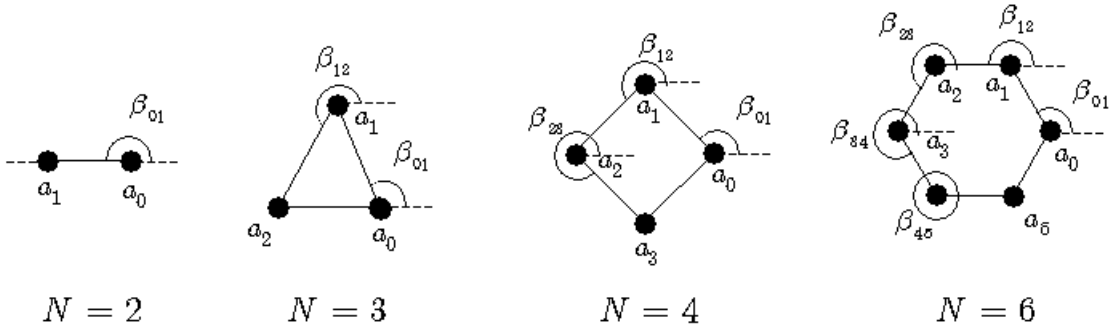


Figure 4.13: External angles between the edges in polygon.

with respect to each vertex

5. the k corresponding to the minimum e_k gives the best possible external angles for the given data and so calculate the ideal external angles of the polygon ϕ_{ij} corresponding to this best k using (4.6). These angles give the globally optimized external angles of the aperture-polygon.

It can be noticed from Fig. 4.10 that the external angle β_{01} corresponds to the angle of the epipolar line. Therefore optimizing the external angles optimizes the angles of epipolar lines between image pairs. So the pixel displacement in an image pair will be ideally along this line.

After epipolar directions have been estimated, then the computation time for optical flow estimation in an image pair can be drastically reduced by searching an image point only *along the epipolar line*. An image matching measure like *normalized cross-correlation* is used to find the displacement of an image window for tracking. The search range along the epipolar line can be restricted to a short range (say 4 pixels) on either side of the chosen image point. As the image displacement created due to the moving-aperture lens is very small and does not exceed this search range, the reduction in computation using epipolar lines is significantly better compared to the general approach described above or the brute-force block-matching method in Chapter 3.

4.6 Disparity Estimation

The optical flow estimation step produced $N-1$ optical flow maps $V_{01}, V_{12}, V_{23}, ..., V_{N-2,N-1}$ from the adjacent image pairs in the sequence. The flow vectors in these maps represent the motion of each image

point over the image sequence. Therefore, we have $N-1$ flow vectors for each image point, from each image pair in the sequence.

Now consider tracking the flow of an image point p_0 in I_0 over the image sequence. Estimating the optical flow between each $N-1$ image pairs gives $N-1$ flow vectors namely, $v_{01}, v_{12}, v_{23}, ..., v_{N-2,N-1}$. These vectors in turn can be thought of as vector edges of a polygon with one vertex at p_0. By understanding that these vectors represent the motion of an image point, these vertices can be used to find the $N-1$ vertices, $p_0, ..., p_{N-1}$ of a polygon (corresponding to N images in the sequence), with respect to p_0 in the $x-y$ plane. Let us refer this polygon as the 'pixel-polygon' (to differentiate it from the aperture-polygon).

4.6.1 Circle Fitting

Given N vertices representing the motion of an image point, a circumscribing circle for the pixel-polygon can then be identified. A circle fitting procedure (see Appendix A) with these N vertices can be used to estimate the radius d and center of the circle p_c which best fits the points $p_0, p_1, ..., p_{N-1}$. With respect to the location of an image point in the first image I_0, the center of the circle will be located off this point, due to disparity. When there is no parallax at a point, the radius of this circle will be zero.

In our work, the term 'disparity' is used to represent the displacement of an image point in a sequence, about its apparent center. It is noted that this definition is slightly different from the conventional understanding of the term. Here, the radius of the circle d corresponds to the disparity at the image point p_0.

In our example, the flow vectors at an image point are given by $V_{01}, V_{12}$and V_{23} by analyzing the image pairs $I_0 - I_1$, $I_1 - I_2$ and $I_2 - I_3$ in the sequence. These three vectors describe the three edges of a square as shown in Fig. 4.14. A circle circumscribing this pixel-polygon will represent the disparity of the image point, by its radius.

The radius of the circle d represents the magnitude of disparity (in pixels). As discussed earlier, the disparity is directly proportional to the distance of the object in the 3D scene corresponding to the image point. An image point representing an object located at a short distance from the plane of focus (either side) will have a small radius. Whereas, the radius of the circle fit for an image point corresponding to an object located at a far distance from the plane of focus will be relatively large. But the radius is a scalar value and so does not offer location information of an object in 3D, from the plane of focus from radius

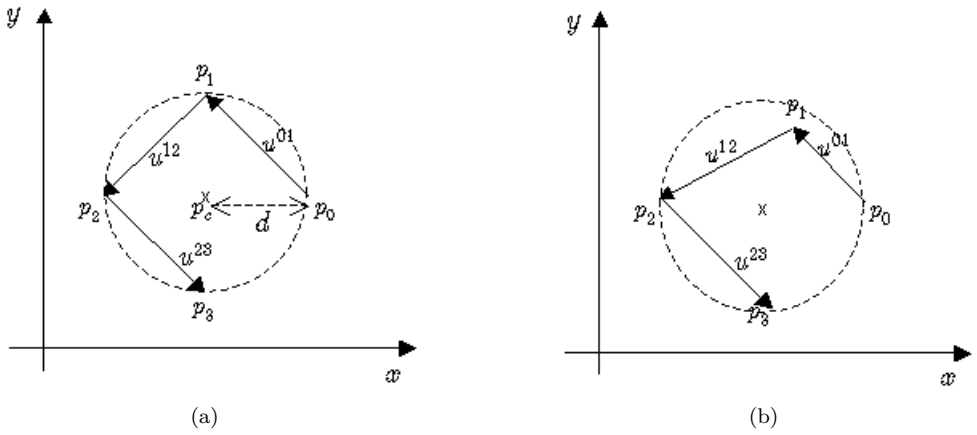


Figure 4.14: Illustration of a pixel-polygon, whose edges represent the flow vectors corresponding to an image point in the image sequence. The radius of the circumscribing circle corresponds to the disparity of the particular image point. (a) An ideal case where the optical flow vectors were noiseless and so a circumscribing circle is estimated with no error. (b) A typical case, where noise in optical flow vectors create problems in the circle fitting. The noise in flow vectors arise mostly due to intensity variations in the images.

alone. Therefore it is not possible to infer if an object is located in front or behind the plane of focus. If an object is at the plane of focus, then its disparity (and radius) is zero; so there is no ambiguity in its location information.

4.6.2 Relative Location Estimation

It is possible to obtain the relative location of an object from the plane of focus by analyzing the coordinates of the circle center and the image point in the $x-y$ plane. Because of the anti-phase relation in the moving-aperture image sequence, the estimated centers of image points corresponding to objects in either side of the plane of focus will be located in opposite directions on the $x-y$ plane, with respect to an image point. By determining the direction of offset between the image point and its corresponding circle center, we can group the image points into two classes - in front or behind the plane of focus. This grouping information can be included with the disparity, by attaching a sign (+1 or -1) to the radius information. There still exists an ambiguity in deciding if an image point is in front or behind the focus, because the decision is made with respect to the image point from the first image I_0 in the sequence. Unless the physical location of the aperture position a_0 is known for the image sequence, the grouping in two classes can at best only decide the possible locations but can not give the exact location in 3D, corresponding to an image point.

Again, using the dense optical flow maps $V_{01}, V_{12}, V_{23}, ..., V_{N-2,N-1}$, we can find the disparity correspond-

ing to each point in the image and thus obtain a dense disparity map $\mathcal{D}$. The disparity map is of the same size as the image and so contains signed radius values corresponding to each pixel in the image. This signed radius information represents the disparity of a corresponding object in 3D and its the relative location, from the plane of focus in the scene. Therefore by analyzing the magnitude and relative location of disparity values, the 2D depth layers in the scene can be extracted.

4.7 Layer Extraction

To extract depth layers in the image, we are interested in forming groups of pixels based on radius values. Then the average (mean) disparity of each layer can be estimated and then label image points with integer values thereby assigning a layer to the image point.

The dense disparity map $\mathcal{D}$ contains signed radius values for each image point. This map can be considered as a set of disparity values, i.e., $\mathcal{D} = \{d_1, d_2, ..., d_M\}$, where each element represents the disparity at a point in the image, and $M = M_x \times M_y$ gives the size of the image. A histogram of disparities from the map describes the number of image points against disparity values in the map. This histogram reveals the possible disparity layers in the image based on distinct peaks in the distribution.

4.7.1 Layer Estimation

The number of layers in a scene L, can be estimated by analyzing the histogram. A threshold in disparity bin separation of the possible layers (say 1 pixel) is used to eliminate layers which are almost equal in disparity. This refinement leads to an estimate of the number of layers in an image, provided they are sufficiently separated in the scene. This estimate inherently also gives the mean disparity of the identified layers in the image. Therefore, the analysis of disparity distribution from the map $\mathcal{D}$ gives an estimate of the number of layers in a scene (L) and also the average disparity values (μ_1, μ_2,..., μ_L) corresponding to these layers.

When the objects in a scene are sufficiently separated in depth (Z distance), then the histogram will also show clear peaks corresponding to these objects. To identify an image region as a layer in image, it is essential to set a threshold for the minimum image area (i.e., histogram count), say 5% of the entire image size, below which a peak in the disparity histogram will not be considered as a separate layer. The disparity bins whose count exceeds this minimum are considered as the disparity values corresponding to possible layers in the image. In most cases, this results in too many layers for a good representation of the scene.

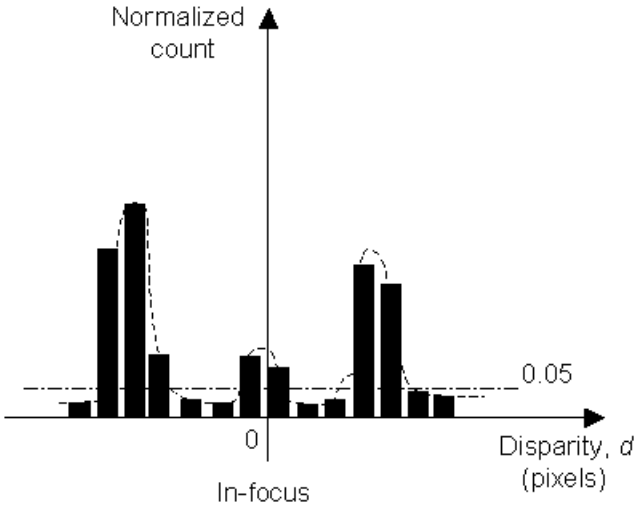


Figure 4.15: Illustration of a disparity histogram for the example scene. A peak near the disparity of zero corresponds to the image region which represents an object on the plane of focus. The peaks at non-zero disparities indicate the image regions corresponding the image regions representing those not on the plane of focus.

For our example scene with three distinct objects, the normalized distribution of disparities will appear as shown in Fig. 4.15. The normalization of bin counts in the histogram is with respect to the total size of the disparity map, M. The image region corresponding to an object on the plane of focus in the 3D scene has no disparity and so will lead to a peak in the histogram (at $d = 0$). However image regions corresponding to objects not on the focal plane will have signed disparities and so result in multiple peaks in the histogram. The threshold on the image size helps to identify the potential peaks in the histogram and the minimum disparity for layer separation helps to automatically estimate the number of layers in the image.

The histogram of disparities can be considered approximately as a probability density function and so can be represented using a multiple Gaussian mixture model (GMM). A GMM is generally used to model data whose probability distribution function (PDF) is a mixture of multiple Gaussian distributions each of which has its own mean and variance and are mixed in different proportions, which add up to 100%. In this analysis, the mean disparity of each layer μ, has to be estimated and also the corresponding standard deviation or spread σ, (not to be confused with the robust standard deviation used in Section 4.5.2) for each layer has to be calculated to enable in assigning a layer label for each image point. We are not interested in knowing the proportion η of individual mixtures, as it is not used in layer extraction. We can define the layer extraction problem using GMM as follows: given a set of M disparity data $\mathcal{D} = \{d_1, d_2, ..., d_M\}$,

whose distribution follows a Gaussian model, find the parameters of L mixtures, that would have most likely generated the disparity data. The three parameters of each mixture in the GMM are mean μ, standard deviation σ, and mixture proportion η. Let these parameters be represented as Θ and the distribution of disparities as $f(d;\Theta)$. So formulating the disparity distribution as a GMM is equivalent to writing its distribution as [70]

$$f(d;\Theta) = \sum_{j=1}^{L} \eta_j g(d;\mu_j,\sigma_j) \tag{4.8}$$

where

$$g(d;\mu_j,\sigma_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-(\frac{d-\mu_j}{\sigma_j})^2} \tag{4.9}$$

are the individual Gaussian distributions. The likelihood function in this case can be written as

$$\Lambda(\mathcal{D};\Theta) = \prod_{i=1}^{M} \sum_{j=1}^{L} \eta_j g(d_i;\mu_j,\sigma_j) \tag{4.10}$$

and therefore the estimation of parameters as a maximum likelihood is defined as

$$\hat{\Theta} = \arg\max_{\Theta} \Lambda(\mathcal{D};\Theta) \tag{4.11}$$

The log likelihood of (4.10) is

$$\lambda(\mathcal{D};\Theta) = \sum_{i=1}^{M} \log \sum_{j=1}^{L} \eta_j g(d_i;\mu_j,\sigma_j)$$

In our example, modeling the disparity histogram with a mixture of three Gaussian distributions will appear as shown in Fig. 4.16.

Expectation-Maximization (EM) [71] is an iterative technique which is often used to estimate hidden parameters in the distribution of any data. (An introduction to EM technique is given in [72, 70].) It is often used in estimating the model parameters, where the data is given but where the parameters of the function which generated the data are unknown (or hidden). In formulating the disparity distribution as a GMM, we are given the disparity data but are not given the parameters of each mixture. These unknown parameters are to be estimated and hence the EM technique suits the problem.

The number of layers estimated L and the disparity bin information of possible layers (μ_1, μ_2,..., μ_L) from histogram analysis in the earlier step are used to set the number of mixtures in the GMM and initialize the mean of each mixture respectively. The standard deviation and proportion of each mixture are initialized to random values and the EM procedure will then estimate the best values. The EM procedure calculates the conditional expectation of *a posteriori* density function of the parameters of each mixture (E-step) and

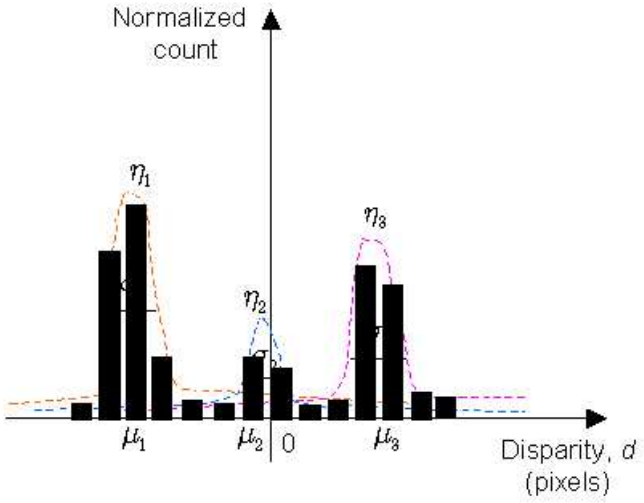


Figure 4.16: Illustration of a disparity distribution as a multiple Gaussian mixture model (GMM). The three parameters of each Gaussian distribution (mean, deviation and proportion of mixture) determine their contribution to the model.

then it estimates the parameters that maximize this *a posteriori* probability (M-step).

The conditional probability of choosing a mixture k from $\{1, 2, 3, ..., L\}$, given that data d_i was observed, is given by

$$p(k|i) = \frac{\frac{1}{\sqrt{2\pi}\sigma_k} e^{-\left(\frac{d_i-\mu_k}{\sigma_k}\right)^2}}{\sum_{j=1}^{L} \eta_j \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\left(\frac{d_i-\mu_j}{\sigma_j}\right)^2}}$$

and the probabilities of choosing a mixture given that its total must sum to 1 is

$$\sum_{k=1}^{L} p(k|i) = 1$$

Thus the E-step in the EM procedure estimates the conditional probabilities for every k in the model. These conditional probabilities can also be considered as normalized weights as the denominator is the sum of probabilities, which is 1. The M-step then estimates the parameters of the conditional probability which maximize the probabilities. These estimates of the parameters of the Gaussian mixtures are given by [70]

$$\eta_k = \frac{1}{M} \sum_{i=1}^{M} p(k \,|\, d_i) \tag{4.12}$$

$$\mu_k = \frac{\sum_{i=1}^{M} p(k|i)\, d_i}{\sum_{i=1}^{M} p(k|i)} \tag{4.13}$$

$$\sigma_k = \sqrt{\frac{\sum_{i=1}^{M} p(k|i) \parallel d_i - \mu_i \parallel^2}{\sum_{i=1}^{M} p(k|i)}} \tag{4.14}$$

where $k = 1, 2, 3, ..., L$. The EM procedure typically converges in less than 10 iterations and estimates the mean disparities μ_k, their deviation σ_k and their proportions η_k in the distribution. In the context of layer extraction, it therefore estimates the mean disparity (in pixels) and deviation of disparity for each of the L layers in the image from the data in dense disparity map $\mathcal{D}$.

4.7.2 Segmentation

In addition to estimating the mixture parameters, the EM procedure also identifies the best possible distribution from which each data value could have been obtained. Thus each data is "tagged" with a class label, corresponding to the mixture which possibly generated it. This "assignment" of a label to each image point therefore corresponds to a layer in the scene.

Let us represent the labelled map output from the EM step as $\mathcal{S}$, whose labels are $1, 2, 3, ..., L$. In this map $\mathcal{S}$, clusters of same labels form regions in the map, which directly correspond to a segmentation of the original image based on the distance in the scene. Therefore the map $\mathcal{S}$ represents the layer segmented result of the disparity map $\mathcal{D}$.

4.7.3 Layer Smoothing

The layer segmented map $\mathcal{S}$, extracted from the EM step generally contains noisy regions or 'holes'. So it should be refined by removing some of the holes and by smoothing region boundaries. Let this smoothed result be represented as $\mathcal{L}$. This smoothed layer map can be considered as a juxtaposition of regions from various layers in the image. Therefore the image layers of the scene under analysis can be obtained by separating the regions of each class in $\mathcal{L}$ as 2D image planes, based on their labels.

Towards this goal of smoothing the segmented map $\mathcal{S}$, a second order Markov Random Field (MRF) is used for a smoothing operation [73, 74]. (An introduction to MRF is given in [74].) We assume that the noise is only present in the labels themselves and shall assume that the number of layers was estimated correctly.

The smoothed layer map $\mathcal{L}$ is considered a Gibbs random field, where the region labels are random variables. The segmented map $\mathcal{S}$ is considered the observation $\mathcal{S} = \{s_1, s_2, s_3, ..., s_M\}$, obtained by adding an independent, Gaussian noise to the ideal map $\mathcal{L}$. Both fields are of size M (a $M_x \times M_y$ lattice). Let

t-12	t-9	t-6	t-7	t-11
t-10	**t-4**	**t-2**	**t-3**	t-8
t-5	**t-1**	**t**	**t+1**	t+5
t+8	**t+3**	**t+2**	**t+4**	t+10
t+11	t+7	t+6	t+9	t+12

Figure 4.17: Relative neighborhood of a position t, in a Markov random field. For a second order field, the neighborhood is defined as $\{t, t \pm 1, t \pm 2, t \pm 3, t \pm 4\}$.

s be the (known) segmentation label at some position t in the segmented map $\mathcal{S}$, and l be its ideal label (unknown) in the ideal layer map $\mathcal{L}$. (Here, s and t are not to be confused with the notations used in Section 4.5.2.) This is a classical image segmentation problem where the objective is to map the value s (similar to intensity in an image) to its ideal label l (segmentation label).

$$f : \mathcal{S} \rightarrow \mathcal{L} \Rightarrow \mathcal{S} = \{s_1, s_2, s_3, ..., s_M\} \rightarrow \mathcal{L} = \{l_1, l_2, l_3, ..., l_M\}$$

We model the MRF as a second order random field which is equivalent to using a 8-connected neighborhood on the random field.

Let the second order neighborhood of an ideal label at any position t in the ideal label field $\mathcal{L}$ be represented as $C_t = \{t, t \pm 1, t \pm 2, t \pm 3, t \pm 4\}$. This represents the 8-connected neighbors of the location t as shown in Fig. 4.17. By the Markov random field property of spatial locality, the label at any site t is determined only by its local neighborhood and is not influenced by the value at any other position in the field. Therefore the second order neighborhood information is sufficient to estimate the label l, at any site in the field.

The conditional *a posteriori* probability of observing a label l_k, given a label s at an arbitrary site t and the second order neighbors of C_t is [73]

$$p(l_k|s, C_t) = \frac{1}{T} e^{-U(l_k)} \tag{4.15}$$

where T is a normalizing constant defined as $T = \sum_{j=1}^{M} e^{-U(l_k)}$ and the energy function is

$$U(l_k, C_t, s) = \left[\frac{1}{2} ln(\sigma_k^2) + \frac{(s_j - \mu_k)^2}{2\sigma_k^2} + V_1(l_k) + V_2(l_k)\right] \tag{4.16}$$

The function $V_1(l_k)$ assigns a weight for a label and can be used to prefer any label while the function $V_2(l_k)$ compares the labels in a neighborhood and assigns a weight depending on the number of same or different labelled neighbors. These two terms represent the contextual information in the MRF and so without these

terms, the energy minimization would be independent of the neighborhood labels. For the layer smoothing problem, these functions are defined as [73]

$$\begin{aligned} V_1(l_k) &= 0 \\ V_2(l_k) &= \begin{cases} -1, & l_k = l_j \\ 1, & l_k \neq l_j \end{cases} \qquad l_j \in C_t,\ \tau > 0 \end{aligned}$$

Our goal is to obtain the maximum a posteriori (MAP) estimate which maximizes the conditional probability. But the search for a MAP estimate which minimizes the posterior energy is exhaustive and so is computationally expensive. Therefore various global and local optimization procedures are used for this optimization.

In our layer extraction problem, we use the Iterative Conditional Modes (ICM) [75] – a computationally feasible, local optimization procedure. The normalizing function in (4.15) is a constant and so can be ignored. Therefore, maximizing the conditional probability is equivalent to minimizing the energy function, (ignoring the negative sign in (4.15)). For the ICM smoothing of MRF, the procedure is as follows:

1. initialize the ideal layer map $\mathcal{L}$ with the segmentation output $\mathcal{S}$ from the EM step,
2. for each position $t = 1$ to M, in $\mathcal{L}$
   find the energy $U(l_k, C_t, s)$, for all integer labels $k = \{1, 2, 3, ..., L\}$ and update position t to the label l_k, which maximizes the energy and
3. repeat step (2) until the maximum number of allowed iterations or the number of label changes in $\mathcal{L}$ is negligible.

This optimization procedure is known to converge typically in less than 5 or 6 iterations. Because ICM is a local optimization procedure, the smoothed map $\mathcal{L}$ and hence the layers in the image are dependent on the initialization. In our case, as the MAP output of the EM step is used and this procedure typically converges in less than 10 iterations yielding highly satisfactory results.

From the final, smooth, integer labeled layer map $\mathcal{L}$, the regions are grouped based on the labels $\{1, 2, 3, ..., L\}$. Then L new images are created with a constant color (black or white). For each labeled region, the corresponding image regions from the image I_0 are extracted and assigned to the same locations in each of the new image. The L images with different regions of the image I_0 represent the L layers in the scene.

4.8 Image Compositing

After the layers in the scene are extracted from the intensity image I_0, compositing can be performed. In each layer, the region with intensity information represents the 'foreground' region of interest in a scene. It is necessary to extract extract a binary matte (α) for compositing.

For a layer of interest, the new image created is taken as the foreground image I_f, for compositing. Its corresponding binary matte α, is created with a transparency value of 1 in the regions corresponding to the 'foreground' and setting a transparency value of 0 (opaque) in the remaining regions of the matte. An arbitrary image I_b is chosen as the background in compositing.

The introduction to image compositing in Chapter 2 explained the process as an overlaying operation. Thus the composited output I_{comp} of the image I_f with the chosen background image $I_b$can be written as

$$I_{comp} = \alpha I_f + (1 - \alpha) I_b$$

The result of the entire process will be a composited image of an object from the scene with a different chosen background image.

Because the scene is stationary, the layer regions extracted from the moving-aperture image sequence do not change. Hence the image layers and the binary matte also do not change. Therefore, compositing a sequence of background images with the extracted foreground and matte also produces a sequence of composited images.

In summary, the described methodology estimated the optical flow in a moving-aperture image sequence, then calculated a dense disparity map for the scene. This disparity map was then analyzed to automatically extract layers in the scene. From these extracted layers, a layer of interest was manually specified. Its corresponding matte was automatically extracted and then the specified layer was composited on the chosen background image.

4.9 Moving-aperture Lens Model

The moving aperture lens can be used in a range estimation application. Towards this, it is necessary to analyze the mathematical model of the lens and derive a relation which will enable range estimation.

Consider the aperture in a moving-aperture lens, located at a distance R from the center of the aperture plane O and let the focal length of the lens be f. Although the aperture is located at an offset distance, light continues to reach the image plane of the camera, undergoing a deviation (bend) in its path due to refraction through the lens elements. Let the camera be focused on a point B, at a distance Z_0 from the aperture plane. Consider a point C located at a distance ΔZ_1 ($Z_1 \triangleq Z_0 + \Delta Z_1$), away from B. On the image plane, a clear, well defined (sharp) image of an object at point B will be formed at a distance f_0 from the aperture plane. Similarly, for an object at point C, its sharp image will be formed at a distance d_1 from the aperture plane, following the thin lens equation (the distance notation $d_1$should not be confused with the disparity representation in d as used earlier). This creates a slightly blurry image at the point G on the image plane, located at a distance r_1 from its center. This is illustrated in Fig. 4.9.

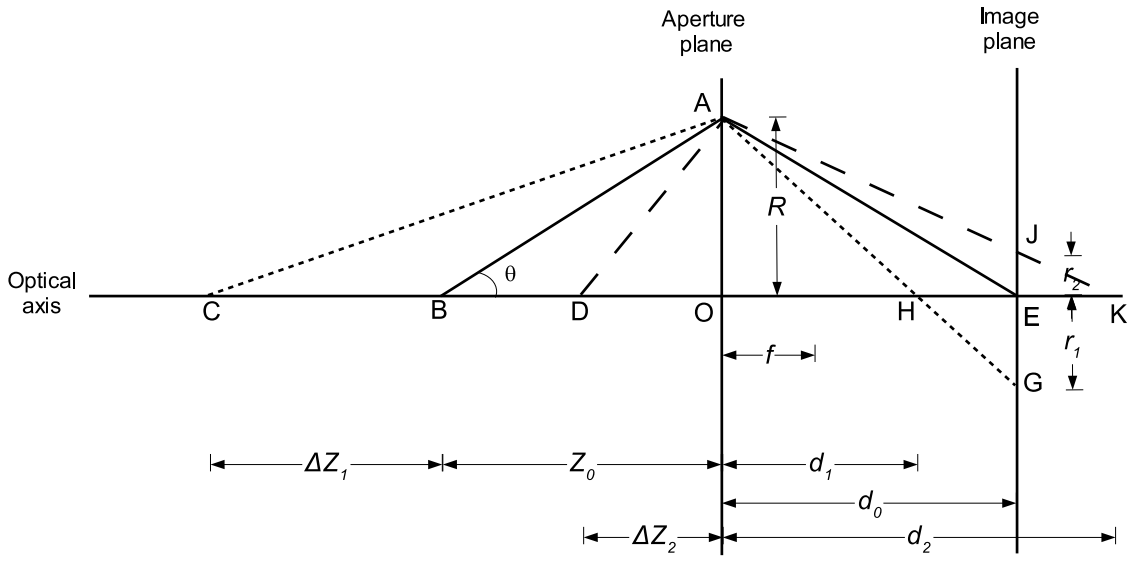


Figure 4.18: Moving aperture lens model. Here, the point A is the aperture of the lens at a distance R from the plane center O. B is the point of focus (and C is a point far away from focus), $E\,(D)$ is the point where a clear image corresponding to point $B\,(C)$ is formed at a distance d_0 (d_1) from the aperture plane. The focal length of the lens is f. At the point G, a slightly blurred image corresponding to an object at point C is formed, at a distance r_1 from the center of the image plane.

Often, in a real world application using this type of lens, the parameters that can be measured are the point of focus Z_0, distance of the far away point ΔZ_1 (or Z_1), aperture offset R and the focal length of the lens f. Now the objective is to find the image location distance r_1, corresponding to the point at a real

world distance Z_1, using this lens model. Note that r_1 is not the same as the distance in an image ρ_1, due to scaling of the image plane by imaging elements in the camera (like CCD size). But they are proportional and is related by a constant

$$\rho_1 = \kappa r_1 \tag{4.17}$$

where κ is a scaling constant for the image plane to image conversion. In some cases, the scaling constant is different for x and y axes of the image plane. In this work, we will assume that the scaling is uniform on both axes and therefore it allows us to write the relation in (4.17). The constant κ is called an *intrinsic parameter* of the camera and can be estimated from a simple camera calibration procedure using tools like Camera Calibration Toolbox in Matlab [76] (also see Appendix B). Because we are only interested in an intrinsic parameter, the calibration experiment can skip the lengthy procedure involved in estimating the extrinsic parameters and thus simplify the calculation of κ.

The thin lens equation, for the point B can be written as

$$\frac{1}{Z_0} + \frac{1}{d_0} = \frac{1}{f} \tag{4.18}$$

which gives the relation

$$d_0 = \frac{f\, Z_0}{Z_0 - f} \tag{4.19}$$

For the point C, we can write

$$d_1 = \frac{f\, Z_1}{Z_1 - f} \tag{4.20}$$

From $\triangle$OAH and $\triangle$HEG of Fig. 4.9, the following can be written

$$\begin{aligned} \frac{OA}{OB} &= \frac{EG}{DE} \\ \Rightarrow \frac{R}{d_1} &= \frac{r_1}{d_0 - d_1} \end{aligned} \tag{4.21}$$

Rewriting (4.21),

$$r_1 = R\left[\frac{d_0}{d_1} - 1\right]$$

and then substituting (4.19) and (4.20), we get

$$r_1 = R\left[\frac{f\, Z_0}{(\, Z_0 - f\,)} \frac{(\, Z_1 - f\,)}{f\, Z_1} - 1\right] \tag{4.22}$$

After simplifying, we get

$$r_1 = \frac{R\,f}{(Z_0 - f)}\,\frac{(Z_1 - Z_0)}{Z_1} \tag{4.23}$$

But we know that $(Z_1 - Z_0) = \Delta Z_1$ and therefore

$$r_1 = \frac{R\,f}{(Z_0 - f)}\,\frac{\Delta Z_1}{Z_1} \tag{4.24}$$

This relation gives the distance r_1 on the image plane corresponding to the distance Z_1 from the aperture plane and it shows the proportionality relation of its distance on the image plane. In other words, 4.22 indirectly gives the offset distance where an image corresponding to the object at C is produced.

Consider the other possible case, where this image displacement r_1 is known and the real world distance Z_1 is to be estimated, as in a range estimation application. Rewriting (4.23) gives

$$\begin{aligned} r_1 &= \frac{R\,f}{(Z_0 - f)}\left(1 - \frac{Z_0}{Z_1}\right) \\ \frac{Z_0 - f)\,r_1}{R\,f} &= 1 - \frac{Z_0}{Z_1} \qquad (4.25) \\ \frac{Z_0}{Z_1} &= 1 - \frac{(Z_0 - f)\,r_1}{R\,f} \qquad (4.26) \\ &= \frac{R\,f - r_1(Z_0 - f)}{R\,f} \\ \Rightarrow Z_1 &= \frac{R\,f\,Z_0}{R\,f - r_1(\,Z_0 - f\,)} \qquad (4.27) \end{aligned}$$

This relation gives the real world distance of an object, when the location offset in the image plane r_1 (or in the image ρ_1) is known. This result agrees with similar, independent work by Adelson [77] and by Rohaly and Hart [78].

An interesting corollary arises from the above derivation. Suppose two points in the real world are at known distances Z_1 and Z_3 (where $Z_3 > Z_1$). Let their corresponding image location offsets be r_1 and r_2 ($r_3 > r_1$) respectively. To be able to resolve these two offsets on the image plane, there exists a condition on the real world distances, which can be derived as follows. Here $Z_3 = Z_0 + \Delta Z_3$ from which an equation for r_3 can be written similar to (4.22). Therefore

$$\begin{aligned} r_3 - r_1 &= \frac{R\,f}{(Z_0 - f)}\left[\frac{(Z_3 - Z_0)}{Z_3} - \frac{(Z_1 - Z_0)}{Z_1}\right] \\ \Rightarrow r_3 - r_1 &= \frac{R\,f}{(Z_0 - f)}\left[\frac{Z_0\,(Z_3 - Z_1)}{Z_1 Z_3}\right] \end{aligned} \tag{4.28}$$

As $Z_1 = Z_0 + \Delta Z_1$, (4.28) becomes

$$r_3 - r_1 = \frac{R\, f\, Z_0}{(Z_0 - f)} \left[\frac{(Z_0 + \Delta Z_3) \; - \; (Z_0 + \Delta Z_1)}{Z_1 Z_3} \right]$$

$$r_3 - r_1 = \frac{R\, f\, Z_0}{(Z_0 - f)} \left[\frac{(\Delta Z_3 - \Delta Z_1)}{Z_1 Z_3} \right] \tag{4.29}$$

which can be rewritten as

$$(\Delta Z_3 - \Delta Z_1) = \frac{(Z_0 - f)}{R\, f\, Z_0}\, Z_1 Z_3\, (r_3 - r_1) \tag{4.30}$$

In most cases, the focal length of the lens is very small compared to the distance of objects, i.e., $Z_0 \gg f$ and so $(Z_0 - f) \approx Z_0$. Therefore we can write

$$(\Delta Z_3 - \Delta Z_1) \cong \frac{1}{R\, f}\, Z_1 Z_3\, (r_3 - r_1)$$

From this we can write a relation for the relative separation distance between two objects based on their image plane displacements as

$$r_3 - r_1 \propto \left[\frac{(\Delta Z_3 - \Delta Z_1)}{Z_1 Z_3} \right] \tag{4.31}$$

which signifies the proportional relation between object distances and their relative image plane displacements. The larger the separation between two objects $(\Delta Z_3 - \Delta Z_1)$, the larger is $(r_3 - r_1)$ and so is easy to distinguish two distance planes on the image (plane) as two different layers.

When $Z_3 < Z_1$, then $\Delta Z_3 < \Delta Z_1 \Rightarrow r_3 < r_1$ and so leads to

$$r_1 - r_3 \propto \left[\frac{\Delta Z_1 - \Delta Z_3}{Z_1 Z_3} \right] \tag{4.32}$$

which still preserves the underlying relation.

Let Δr_{min} be the minimum noticeable image plane displacement. Therefore from (4.30) we find that the distance separation in the scene must satisfy the condition

$$(\Delta Z_3 - \Delta Z_1) \geq \frac{(Z_0 - f)}{R\, f\, Z_0}\, Z_1 Z_3\, \Delta r_{min} \tag{4.33}$$

If this condition is not satisfied, then two distance layers can not be distinguished from the image analysis.

The theoretical relations derived here suggests that the moving-aperture lens can be used in passive range applications. Therefore the images from the moving-aperture lens can be used to estimate the relative locations and distances of objects in the 3D scene, with the knowledge of a few camera parameters. The

absolute locations of objects in the scene can be calculated, if Z_0 is known. Therefore this range estimation technique is different from that of a triangulation based method or a focus adjustment method suggested in [79].

4.10 Summary

This chapter explained the procedure for layer extraction and image compositing using the moving-aperture lens and also presented a mathematical model of the moving-aperture lens. With the knowledge of the distance of the focal plane, real world distances of objects in a scene can be determined. In the absence of the focal distance information, relative distances of objects with respect to the plane of focus can be estimated. As already mentioned, information on the initial position of the aperture corresponding to the first image in a sequence is necessary to remove the ambiguity in locations of objects with respect to the plane of focus. The mathematical model of the lens helps to determine the relative distances of objects at different distances in the scene and with the knowledge of the focal distance, passive range estimation is possible using the lens.

Chapter 5

Robust Circle Fitting

The optical flow estimated in moving-aperture image sequences using a block-matching technique in Chapter 3 produces highly noisy results due to image quantization and intensity values. The estimation of disparity (using circle fitting) from such noisy data is very error prone and the parameters estimated do not agree to the values expected from a priori knowledge of the image sequences. Therefore a robust procedure is necessary to estimate the circle parameters from noisy data. This chapter presents an introduction to robust methods used in statistics for parameter estimation in presence of noisy observations, generalized maximum-likelihood estimators. Then a robust circle fitting procedure for circle fitting is described. This method is simple yet produces good estimate of parameters in the presence of noisy data.

The notations used in this chapter bear no relation to variables used in other chapters. Also, a bold typeface notation indicates a vector (or a matrix).

5.1 Background

The problem of fitting a circle or an ellipse to a set of 2-dimensional (2D) points arises frequently in many disciplines related to image analysis. Most approaches minimize an error criterion that is based on an implicit representation of the curve, and assume that a high level of accuracy can be achieved for the resulting fits. However, in spite of the ubiquity and seeming simplicity of the problem, most existing circle fitting methods (e.g., [80, 81, 82, 83, 84, 85]) are not robust in the sense that they are easily affected by the presence of outliers. The most common approaches of ellipse fitting are known as "algebraic" methods, because they minimize an error criterion based on an implicit representation of the ellipse (e.g., [86, 83, 87, 88]). This

traditional method allows using closed-form expressions to obtain convenient solutions, although the error criterion used in the minimization has no physical meaning. The other approach, known as orthogonal (or "geometric") methods, attempt to minimize the Euclidean distances from an ellipse to the set of given points [89, 90]. Often the results obtained using these methods are more intuitively appealing than those obtained using traditional methods. In case of circle fitting, algebraic and geometric methods minimize the orthogonal (radial) distance and so there exists no difference in fitting approaches.

Common approaches to circle fitting rely on the minimization of an objective function that is based on the sum of squared errors between the given points and the computed circle model. Because the fitting problem is non-linear, the generalized least-squares (GLS) techniques accomplish the minimization with an iterative search, often using the Gauss-Newton or Gauss-Seidel methods [82, 91]. The GLS technique has also been used to fit other conic sections, such as ellipses and hyperbolas [90]. Although the GLS technique performs well in many situations, its accuracy tends to degrade drastically when one or more outliers are present in the data. Furthermore, this approach perform poorly even for the noiseless case when most of the given points are not evenly distributed as when clustered on one side of the circle.

This study differs from other previous work, in that specifically very small data sets are considered for circle fitting (10 points or fewer) with outliers. Except for the least-median approach, which is highly robust, previously published approaches either require large data sets, or do not include severe outliers in the published test cases. This work demonstrates that relatively simple, robust statistical methods can be used for circle-fitting, whose results are superior to those from traditional approaches, when severe noise is present. Other circle-fitting approaches based on moment analysis and on the Hough transform exist, but they are not considered here in this study.

5.2 Robust Techniques - A Review

5.2.1 Overview

The concept of robustness implies the ability of a method to continue its good performance when few unfavorable data are introduced among the existing set of data and robustness is used widely in parameter estimation problems. In an estimation problem, there will exist many good data points and some unfavorable data points. In most cases, the presence of these unfavorable data points can skew or bias, or worse, introduce error in estimates of parameters. Therefore identifying these unfavorable data is a major step in robust

statistics and so considerable attention is devoted to this step in robust estimation procedures. A good review of robust statistical techniques is given in [69, 68, 92, 93].

In a regression problem, the spatial distribution of given data points change (or influence) the value of parameters estimated, depending on the nature of the data distribution. If the location of a point in the distribution is such that it significantly changes the estimate of regression parameters, then it is called a *leverage point*. A leverage point can be good or bad, depending on its location with respect to the other data. A bad leverage point is unfavorable for parameter estimation and it is usually located far away from the majority of the distribution i.e. it *outlies* the bulk of the data. The presence of such an 'outlier' implies that the parameters estimated using this data are prone to a large error in estimates.

When the number of outliers is more than a certain fraction of the given data, any regression method used to estimate the parameters (also called an estimator) fails to produce a good estimate . There exists a limit beyond which an estimator can not estimate the parameters reliably. This limit on the number of data points for a reliable estimate of parameters is called a *breakdown point* γ, and it refers to the number of outliers or fraction of corrupted data that an estimation procedure can withstand in performance and still produce a good (robust) estimate [94]. The breakdown point is a measure of robustness of an estimator and so a high value of γ implies that the method is very robust.

The commonly used sample mean is the best example of an estimator, as it "estimates" the mean of given data. The least-squares approach is also a good example of an estimation procedure. But for these commonly used methods, the breakdown point is $\gamma = 0\%$, which implies that the presence of a single outlier will lead to a poor estimate of the parameter. A method with a breakdown point of 30% indicates that it can estimate the parameters reasonably accurate until 30% of the given data become outliers. Beyond this level of corruption, the estimation method can not estimate the parameters reliably. The sample median, another commonly used estimator has a breakdown point of $\gamma = 50\%$, which indicates that the method can reliably estimate the median even when the data contains 49% of outliers. The estimators with such large values of γ are called *high-breakdown-point* estimators. A contamination of more than 50% of the data is seldom useful for estimators, because it is the majority of the data that determines the parameter estimate.

5.2.2 Outlier Identification

Any regression method of practical importance must deal with noise and outliers in the given or measured data. The important step in any robust method is the identification of outliers. Consider a general case with a set of M observations in N dimensions: $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \ ... \ , \mathbf{z}_M\}$, and $\mathbf{z}_i \in \Re^N, \ i = 1, 2, 3, ..., M$. Let z_{ij} represent the i^{th} observation in the j^{th} dimension.

For a normally distributed multivariate data, the maximum-likelihood estimate of the mean of the data is given by its sample mean, $\boldsymbol{\mu}_s$. The sample mean is an affine equivariant and statistically efficient estimator. But given its breakdown point ϵ =0%, any single outlier in the data can skew the estimate of the mean drastically. In spite of this disadvantage, the maximum-likelihood type estimator is commonly used for reasons of tradition and simplicity.

A robust method must identify outliers and reduce the influence of the outliers on the estimation procedure. In a way, this outlier identification step can be considered as a 'weighting' procedure. Various methods are used to identify outliers in a given data. One widely used technique is based on the Mahalanobis distance, defined as

$$d_i = \sqrt{(\mathbf{z}_i - \boldsymbol{\mu}_s)^T \, \mathbf{C}^{-1} \, (\mathbf{z}_i - \boldsymbol{\mu}_s)}$$

where $\mathbf{C}$ is the covariance matrix of the given data. For a normally distributed multivariate data, the squared Mahalanobis distances approximately follow a $\boldsymbol{\chi}^2_t$ distribution with t degrees of freedom. The observations which correspond to a very large squared Mahalanobis distance are considered to deviate from the general distribution and so can be identified as outliers. This threshold or cutoff distance is usually taken from the 97.5% quantile of the $\boldsymbol{\chi}^2$ distribution. Classical techniques identify outliers based on this idea, tagging any data whose distance is larger than the assumption (cutoff), as an outlier. Some other outlier identification techniques are the minimum covariance determinant (MCD) and robust distance (RD) measures [95]. Once the outliers are identified, their influence in an estimation method can be eliminated.

5.2.3 Generalized M-estimators

The parameters of a model are generally estimated using a maximum-likelihood approach. Consider a regression model

$$\mathbf{d} = f(\mathbf{p}) + \mathbf{e} \tag{5.1}$$

where $\mathbf{d} = [d_1, d_2, d_3, ..., d_M]$ are measured values, $f(\mathbf{p}) = [f_1(\mathbf{p}), f_2(\mathbf{p}), f_3(\mathbf{p}), ..., f_M(\mathbf{p}),]$ describes the model in K parameters $\mathbf{p} = [p_1, p_2, p_3, ..., p_K]$. In this model, the measurements $\mathbf{d}$ (related to the observations or data $\mathbf{Z}$) and the errors $\mathbf{e}$ in the model are known and the model parameters $\mathbf{p}$ are unknown. The objective is to determine the model parameters which optimally fit the given data.

The residuals in fitting a model to the measurements are given by

$$r_i = d_i - \hat{d}_i = d_i - f_i(\hat{\mathbf{p}}) \qquad i = 1, 2, 3, ..., M$$

where $\hat{\mathbf{p}}$ represents the estimated parameters. The estimated parameters $\hat{\mathbf{p}}$ are ideally expected to be the same as the true parameters of the model $\mathbf{p}$. This occurs when the residuals are zero or close to zero. Therefore an estimation procedure minimized the sum of the residuals in determining the model parameters.

The objective function for the M-estimator is

$$J = \sum_{i=1}^{M} \rho\left(\frac{r_i}{s}\right) \tag{5.2}$$

where $\rho(r_i)$ is some function of the residuals. The solution of this objective function is obtained as

$$\frac{\partial J}{\partial \mathbf{p}} = 0 \Rightarrow \sum_{i=1}^{M} \frac{1}{s}\, \psi\left(\frac{r_i}{s}\right) \frac{\partial r_i}{\partial \mathbf{p}} = 0 \tag{5.3}$$

where $\psi\left(\frac{r_i}{s}\right) = \frac{\partial \rho(r_i/s)}{\partial r_i}$ is widely used in robust statistical methods to determine the influence of a data on an estimation procedure and s is a scaling term used for standardization (becomes variance, when errors in model are independent and normally distributed).

There exists a large class of influence functions [96, 89]. For the least-squares method, $\psi(r_i) = r_i$, with no scaling (or s=1), which weighs all the data equally. So the presence of even a single outlier can easily influence the estimation procedure and skew the parameter estimate in a least-squares based approach. Therefore this explains the non-robust nature of the least-squares method.

The robust Huber $\psi(.)$ function is given by

$$\psi\left(\frac{r_i}{s}\right) = \begin{cases} \frac{r_i}{s} & , \left|\frac{r_i}{s}\right| \leq k \\ k\, sign\left(\frac{r_i}{s}\right) & , \left|\frac{r_i}{s}\right| > k \end{cases} \tag{5.4}$$

where k is a tuning constant. The value of k is generally recommended to be 1.345 for a Gaussian model

[89]. In this work, we are using the Huber skipped-mean defined by

$$\psi\left(\frac{r_i}{s}\right) = \begin{cases} \frac{r_i}{s} & , \left|\frac{r_i}{s}\right| \leq k \\ 0 & , \left|\frac{r_i}{s}\right| > k \end{cases} \tag{5.5}$$

By using weights for the data in the objective function of a M-estimator, it can become a generalized maximum-likelihood estimator (GM estimator). The solution of its objective function is given by

$$\sum_{i=1}^{M} w\left(\frac{r_i}{s}\right) \frac{r_i}{s} \frac{1}{s} \frac{\partial r_i}{\partial \mathbf{p}} = 0 \tag{5.6}$$

5.2.4 Iterative Reweighted Least Squares

The relation in (5.6) is generally a set of nonlinear equations and so an iterative procedure is required to find its solutions [96]. Towards this, we use the iterative reweighted least-squares (IRLS) technique proposed by Beaton and Tukey [97] to estimate the parameters of the solution.

Assuming an initial value of the parameters $\mathbf{p}$ are available, an iterative procedure for estimating the parameters of (5.6) in a matrix notation is

$$\mathbf{p}^{(t+1)} = \mathbf{p}^{(t)} + [\mathbf{F}^T \mathbf{W}^{-1} \mathbf{F}]^{-1} \mathbf{F} \mathbf{W}^{-1} \mathbf{r}^{(t)} \tag{5.7}$$

where $\mathbf{p}^{(t)}$ is the estimated parameter vector at the t^{th} iteration, the Jacobian $\mathbf{F} = \frac{\partial r_i}{\partial \mathbf{p}} = \frac{\partial f(\mathbf{p})}{\partial \mathbf{p}}$ and the weights are written as

$$\mathbf{W} = \begin{bmatrix} w\left(\frac{r_1}{s}\right) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & w\left(\frac{r_M}{s}\right) \end{bmatrix}$$

The choice of $\psi\left(\frac{r_{i1}}{s}\right)$ in the weights $w\left(\frac{r_1}{s}\right)$ make the IRLS procedure robust to outliers in data.

5.2.5 Other Robust Methods

Two popular methods used in robust regression are the least-median-of-squares (LMedS) and RANSAC. The LMedS method [98] attempts to minimize the median of squared errors instead of the mean of the errors, as in a least-squares method. In this method, a subset of data is drawn from the given data and the model parameters are estimated. The median of the residuals of all data with respect to the estimated parameters is noted and the procedure is repeated for another subset of the given data. When all possible subsets of

the given data has been used in the estimation, the minimum of the median of the residuals is chosen as the best possible solution and so its corresponding parameter estimates are the best estimates of the model. The exhaustive combinatorial search for the best estimate is computationally intensive and so is often replaced by a Monte-Carlo type random sample drawings from the given data [89]. The LMedS method has a high breakdown point of 50%, which is the maximum possible and therefore is highly robust to noise. It is able to estimate the parameters in a regression model even when nearly half the fraction of given data are noisy or outliers.

Random Sample Consensus (RANSAC) introduced by Fischler and Bolles [99] is a popular method for regression in the field of computer vision because of its ability to estimate parameters in presence of severe noise. In this method, a random subset of the given data is chosen to fit the desired model. The parameters of the model are estimated using this random subset. The error in fitting this model to other data, which do not belong to the chosen subset is then calculated. If the number of data confirming to this fit model is greater than a minimum, then it is assumed that the parameters of the model were estimated and the procedure is terminated. If the number of data agreeing to the estimated model parameters is less than the minimum, another random subset from the given data and the procedure is repeated until the number of data agreeing to the estimated model is more than the minimum. The minimum number of data used to declare an estimate is chosen by the user. Thus the performance of RANSAC is dependent on this minimum threshold chosen in the estimation procedure. In some real problems, a good threshold can be hard to pick. The RANSAC procedure is widely used in the computer vision problems such as scene analysis where the objective is to find an epipolar line in image pairs for determining the fundamental matrix.

5.3 Circle Fitting

The main inspiration for this investigation of robust circle fitting methods came from the circle fitting step used in the earlier optical flow estimation approach (see Chapter 3). The data obtained from block-matching was very noisy and hence pathological. A least-squares based circle fitting method performed poorly and the estimates did not agree with the expected results. Prior information of the images indicated that no circle estimated would have a radius greater than 1 pixel radius. But as the available data was pathological, non-robust least-squares method did not consistently produce good results which agreed with this expectation. A significantly large noise present in the data and the unusual spatial distribution of data were the reasons for the failure of non-robust method. In many cases the points were not unique due to relatively coarse

spatial quantization of the image. The key parameter to estimate in the image segmentation application is the radius and it was known that the estimated circle must fit between the two extreme data along any direction.

5.3.1 Problem Statement

An implicit representation of a circle is given by

$$(x - X_c)^2 + (y - Y_c)^2 = R^2 \tag{5.8}$$

here a parameter vector $\mathbf{p} = [\hat{R} \quad \hat{X}_c \quad \hat{Y}_c]^T$, uniquely determines a set of points (x, y) in the plane that lies on the circle. (We adopt the convention that a bold typeface indicates a vector quantity.) The goal of the circle-fitting problem can be stated as follows: Given a set of points $\{(x_i, y_i) \,|\, i = 1, 2, 3, ..., M\}$ for $M \geq 3$, determine the optimal parameter vector $\hat{\mathbf{p}} = [\hat{R} \quad \hat{X}_c \quad \hat{Y}_c]^T$that corresponds to the circle of best fit (here, $K = 3$).

5.3.2 Robust Circle Fitting

Using the regression model in (5.1), we can write

$$\begin{aligned} d_i &= x_i^2 + y_i^2 \qquad\qquad i = 1, 2, 3, ..., M \\ f_i(\mathbf{q}) &= R^2 - X_c^2 - Y_c^2 + 2(X_c x_i + Y_c y_i) \\ &= [1 \quad x_i \quad y_i] \, [(R^2 - X_c^2 - Y_c^2) \quad 2X_c \quad 2Y_c]^T \\ &= \mathbf{h}_i^T \, \mathbf{q} \end{aligned}$$

where $\mathbf{h}_i = [1 \quad x_i \quad y_i]^T$ and $\mathbf{q} = [q_1 \quad q_2 \quad q_3]^T = [(R^2 - X_c^2 - Y_c^2) \quad 2X_c \quad 2Y_c]^T$. The parameters in $\mathbf{q}$ are related to the three circle parameters $\mathbf{p}$ as follows:

$$\hat{\mathbf{p}} = [\hat{R} \quad \hat{X}_c \quad \hat{Y}_c]^T = \left[\sqrt{q_1 + \left(\frac{q_2}{2}\right)^2 + \left(\frac{q_3}{2}\right)^2} \quad \frac{q_2}{2} \quad \frac{q_3}{2}\right]^T \tag{5.9}$$

This regression approach for circle fitting is similar that of Coope [82].

The residuals in the regression are

$$r_i = d_i - \mathbf{h^T}_i \mathbf{q}$$

In this case, the implicit equations for the solution of a GM estimator using standardized residuals become

$$\sum_{i=1}^{M} w\big(\frac{r_i}{s}\big)\ \frac{r_i}{s}\ \frac{1}{s}\ \frac{\partial r_i}{\partial \mathbf{q}} = 0$$

where

$$\frac{\partial r_i}{\partial \mathbf{q}} = \begin{bmatrix} \frac{\partial r_i}{\partial q_1} & \frac{\partial r_i}{\partial q_2} & \frac{\partial r_i}{\partial q_3} \end{bmatrix} = -[1 \quad x \quad y]$$

The scaling term (robust standard deviation) s, is defined as

$$s = 1.4826\ \big[\,1 + \frac{15}{(M-K)}\big]\ \text{median}(|r_i|)$$

Here, the Jacobian is given by $F = -[1 \quad x \quad y]$.

Therefore the IRLS solution of the circle fitting regression problem is

$$\mathbf{q}^{(t+1)} = \mathbf{q}^{(t)} + \big[\mathbf{F}^T\mathbf{W}^{-1}\mathbf{F}\big]^{-1}\mathbf{F}\,\mathbf{W}^{-1}\,\mathbf{r}^{(t)}$$

where $\mathbf{q}^{(t+1)}$ is the estimated value of model parameters in the $(t+1)^{\text{th}}$ iteration. The IRLS solution thus obtained in $\mathbf{q}$ can be used to estimate the real parameters of the circle $\mathbf{p}$, using (5.9).

The non-robust methods commonly use the sum-of-squared error (SSE)

$$SSE = \sum_{i=1}^{M} \Big[(x - \hat{X}_c)^2 + (y - \hat{Y}_c)^2 - \hat{R}^2\Big]$$

as a performance measure to check the circle fit. In the presence of outliers, this SSE measure becomes large due to outliers and is not a good measure to evaluate the validity of estimated parameters. It is therefore not used in this study for performance evaluation. Instead, the results of the robust circle fit are evaluated by comparing the estimated parameters with the actual parameters of the circle.

5.3.3 Experiments

The developed robust circle fitting method was tested using data from simulations and real-images. The simulations were obtained from ideal data on a circle perturbed by different noise sources. The real data represented the locations of edge pixels in an image of real-world circular objects. This image was corrupted by noise to introduce outliers in the data.

In the LMedS procedure used for circle fitting, a three point subset was chosen from the given data set in each iteration. This subset was used in a closed form circle-fitting method by Moura and Kitney [84]

to estimate the circle parameters. With only three points, there is no ambiguity in fitting a circle unless the points are collinear. The residuals are calculated from each parameter estimate and the median of the residuals is minimized to determine the best possible parameters.

Simulation

Consider M data points equally spaced along the circumference of a circle centered at (X_c, Y_c) with radius R. The coordinates of these points are given by

$$\begin{aligned} x_i &= R\cos\big(\frac{2\pi}{M}i\big) + X_c \\ y &= R\sin\big(\frac{2\pi}{M}i\big) + Y_c \end{aligned}$$

where $i = 1, 2, 3, ..., M$. These data points can be perturbed from the ideal circle by introducing noise in tangential and radial directions as

$$\begin{aligned} x_i &= R\cos\big(\frac{2\pi}{M}i + \alpha u[i]\big) + X_c + \beta v[i] \\ y_i &= R\cos\big(\frac{2\pi}{M}i + \alpha u[i]\big) + Y_c + \beta v[i] \end{aligned}$$

where α and β are constants, and $u[i]$ and $v[i]$ are independent, Gaussian noise components in tangential and radial directions respectively. By varying α, we can adjust the distribution of data along the circumference of the circle and with β, we can vary the location of data from the circle (e.g., a large, positive radial noise term creates outliers). Figure 5.1 shows several configurations of points that can be generated using this model.

Image Data

The real data for testing our circle fitting was obtained from an image with objects, circular in shape. The camera was positioned to look down perpendicular (as in imaging a conveyor) to avoid any shape distortion due to perspective projection in the image. Therefore the shape of objects in the image, are nearly ideal circles. An edge detection operation was performed using the Prewitt operator and then salt and pepper noise was added to the edge image (shown in Fig. 5.2). The edge locations from this image were used as the data to test the developed circle fitting method.

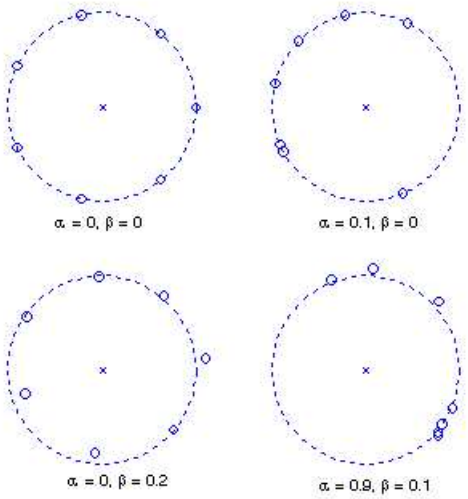


Figure 5.1: Example data configurations with different synthesis parameters. Note that varying by α, the location of points along the circumference can be controlled and varying β causes the points to move inwards or outwards from the circle.

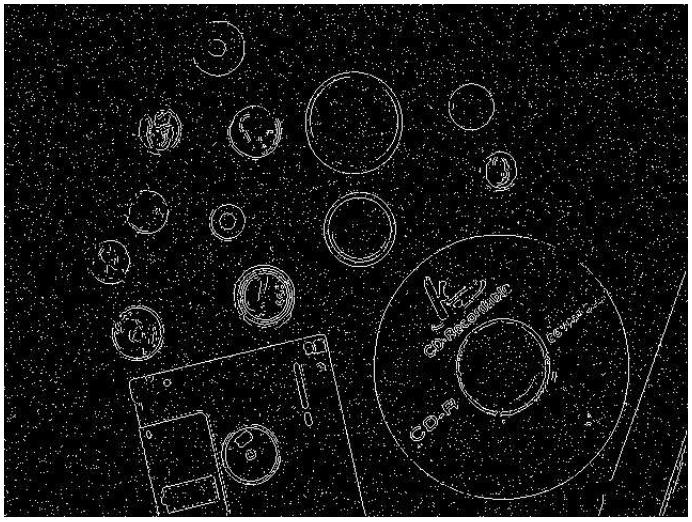


Figure 5.2: Edge image corrupted with noise used to obtain real data for testing the circle fitting.

5.3.4 Results

A least-squares based Gauss-Seidel method [91] was chosen as a representative non-robust circle fitting method to compare the performance of the robust circle fitting method. The robust procedure was initialized using the estimates of center and radius from the least-squares method. The stopping criterion for convergence of parameters was set to $\epsilon = 0.001$.

Simulations

The performance of circle fitting methods was evaluated for different sets of simulation data. Without loss of generality, we chose $R = 1, X_c = 1$ and $Y_c = 1$ for the simulations.

Consider a case with $M = 7$, in which two outliers are present. As shown in Fig. 5.3, good data points tend to lie along the ideal circle, and the outliers are located away from the circle. The presence of such outliers causes the least squares based method to fail and significantly changes the parameter estimates. The robust methods on the other hand, exhibit resistance to outliers by downweighting their influence and so estimate the parameters close to the true values. The robust circle fitting method easily identified these outliers and hence was able to estimate the real circle among the data, same as the LMedS method. The estimated circle from the robust method is exactly the true circle in the given data.

The example data used by Gander et al. [83] is a popular data set to test circle fitting approaches and so it was chosen to test the robust circle-fitting method. Figure 5.4 shows the estimated circles for the Gander data obtained using non-robust and robust techniques. The parameters estimated from least-squares method are same as that reported in [90]. A closer look at the circles estimated shows that a better fitting circle using these data points has been obtained using the robust method. The possible reason is that the robust method downweighted the data which appeared to deviate from the general trend. The estimated values of radius $\hat{R}$, circle center $(\hat{X}_c, \hat{Y}_c)$, and SSE given in Table 5.1 show that the robust method identified a smaller circle possible in the data unlike other methods. As mentioned earlier, the SSE measure does not offer any insight on the parameter estimates when there are outliers in the data.

Real Data

The developed circle fitting method was also tested using the real data from the image in Fig. 5.2. In spite of the extremely noisy data, the robust method identified the true circles shape in the data as shown in Fig.

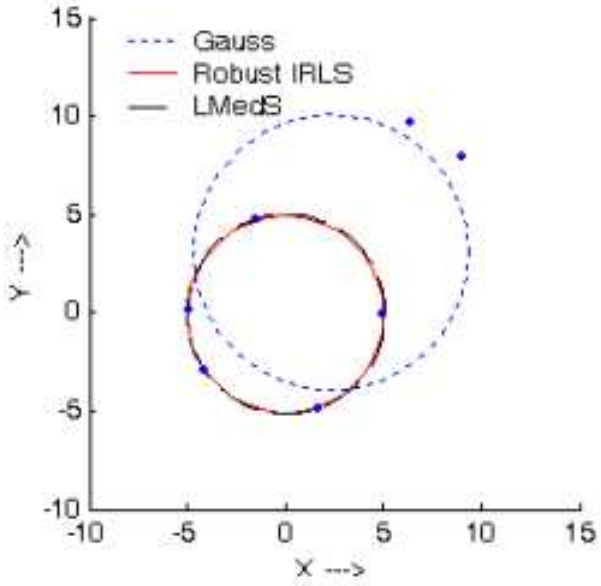


Figure 5.3: An example from simulation with just 7 points 1 and its corresponding results. The robust circle fitting method identified the outliers in the data and ignored their influence to estimate the true parameters of the circle in the data.

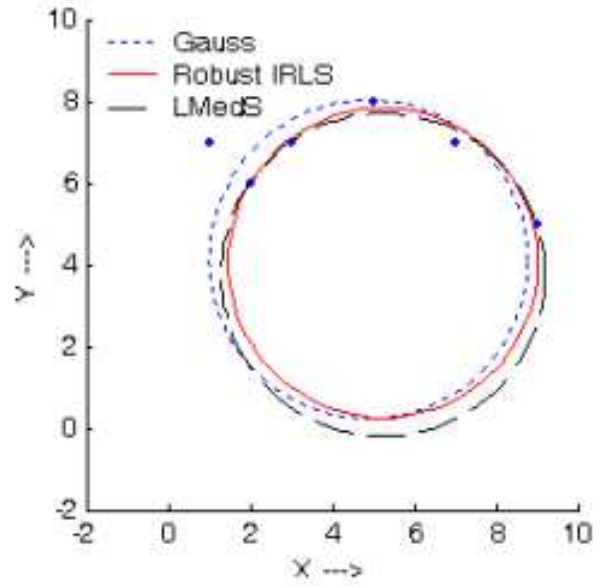


Figure 5.4: Data used by Gander [7] and the estimated circles. Although the ground truth is not known, a visual inspection of the circles fit suggests the presence of an outlier. The robust LMedS method tends to ignore the presence of an outlier while non-robust methods are influenced.

Table 5.1: Estimated circle parameters for data used by Gander et al. [83]. The ground truth in data is not known. Here the Gauss method is a non-robust method, while our robust method (Huber) and the LMedS method are robust methods.

Method	Parameter estimates			
	$\hat{R}$	$\hat{X}_c$	$\hat{Y}_c$	SSE
Least-squares	3.89	4.89	4.12	102.06
Robust GM	3.78	5.25	4.06	163.89
LMedS	3.95	5.25	3.75	179.25

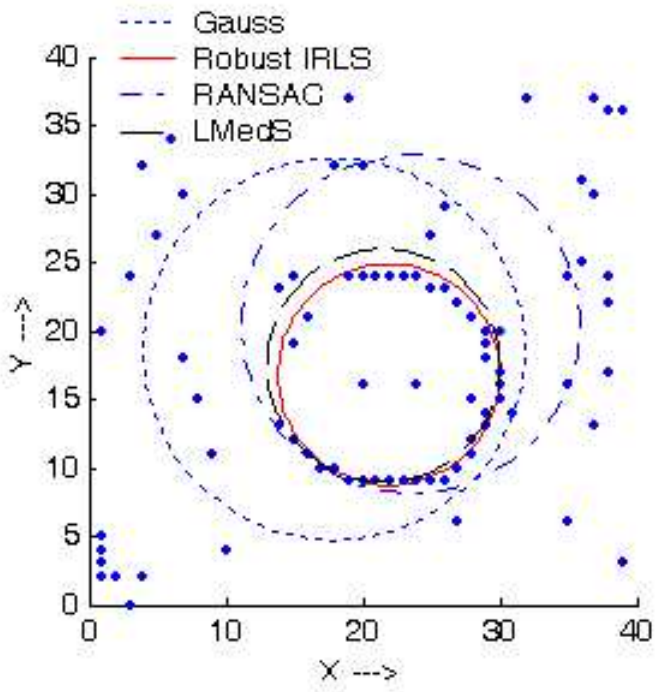


Figure 5.5: Results of circle fitting from an example data from the image in Fig. 5.2. The robust method clearly identified the circle in the noisy data.

5.5. The non-robust method however failed in identifying the true parameter. The estimated parameters of the circle given in Table 5.2 offer an insight into the amount of error present in the traditional least square based method.

Figure 5.6 shows an example where the non-robust method has performed equally well compared to the robust method. But as the other examples show the non-robust method does not consistently perform well in presence of outliers and so the robust method was found to give better results in estimating the circle parameters.

Table 5.2: Parameter estimates corresponding to Fig. 5.5.

Method	Parameter estimates		
	$\hat{R}$	$\hat{X}_c$	$\hat{Y}_c$
Least-squares	13.93	17.89	18.65
Robust GM	8.08	21.89	16.82
RANSAC	12.39	23.5	20.5
LMedS	12.35	23.5	20.5

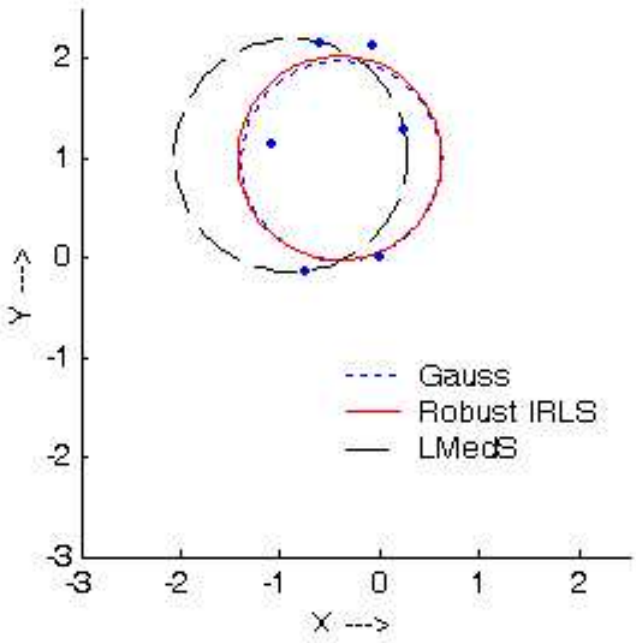


Figure 5.6: An example data from the optical flow calculated using block-matching. From a priori information of the data, the circle should be centered between two extreme points in any direction and the radius is expected to be less than 1 unit (or pixels). In this example, the non-robust method performed equally well in comparison with the robust method. But as the previous examples show its performance degrades in presence of outliers.

5.4 Summary

A robust estimator for circle fitting was designed and its performance was studied through simulations and real data from images. The performance of the robust method was compared with a non-robust least-squares method and robust least median of squares method. The robust method performs significantly well with small number of data points and is also tolerant to the presence of outliers ($< 30\%$).

This chapter presented an introduction to robust statistical methods and a robust circle fitting method with very few data points. The robust methods are well known to perform in presence of severe outliers.

This is a major advantage and hence robust methods are used in applications ranging from simple regression analysis to complex computer vision problems such as line identification, pose estimation and fundamental matrix calculation.

Chapter 6

Robust Ellipse Fitting

The study and development of robust circle fitting methods discussed in Chapter 5 led to an investigation of ellipse fitting in the research. Most available methods for ellipse fitting assume the availability of large data sets and so do not address the problem of fitting ellipses with very few data (typically less than 10). This chapter presents a new, robust ellipse fitting approach inspired by an existing method. This chapter gives an introduction to projection statistics, which is the main approach used to identify outliers. The results obtained from the new robust method using simulations are analyzed. The notations of variables used in this chapter are only related to those used in Chapter 5.

6.1 Problem Statement

An implicit representation of an ellipse relative to a coordinate frame (x, y) is given by

$$\frac{(x' - x_c)^2}{a^2} + \frac{(y' - y_c)^2}{b^2} = 1 \tag{6.1}$$

where $x' = x\cos\phi - y\sin\phi$ and $y' = x\sin\phi + y\cos\phi$. Figure 6.1 shows an example of an ellipse given by this representation. The five parameters of an ellipse are $\mathbf{p} = [x_c,\, y_c,\, a,\, b,\, \phi]^T$, where $(x_c,\, y_c)$ is the center of the ellipse, a and b are the major and minor axes lengths respectively and ϕ is the orientation of the ellipse with respect to the x axis. Thus the ellipse fitting problem can be stated as follows: given a set of data points $\{(x_i,\, y_i) \mid i = 1, 2, 3, \ldots, M\}$, for $M \geq 5$, determine the optimal parameter vector $\hat{\mathbf{p}} = [\hat{x_c},\, \hat{y_c},\, \hat{a},\, \hat{b},\, \hat{\phi}]^T$ which represents the ellipse of best fit to the given data.

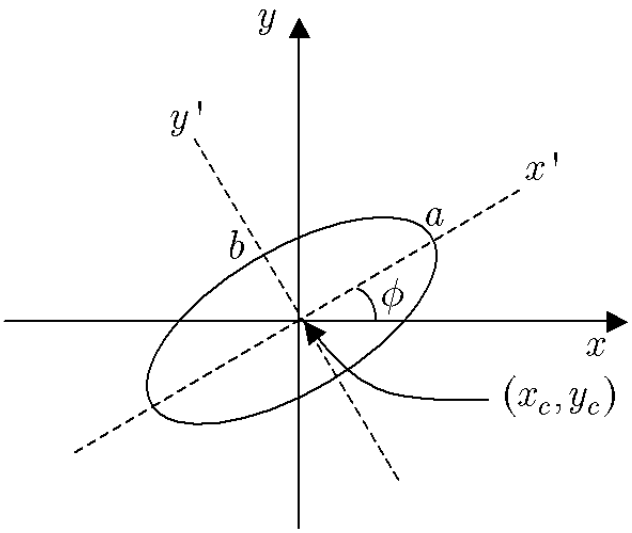


Figure 6.1: An ellipse in a two-dimensional $x - y$ space, rotated about its center (x_c, y_c). The major and minor lengths of the ellipse are a and b respectively. The ellipse is oriented at an angle ϕ from the major axis.

6.1.1 Algebraic Ellipse Fitting

The equation of an ellipse in (6.1) can also be rewritten in the algebraic form of a general conic

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \tag{6.2}$$

which for an ellipse must satisfy the condition $AC < 0$. This algebraic formulation leads to relatively simple procedure to estimate the ellipse parameters. Therefore ellipse fitting methods using this formulation commonly define the residual as

$$r_i = -\left(Ax_i^2 + Bx_iy_i + Cy_i^2 + Dx_i + Ey_i + F\right)$$

and attempt to minimize the error using a least-squares approach $\Sigma_{i=1}^{M} r_i^2$. However the error minimized with this approach has no direct geometric interpretation of the ellipse and so suffers from accuracy as a result [89, 90]. Example of ellipse fitting methods based on this representation are those of Rosin [100], Gander et al. [83], and Fitzgibbon et al. (FPF) [86].

6.1.2 Geometric Ellipse Fitting

In geometric (orthogonal) ellipse fitting, the error defined is based on the orthogonal (perpendicular) Euclidean distance from each data point to the tangent at the nearest point on the estimated ellipse. The orthogonal approach was once considered to be computationally prohibitive compared to the algebraic approach because of the need to compute orthogonal distances in iterative estimation. But this is no longer the case due to the availability of high speed computers.

Ahn et al. [90] developed an iterative Gauss-Newton procedure to estimate the five ellipse parameters. In this procedure, an initial estimate of the ellipse parameters is obtained from a least-squares based circle fitting procedure. The $(\hat{x_c}, \hat{y_c})$ value of the estimated center is then used to transform the given data into a new coordinate system. Then an ellipse is fit in this new co-ordinate system with parameters initialized to the least-squares based estimates. In each iteration, the tangents on the exterior of the estimated ellipse are determined and the orthogonal distances corresponding to each transformed data point is estimated. These orthogonal distances are then used in a minimization to estimate the ellipse parameters $\hat{\mathbf{p}}$. The procedure is repeated until the estimates of the ellipse parameters converge or the maximum limit of iterations is reached.

6.2 Projection Statistics

As mentioned in Chapter 5, identification of outliers in given data is an important step in a robust estimation procedure. The Mahalanobis distance works well for the case of a single outlier but fails to detect multiple outliers in the data. This is because the distances are calculated by projecting the data along the principal axis of distribution and so can not detect multiple outliers. This inability is called the *masking effect* [95].

Stahel [101] and Donoho [102] independently proposed the first, affine, equivariant, multivariate estimators based on the Mahalanobis distance using sample mean and variance. Gasko and Donoho [103] proposed to consider only the distances obtained by projecting the data points onto vectors passing through the coordinatewise median $\mathbf{z}_{med}$ and the data points. The coordinatewise median of a given data is calculated as

$$\mathbf{z}_{med} = [\underset{i}{median}\, z_{i1}\,, \underset{i}{median}\, z_{i2} \qquad ...,\ \underset{i}{median}\ z_{iN}\,]^T \qquad i = 1, 2, 3, ..., M$$

The direction vectors with respect to the coordinatewise median are given by

$$\mathbf{u}_i = \mathbf{z}_i - \mathbf{z}_{med}$$

Normalizing the direction vectors, we have

$$\mathbf{v}_i = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|} \equiv \frac{\mathbf{u}_i}{\sqrt{u_{i1}^2 + u_{i2}^2 + \ldots + u_{iN}^2}}$$

Thus, the projections of given data $\mathbf{Z}$ along all possible directions are given by

$$l_{ij} = \mathbf{z}_i^T \mathbf{v}_j \qquad i = 1, 2, \ldots, M;\ j = 1, 2, \ldots, M$$

whose median is

$$l_{med,j} = median\,\{l_{1j}, l_{2j}, \ldots, l_{Mj}\}$$

To account for the offset (mean) and spread (variance) of these projections, the projections can be standardized as [69]

$$\lambda_{ij} = \frac{|l_{ij} - l_{med,j}|}{1.4826\, median(\,|\,l_{ij} - l_{med,j}|\,)} \tag{6.3}$$

for $i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots, M$. The value 1.4826 is a correction factor normally used for Gaussian approximation [69]. Thus the standardized projections λ_{ij} for the given data are calculated in all possible directions.

The *projection statistics* (PS) of the data are the maxima of λ_{ij} [104]:

$$\lambda_{max,i} = max\,\{\lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{iM}\} \tag{6.4}$$

A weight τ_i corresponding to each data can then be obtained from the projection statistics as

$$\tau_i = min\left(1, \frac{\boldsymbol{\chi}_t^2}{\lambda_{max,i}^2}\right) \tag{6.5}$$

where the threshold $\boldsymbol{\chi}_t^2$ represents the 95% quantile in a $\boldsymbol{\chi}^2$ distribution with t degrees of freedom. As this threshold is a constant, the weights w_i are inversely proportional to the square of the projection statistics. Generally the "good" data have small projection distances and so are assigned a weight close to 1. The farther the location of a data point from the coordinatewise median, the larger is its projection distance and so smaller is its assigned weight. In other words, the weight function in (6.5) tags the data whose projection statistics are larger than the chosen threshold as outliers.

Thus the analysis using projection statistics identifies the outliers in a dataset using scalar quantities and serves as a measure of how the data is distributed along one dimension. The main advantage of this measure is its easy computation despite large dimensionality. However, this measure lacks the affine equivariance

property and hence is a drawback. But this drawback is not a major concern, if the projection statistics are used only to derive weights (as a tool to identify outliers) and not directly in an estimation procedure. Projection statistics was used by Mili et al. [104] in power systems, for a state estimation problem and they demonstrated that this approach is robust to the presence of severe outliers.

6.3 Robust Ellipse Fitting

In [90], Ahn et al. (ARW) used a set of linear equations

$$\mathbf{J}\,\Delta\mathbf{p} = \mathbf{r} \tag{6.6}$$

in a Gauss-Newton iteration to estimate the ellipse parameters, where $\mathbf{J}$ is the Jacobian matrix, $\Delta\mathbf{p} = [\Delta\hat{x_c}, \Delta\hat{y_c}, \Delta\hat{a}, \Delta\hat{b}, \Delta\hat{\phi}]^T$ and $\mathbf{r}$ represents the orthogonal residual distances. The Jacobian matrix is written as

$$\mathbf{J} = \begin{bmatrix} J_{x_1',x_c} & J_{x_1',y_c} & J_{x_1',a} & J_{x_1',b} & J_{x_1',\phi} \\ J_{y_1',x_c} & J_{y_1',y_c} & J_{y_1',a} & J_{y_1',b} & J_{y_1',\phi} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ J_{x_M',x_c} & J_{x_M',y_c} & J_{x_M',a} & J_{x_M',b} & J_{x_M',\phi} \\ J_{y_M',x_c} & J_{y_M',y_c} & J_{y_M',a} & J_{y_M',b} & J_{y_M',\phi} \end{bmatrix}$$

where

$$\mathbf{J}_{x_i',p_j} = \left[\mathbf{R}^{-1}\mathbf{Q}^{-1}\mathbf{B}\right]$$

and

$$\mathbf{R} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} b^2x' & a^2y' \\ (a^2-b^2)y + b^2y_i' & (a^2-b^2)x - a^2x_i' \end{bmatrix}$$

$$
\begin{aligned}
\mathbf{B} &= \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 & \mathbf{B}_4 & \mathbf{B}_5 \end{bmatrix} \\
\mathbf{B}_1 &= \begin{bmatrix} b^2 x' \cos\phi - a^2 y' \sin\phi \\ b^2(y'_i - y')\cos\phi + a^2(x'_i - x')\sin\phi \end{bmatrix} \\
\mathbf{B}_2 &= \begin{bmatrix} b^2 x' \sin\phi + a^2 y' \cos\phi \\ b^2(y'_i - y')\sin\phi - a^2(x'_i - x')\cos\phi \end{bmatrix} \\
\mathbf{B}_3 &= \begin{bmatrix} a(b^2 - y'^2) \\ 2ay'(x'_i - x') \end{bmatrix} \\
\mathbf{B}_4 &= \begin{bmatrix} b(a^2 - x'^2) \\ -2bx'(y'_i - y') \end{bmatrix} \\
\mathbf{B}_5 &= \begin{bmatrix} (a^2 - b^2)x'y' \\ (a^2 - b^2)(x^2 - y^2 - x'x'_i + y'y'_i) \end{bmatrix}
\end{aligned}
$$

This ellipse fitting approach is least-squares based and is non-robust. Therefore it does not estimate the ellipse parameters accurately, when the available data set is very small (less than 10 points, as in the problem of interest) or contains outliers.

This method is modified into a robust ellipse fitting method using projection statistics based robust weighting to identify the outliers in given data. The use of projection statistics modifies the non-robust, least-squares based, ellipse fitting method of ARW into a statistically robust, ellipse fitting method.

From Chapter 5, we know that for a regression model

$$\mathbf{d} = f(\mathbf{p}) + \mathbf{e} \tag{6.7}$$

where $\mathbf{d}$ are the observations, $f(\mathbf{p})$ are some function of the parameters and $\mathbf{e}$ are the residuals, the corresponding iterative reweighted least squares (IRLS) based estimation of parameters is given by

$$\mathbf{p}^{(t+1)} = \mathbf{p}^{(t)} + [\mathbf{F}^T \mathbf{W}^{-1} \mathbf{F}]^{-1} \mathbf{F} \mathbf{W}^{-1} \mathbf{r}^{(t)} \tag{6.8}$$

where $\mathbf{F}$ represents the Jacobian matrix and $\mathbf{W}$ represents the weighting matrix.

Extending the same principle for ellipse fitting, we obtain the parameter update relation

$$\mathbf{p}^{(t+1)} = \mathbf{p}^{(t)} + [\mathbf{J}^T \mathbf{W}^{-1} \mathbf{J}]^{-1} \mathbf{J} \mathbf{W}^{-1} \mathbf{r}^{(t)} \tag{6.9}$$

for the statistically robust, ellipse fitting method, where the superscript t represents the parameter value at the t^{th} iteration. The weights in $\mathbf{W}$ are calculated based on the robust Huber influence function and on the projection statistics. It is assumed that impulsive noise is present in the data and it is independent and identically distributed (i.i.d) Gaussian. This assumption allows to adopt the weighting procedure based on projection statistics and therefore optimally estimate the ellipse parameters $\mathbf{p}$.

The modified robust ellipse fitting procedure identifies the outliers using projection statistics (6.4). The residuals are considered as a two-dimensional data ($N = 2$) and projection statistics are calculated in this 2D space using the residual components as observations, $\mathbf{Z}$. Because the residual distances directly correspond to the transformed data, the projection statistics can identify the outliers based on the residues which deviate from the majority of the residuals. As the given data is transformed based on the estimated ellipse in each iteration, the weights assigned to each data point also change with iteration. The farther a data point lies from the estimated ellipse, larger is its orthogonal residual distance and so lower is its weight assigned.

For an illustration of this procedure, consider the data in Figure 6.2, where data points (indicated by 'o') are generated by adding noise to ideal data from an ellipse. The parameters of the ellipse are $[x_c = 0, y_c = 0, a = 7, b = 2, \phi = \pi/6]$ with $M = 7$. In addition to the outlier noise, this data includes environment noise (Gaussian) and therefore all data do not lie exactly on the generated ellipse. The orthogonal points (indicated by '*') corresponding to given data on the estimated ellipse in an iteration are also shown in this figure. The error components along both axes are then analyzed in a 2D projection space for identifying outliers as shown in Fig.6.3. The associated projection statistics corresponding to this example is [0.2001, 0.7388, 0.6745, 3.1022, 0.5413, 19.1283, 2.0837], where the largest value clearly identifies the outlier. Based on these values, the weighting function downweights this outlier significantly in the estimation procedure.

The IRLS procedure requires an initial estimate of ellipse parameters. So it is initialized with the center of the ellipse as the coordinatewise median of given data (a robust feature), the major and minor axis lengths as an arbitrary value (say 1) and the orientation angle as zero.

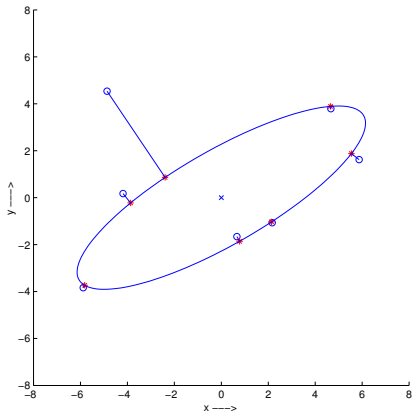


Figure 6.2: Example of outlier detection using the method of projection statistics. Seven data points are shown ('o') including a single outlier. Orthogonal distances to an ellipse are indicated for each point.

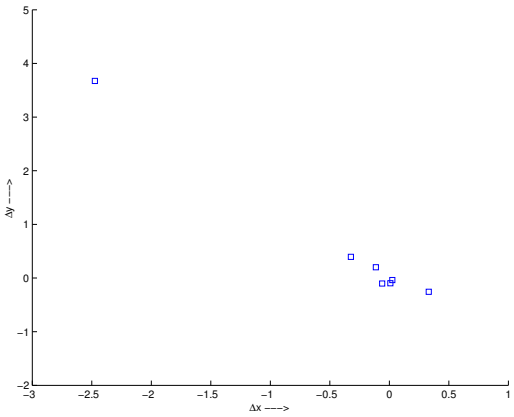


Figure 6.3: Two-dimensional space of orthogonal error vectors corresponding to example in Figure 6.2. These are used in the calculation of projection statistics. For inlier points, the error vectors cluster near the origin.

6.4 Results

The performance of the robust orthogonal ellipse fitting procedure was assessed by comparing its results from those of the existing non-robust methods and other well known robust techniques. Experimental simulations were used in this analysis so that the ground truth is known and therefore the performance could be measured.

6.4.1 Experimental Setup

Estimates of ellipse parameters from simulations with $M = 9$ data points over 1000 runs in a Monte-Carlo approach were collected to analyze the performance of the ellipse fitting methods. Because the true ellipse

parameters were known, the estimates obtained can be evaluated for accuracy. For simulation, the ellipse parameters $[x_c = 0,\, y_c = 0,\, a = 7,\, b = 2,\, \phi = \pi/3\,]$ were chosen.

To each of the M data points, a small measurement error (equivalent to environment noise, $\sigma^2_{env} = 0.2$) was added. In addition, few selected points among the given data were made outliers by adding a noise of higher variance (σ^2_{noise}). The number of outliers created is dependent on the percentage of contamination chosen in the simulation. As mentioned earlier, the initial estimate in the iteration was chosen to be a unit circle, centered at the coordinatewise median of the given data, with an orientation angle $\hat{\phi} = 0$.

With $M = 9$ and 30% contamination, two points in each data set are corrupted by impulse noise. This leaves only 7 data that are valid. This is barely enough redundancy to estimate a fitting ellipse or its parameters. Therefore this case, in a way, can be considered as a "stress test" for the robust ellipse fitting method.

The algebraic ellipse fitting method proposed by Rosin [88] was chosen to compare the performance of ellipse fitting methods. This algebraic method is a least-squares fitting method, in which the constraint $A + C = 1$ was chosen, as the ellipses are centered at the origin. The algebraic ellipse parameters obtained in this method were converted to geometric ellipse parameters using the method given in [105].

The LMedS and RANSAC implementations in this study used the ellipse fitting procedure outlined in [86]. In any iteration, both methods fit an ellipse, using a subset of five points from the given data. When the error due to the ellipse parameters estimated using this subset, with respect to other points in the data is within an allowable limit, the parameters are assumed to have been identified and no more iterations are carried out. An exhaustive LMedS method was implemented and so it almost always identified the ellipse parameters close to the "ground truth". The RANSAC implementation used a threshold value of 50% (two, when $M = 9$) to consider any fit good. i.e., if any two data points, other than those chosen in an iteration, agree with the estimated parameters, then the fit is declared to be good and the iteration is terminated.

6.4.2 Observations

The performance of the RANSAC method is dependent on the threshold values chosen and the results are not deterministic due to its random sampling nature. In other words, the estimates obtained from RANSAC each run may not be the same in each run. With the chosen threshold in this study, the RANSAC method occasionally found an incorrect ellipse although the parameters agreed with the specified threshold. One

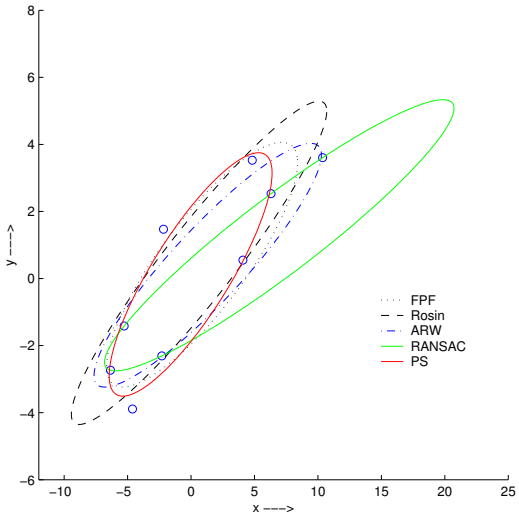


Figure 6.4: An example case where RANSAC method produces an incorrect ellipse estimate, yet which agrees to chosen threshold values.

such case of this anomaly is shown in Figure 6.4. It can be seen that the RANSAC result (shown in a green solid line) is substantially different from the ground truth and also the estimates of other methods. This is because RANSAC picked the outlier among the subset of points to fit in an iteration, and incidentally an ellipse was identified to fit the data. So a further search was not performed, producing a poor estimate of ellipse parameters.

The exhaustive LMedS method in the simulation always identified the 'true' ellipse and therefore served as a reference, to compare the performances from other methods. Because these parameters were close to the 'true' parameters, its results are not included in the following discussion.

6.4.3 Performance Measure

For a statistical analysis of the performance of ellipse fitting methods in a Monte-Carlo type simulation, a histogram analysis on each parameter estimate can be used. The mode (or peak) of a histogram for each of the ellipse parameter offers a good insight into the ellipse fitting procedure. Ideally, we expect the modes to be centered about the true ellipse parameters. Let the maximum number of the estimates that correspond to the ellipse center be denoted as $(\eta_{\hat{x_c}}, \eta_{\hat{y_c}})$. Similarly, let $\eta_{\hat{a}}$ and $\eta_{\hat{b}}$ represent the maximum number of estimates for major and minor axes and let $\eta_{\hat{\phi}}$ be the corresponding value for the orientation angle. The ideal value for all these numbers is one, indicating that 100% of all the estimates are equal the true ellipse parameters, A value close to zero indicates that the estimator did not find the true parameters.

In this statistical analysis, the statistical bias (δ) in the fitting method is also calculated. If the resulting ellipses are biased, then the estimates tend to deviate from the true parameter value. The bias in five ellipse parameters ($\delta_{\hat{x_c}}$, $\delta_{\hat{y_c}}$, $\delta_{\hat{a}}$, $\delta_{\hat{b}}$, $\delta_{\hat{\phi}}$) are calculated in this study and ideally, the bias should be zero.

Because the five parameters in an ellipse are not independent, a figure of merit to compare the results is defined as:

$$\gamma = \frac{1}{5}\big(\eta_{\hat{x_c}} + \eta_{\hat{y_c}} + \eta_{\hat{a}} + \eta_{\hat{b}} + \eta_{\hat{\phi}}\big) \tag{6.10}$$

This measure gives an indication of the quality of an ellipse fit. Although parameters like centroid and eccentricity of the fit ellipse could be used, they do not weigh the ellipse parameters equally and so are not used in this analysis.

6.4.4 Performance Analysis

First the case of high noise variance ($\sigma^2_{noise} = 10$) was considered for analysis. This case poses a strong challenge in estimation. With $M = 9$ and 30% contamination, this adds high impulsive noise to two data points.

Table 6.1 summarizes the results obtained from a Monte-Carlo simulation of 1000 runs, in this high noise case. The first three rows in the table correspond to non-robust methods and the last two rows are robust methods. The performance measure of each non-robust method is significantly less than $\gamma = 0.5$ compared to the robust methods. The RANSAC method and the new robust PS method both exhibit $\gamma \approx 0.5$. Moreover, a closer look at the individual η values clearly shows poor performance, as expected from the non-robust methods. The performance of the robust PS method is on par (at times slightly better) than the RANSAC method. It should be noted that the PS method has statistics and bias which are very similar to the RANSAC method.

The advantages of the PS method are in the independence of the threshold value, unlike RANSAC, and it is also deterministic. An observation on the bias of the parameter estimates, in addition to the statistics, again show that the robust methods perform well in presence of noise.

Table 6.2 presents results for the case when the noise level was set to medium ($\sigma^2_{noise} = 5$) in the simulations. In this case also, the robust methods perform better than the non-robust methods. The non-robust algebraic method of Rosin and the ARW method improve slightly in performance with medium noise. The increase in performance measure for our PS method against RANSAC can be explained using the

Table 6.1: Comparison of statistics (and bias) of ellipse fits from representative ellipse fitting methods, with 1000 runs ($M = 9$, $\sigma^2_{noise} = 10$, contamination = 30%). The first three rows of the table represent non-robust methods. Our robust PS method is on the last row, and it exhibits performance essentially equivalent to the RANSAC method.

Method	$\eta_{\hat{x_c}}\,(\delta_{\hat{x_c}})$	$\eta_{\hat{y_c}}\,(\delta_{\hat{y_c}})$	$\eta_{\hat{a}}\,(\delta_{\hat{a}})$	$\eta_{\hat{b}}\,(\delta_{\hat{b}})$	$\eta_{\hat{\phi}}\,(\delta_{\hat{\phi}})$	γ
FPF [86]	0.26 (-0.23)	0.24 (-0.14)	0.19 (-0.47)	0.17 (0.20)	0.43 (-0.01)	0.26
Rosin [88]	0.37 (-0.23)	0.26 (-0.46)	0.37 (2.30)	0.40 (0.50)	0.40 (1.54)	0.37
ARW [90]	0.31 (-0.30)	0.25 (-0.45)	0.44 (0.90)	0.32 (0.57)	0.33 (0.04)	0.33
RANSAC	0.44 (-0.49)	0.45 (-0.12)	0.62 (1.10)	0.57 (-0.05)	0.55 (0.06)	0.53
PS	0.46 (-0.34)	0.42 (-0.31)	0.62 (0.98)	0.58 (-0.09)	0.49 (0.06)	0.52

Table 6.2: Comparison of statistics (and bias) of ellipse fits from representative ellipse fitting methods, with 1000 runs ($M = 9$, $\sigma^2_{noise} = 5$, contamination = 30%). In this case, our PS method performed better than the RANSAC method.

Method	$\eta_{\hat{x_c}}\,(\delta_{\hat{x_c}})$	$\eta_{\hat{y_c}}\,(\delta_{\hat{y_c}})$	$\eta_{\hat{a}}\,(\delta_{\hat{a}})$	$\eta_{\hat{b}}\,(\delta_{\hat{b}})$	$\eta_{\hat{\phi}}\,(\delta_{\hat{\phi}})$	γ
FPF [86]	0.28 (0.02)	0.26 (-0.16)	0.19 (-0.00)	0.17 (-0.19)	0.19 (-0.01)	0.22
Rosin [88]	0.44 (-0.40)	0.44 (0.03)	0.45 (-1.08)	0.43 (-0.50)	0.48 (1.60)	0.45
ARW [90]	0.51 (-0.17)	0.40 (-0.27)	0.61 (0.88)	0.46 (-0.41)	0.43 (0.06)	0.48
RANSAC	0.58 (-0.36)	0.46 (-0.02)	0.65 (1.14)	0.64 (0.10)	0.62 (0.00)	0.57
PS	0.43 (-0.41)	0.36 (-0.49)	0.71 (-0.25)	0.41 (-0.50)	0.58 (0.04)	0.69

discussion in Section 6.4.2. Comparatively, RANSAC has a smaller bias among all the methods in this case. For low noise cases ($\sigma^2_{noise} = 1$), the performance of the non-robust methods improves and comes close to that of the robust methods.

The presence of high variance noise leads to poor performance by non-robust methods and it is clear from the analysis that the projection statistics based orthogonal ellipse fitting method performs significantly better than other methods, with very few data points under high noisy conditions and with low data redundancy.

6.5 Summary

This chapter presented a new robust approach for orthogonal ellipse fitting. The new method used projection statistics to identify outliers and estimating the weights. Most previous work has utilized either high data redundancy or has not included outliers. In contrast, this ellipse fitting study was concerned about fitting an ellipse with low data redundancy in presence of high variance noise. The method is successfully tested in Monte-Carlo type simulations with small data sets with high noise. The results show that the new method

performs on par with (or better, in some cases) the RANSAC based method. The main advantage compared to the RANSAC method is the deterministic nature of the parameter estimates unlike the estimates from RANSAC method. From the simulations, it is also showed that the developed method has a statistically good performance with negligible bias in the estimated ellipse parameters.

Chapter 7

Results

This chapter presents the results obtained from the layer extraction and image compositing methodology given in Chapter 4. Also a study on the passive range estimation technique described earlier is presented along with example results from various experiments. The composited images clearly demonstrate good results for both indoor and outdoor scenes, involving objects and people in a stationary scene.

7.1 Image Acquisition

The experiments used to test the developed methodology were performed using a moving-aperture lens designed in early 1996 [60]. This lens consisted of a square aperture created by four leaves and was available in two lens assemblies – 24mm/f:2 and 50mm/f:1.4. The former lens provided a large field of view but had been battered by extensive use over years while the latter lens, although relatively new, had a smaller field of view which was found to be a disadvantage from the experiments.

The MOE controller knobs were generally set to provide maximum disparity in the images. This is possible by choosing a setting with a large f-stop (small aperture) and a large amplitude of rotation with nearly maximum scaling along the X and Y axes of the controllers.

The moving-aperture lens was attached to a Sony DFW-V300 digital camera, interfaced to a computer using IEEE 1394 (Firewire) [106]. This camera is a single CCD design (Sony ICX084AK image sensor with square pixel), capable of capturing color images in different resolutions – 160×120 pixels, 320×240 pixels and 640×480 pixels at three sampling rates YUV(4:4:4), YUV(4:2:2), and YUV(4:1:1) respectively. The image capture rates supported by this camera are 3.75, 7.5, 15, and 30 frames per second (fps). The camera

operates as an interlaced type camera at 640×480 pixels capture mode whereas it operates in a progressive scan mode at other lower resolutions.

In this research, the 640×480 resolution was chosen as it is the standard broadcast video (VGA) size and therefore is a good image size to aim for real time applications. At this image resolution, the camera operates in the interlaced mode. The images were captured at 30 fps (which is close to the 29.97 fps standard in NTSC television broadcast), with a frequency of aperture rotation at 5 Hz (or rotations per second). This setting gives a maximum possible integer number of images in one cycle of aperture rotation (see Chapter 4) and hence best suits the image analysis. It also ensures that there are more than 3 images in an aperture cycle and thus provides redundancy in disparity estimation of a scene. From various experiments, it was observed that any higher frequency of rotation tends to introduce motion blur in the images, which is a severe problem for image analysis. Moreover the effect of operating the camera in interlacing mode becomes pronounced in the images, even for stationary scenes, at any higher frequency of rotation. Therefore the settings mentioned above offered the optimum configuration for analyzing the moving-aperture images.

The image capture software (Coriander[1]) stores the uncompressed, raw images from the camera in YUV411 format, which represents the luminance (Y component) and chrominance (UV components) in the image. The 411 sampling gives a fine sampling of the luminance component and a coarse sampling of the chrominance component in the image. This camera is a single CCD camera, utilizing a Bayer color filter array, in which the photo-sensors are arranged as shown in Fig. 7.1 (this array pattern is called the Bayer pattern [107]). The green color is sampled twice that of red or blue, because the human visual system has varying resolution at different spectral wavelengths. The Bayer pattern is often used in low-cost color CCD cameras to reduce the cost involved in placing three separate photo sensors for red, green and blue colors in a camera. Therefore the final image output from the camera undergoes an interpolation of information from these photo sensors on the CCD to convert the captured image into YUV411 (luminance-chrominance) format. The Bayer array interpolation or 'de-mosaicing' module of a camera is a vital component which ensures that the information on the CCD plane is best represented in the camera image output. Due to this de-mosaicing step, the image output (YUV) from the camera is not a pixelwise, exact representation of the information on the CCD plane and therefore is a drawback when using the single CCD cameras.

The developed methodology uses grayscale images. Therefore the captured images are converted to grayscale by simply ignoring the chrominance portion of the captured images and instead using only the

[1] http://coriander.sourceforge.net

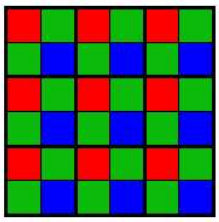


Figure 7.1: Photo-sensors arranged in the Bayer pattern on a single CCD array.

luminance information.

Given the emphasis of this research, only stationary scenes were considered. A short sequence of images from stationary scenes (typically for 1 or 2 seconds) with less than 30 images were captured. Of the images in a sequence, typically six were used for analysis as they represent the one complete rotation of the aperture. (It is possible to analyze longer image sequences, but in a stationary scene analyzing longer sequences offers no advantage.)

The chosen configuration of 6 images per aperture rotation makes the aperture-polygon (Chapter 4), a hexagon. The methodology given in Chapter 4 and illustrated for a 4-point square (aperture-polygon) is easily extendable to a hexagon case. This setting gives N=6 and hence the expected exterior angles of the hexagon vertices are $\theta = \frac{2\pi}{6} = \frac{\pi}{3}$radians.

7.2 Additional Steps in Post-processing

The results in this chapter were obtained with minor additional steps in the methodology described in Chapter 4. These additional steps will not be necessary if the obtained images aere of good quality. Unfortunately, with the version of moving-aperture lens used in this research, there were visible intensity variations in the images and this posed a problem in optical flow estimation. This error can be easily noticed in the optical flow vectors shown later in the results. Therefore, certain post-processing steps were necessary to automatically extract layers from an image sequence.

Because the disparity map from circle fitting step was noisy, a median filter of size 5×5 was used to smooth the disparity map $\mathcal{D}$. Moreover, the values in the filtered map were almost similar in disparity (with small deviation) that automatically extracting layers posed problems. Therefore a scaling was introduced. The objective is to segment the disparity map and extract layers, automatically and so a scaling (by 10) will

uniformly increase the disparity values and preserve the underlying relative segmentation structure.

7.3 Layer Extraction and Image Compositing

This section presents results that were obtained image sequences from various types of stationary scenes involving objects and people – both indoors and outdoors. The results are explained in detail for the first image sequence. The results for other image sequences are obtained using the same approach and so can be understood from the first example. Small changes in the approach, if any, used to obtain the results are clearly mentioned in the discussion.

7.3.1 Textured Cloth

In this experiment three textured cloth objects were placed at three separate distances of 5m, 8m, and 11m from the camera with the f=50mm lens. The cloth objects were suspended vertically to emulate vertical 2D distance (depth) planes in the scene. The field of view of the camera was carefully chosen to include objects at other distances, such as the ceiling and floor of the aisle. The ceiling and floor areas extend into the distance. All the objects in the scene were stationary. An image sequence of this stationary scene was captured using the moving-aperture lens camera to test the layer extraction and image compositing methodology presented in Chapter 4.

Figure 7.2 shows an image I_0 from the captured sequence. This image shows the three distinct textured objects at different distances. The image region on the top is the ceiling and two small regions in the bottom right area correspond to the floor. Both the ceiling and floor are at greater distance than 11m.

By varying the focus setting of the lens, the stationary distance plane in the scene can be varied. The camera focus can be set to distance of 5m (near focus), 8m (mid focus), 11m (far focus) or the ceiling (background).

Let us consider the mid-focus case, where the camera is focussed at the cloth at 8m. The image sequence captured from the moving-aperture lens will exhibit an apparent motion (planar parallax). Being in focus, the image region corresponding to the middle plane will appear stationary in the sequence while all other regions in the image will appear to move.

Although the two non-focal distance planes (5m and 11m) are at same distances from the plane of focus

Figure 7.2: An image from the textured cloth sequence, with the middle textured cloth at 8m in focus.

(8m) but on opposite sides, the apparent image motion induced in the images is not of the same magnitude. This can be understood from the relation 4.24 in Chapter 4. When the camera is focussed at Z=8m and the other two objects are at $Z_1 = 11m$ and $Z_2 = 5m$, then $\Delta Z_1 = 3m$ and $\Delta Z_2 = -3m$. Therefore their corresponding image displacements are

$$\begin{aligned} v_1 = \kappa r_1 = \kappa \frac{R\,f}{(Z-f)} \frac{\Delta Z_1}{Z_1} \quad &\Rightarrow \quad v_1 \propto \frac{\Delta Z_1}{Z_1} \\ v_1 \propto \frac{3}{11} \quad &\Rightarrow \quad v_1 \propto 0.27 \\ v_2 \propto \frac{-3}{5} \quad &\Rightarrow \quad v_2 \propto -0.6 \end{aligned}$$

Optical flow estimation

Figure 7.3(a) shows the optical flow v_{01}, estimated from the images I_0 and I_1 in one sequence. The flow map was estimated using the epipolar line search method, after a global optimization of the aperture polygon angles. (The vectors in Figure 7.3 are scaled for display as the real values were too small to represent the relation.) The vectors (drawn to scale) show that the magnitudes are proportional to the distance from the plane of focus and their direction depends on the location of the focal plane.

The 1D search is limited to a range of 4 pixels at 0.2 pixel interval, on either side of an image point, at the angle determined from the optimization step.

The dense optical flow map in Fig. 7.3(a) can itself be used for segmentation. But as the map shows, the estimates are quite noisy and so it becomes a problem for segmentation based only on one dense map.

Therefore we need to estimate multiple, dense flow maps in the sequence, for one complete aperture rotation. Figures 7.3(b)–(d) show the optical flow maps v_{12}, v_{23}, v_{34} and v_{45} estimated from image pairs $I_1 - I_2$, $I_2 - I_3$,..., and $I_4 - I_5$ in the sequence.

Disparity estimation

Five optical flow maps are used to estimate the disparity in the images. For every point in the image, we have five corresponding optical flow vectors from the flow maps $v_{01}, v_{12}, v_{23}, ..., v_{45}$. Therefore we can find six vertices from the five flow vectors for each image point.

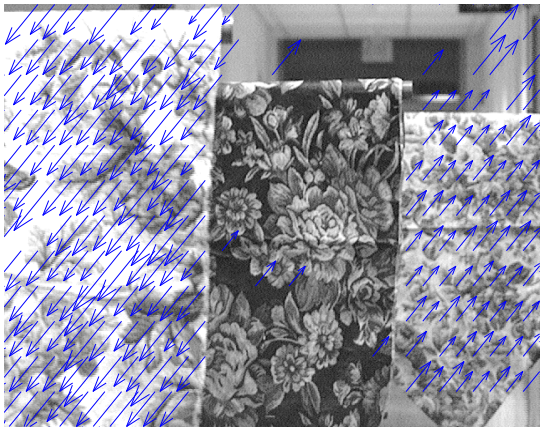
From the six vertices corresponding to each image point, the system fits a circle to estimate the disparity at every point. This circle fitting step estimates the radius and center of the best fitting circle at each point. The location estimation procedure then assigns a sign to the radius, to group the magnitudes into two classes – layers in front, or behind the plane of focus. No attempt is made to resolve this ambiguity as the intended layer extraction application is not dependent on this decision.

The corresponding disparity map $\mathcal{D}$ obtained from the flow maps is shown in Fig. 7.4(a), which clearly indicates the presence of multiple layers. The image regions corresponding to the object in focus have a disparity of approximately zero, and the image regions of objects on opposite sides have increasing disparities, with opposite signs, as their distances increase from the plane of focus. The distribution of disparities in the map is given in Fig. 7.4(b), where each peak corresponds to an image region with a given disparity. The magnitudes of disparities are shown in Fig. 7.4(c), and the signs of disparities shown in Fig. 7.4(d).

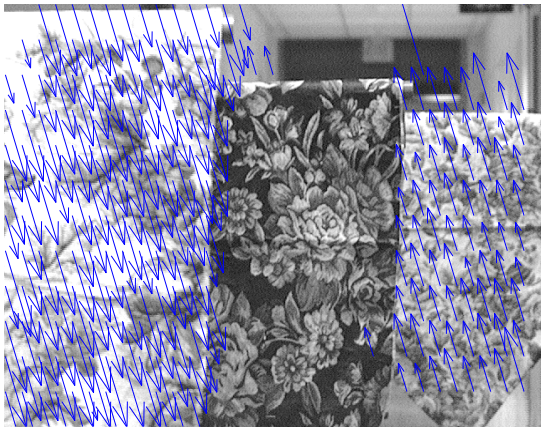
Although the 1D epipolar search is limited to 4 pixels on either side of epipolar line, the circle fitting step estimated radii larger than this value because of the noise. Therefore any radius value estimated as more than given range of 4 pixels is forced to a disparity of +5 pixels. This tagging helps to identify the image point as having an incorrect estimate and so explains the small peak in the histogram in Fig. 7.4(b) at +5 pixels.

The disparity map after median filtering is shown in Fig. 7.5(a) and its corresponding histogram is shown in Fig. 7.5(b). From the histogram, it is clear that the peak at +5 pixels in Fig.7.4(d) has been removed by the filtering process.

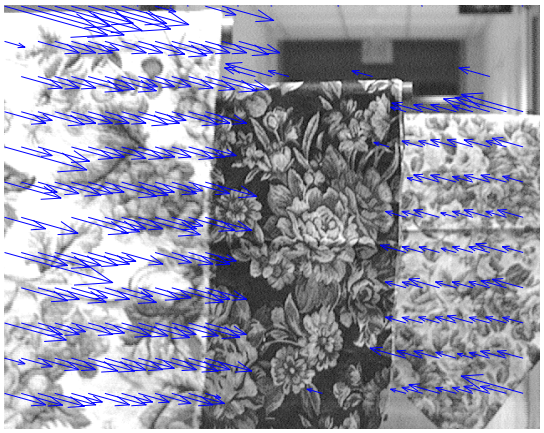
(a)



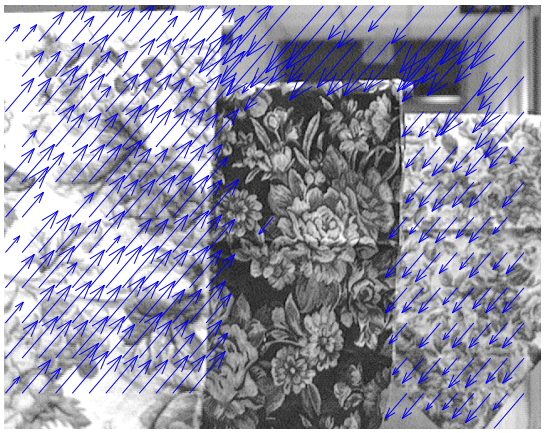
(b)



(c)



(d)



(e)

Figure 7.3: Optical flow maps. (a) v_{01}, estimated from image pair $I_0 - I_1$ in the sequence, and other maps: (b)v_{12}, (c)v_{23}, (d)v_{34}, (e)v_{45} from the consecutive image pairs. The vectors are displayed sparsely for clarity, although they are estimated densely, for every point in the image.

Layer estimation

The signed radius values in Fig. 7.5(a) indicating the image disparity is used for layer analysis in the scene. A segmentation based on the disparities will yield the image region that corresponds to distance planes in the scene.

First the number of layers in the scene is estimated by analyzing the distribution of values in the disparity map. If the object distances are sufficiently separated (in Z), then the disparity histogram will show distinct peaks corresponding to these planes.

Figure 7.5(b) shows the histogram of disparities corresponding to Fig. 7.5(a). The number of bins was empirically chosen as 40, sufficient to distinguish the different peaks. It should be noted that the histogram shows a small peak at +5 pixels disparity, which corresponds to the incorrect estimates, tagged in the circle fitting procedure.

With the assumption that an object in a scene must occupy at least 5% of the image area, to be identified as a layer, we threshold the histogram at 0.05 and increase the floor of the histogram for analysis. The threshold can also be set at 5% of the maximum of the peak in the normalized histogram, to provide a scene dependent criterion.

Then the bins whose peaks are greater than the 0.05 threshold are identified as representing candidate scene layers. Due to the optical properties of the lens and depth of field, these peaks corresponding to possible layers are generally not distinct. Therefore a distance separation criterion is necessary to identify the best layers. This is done by considering that distance planes are sufficiently separated and a minimum separation of 0.5 pixels (with scaling becomes 5 pixels) is assumed. The minimum disparity separation enforced on the candidate peaks determine the peaks which really correspond to the layers in the image. The bins of the finalized peaks give the disparity of each layer in the scene. Thus the system determines the exact number of layers in the image scene, along with an estimate of image disparity.

In Fig. 7.5(b), the candidate peaks (scaled by 10) in the disparity map were identified as

$$[-16.24 \quad 2.96 \quad 4.88 \quad 10.64 \quad 20.24 \quad 22.16 \quad 26.00]$$

Because the main objective is to segment the disparity map, a scaling of disparities did not affect the segmentation. It in fact helped to separate the layers clearly. After the disparity map is scaled by a factor

of 10 and using the minimum layer separation criterion, the best peaks corresponding to the layers were

$$[-16.24 \quad 2.96 \quad 10.64 \quad 20.24]$$

Therefore the system correctly identified the presence of 4 layers in the scene.

Layer extraction

With the number of layers determined and an estimate of their disparities known, the disparity map must be segmented, to determine image regions that are associated with scene layers.

The histogram of disparities can be treated as a distribution of data from mixture of Gaussian distributions, each with its own mean and variance. The number of mixtures estimated directly correspond to the number of layers and their mean values correspond to the mean disparity of layers.

Using the EM approach, a layer is associated with each image point. This procedure assigns one of the integer labels $\{1, 2, 3, 4\}$ representing the layer at each point. The label map thus obtained corresponds to the segmented result of the disparity map and hence gives the layers extracted for the scene. Figure 7.6(a) shows the Gaussian mixtures models on the filtered disparities, estimated using the EM procedure. The parameters of the GMM estimated are

$$\begin{aligned} \text{mean} &= [-16.24 \quad 2.96 \quad 10.64 \quad 20.24] \\ \text{variance} &= [0.92 \quad 0.30 \quad 0.34 \quad 209.21] \\ \text{proportionality} &= [0.35 \quad 0.26 \quad 0.17 \quad 0.23] \end{aligned}$$

Note that although the standard deviation of disparity for layer 4 is 209.21. This layer corresponds to the ceiling and floor, and is estimated with large error. But because other layers are estimated more accurately, this standard deviation is not a cause of great concern. However, the EM method identified the mean for layer 4 as a component with a lower average, with large variance. This creates problems in labeling layer 4, as can be seen from Fig. 7.6(b), in regions corresponding to layer 4.

Layer smoothing

As the result in Fig. 7.6(b) shows, the initial layer extraction does not yield smooth, uniform image regions. The result appears noisy and therefore needs to be smoothed. This can be solved by using a region smoothing

approach, considering the context of neighbor labels and the corresponding disparity values. Markov Random Fields (MRF) is a technique where contextual information can be used in an energy function to perform this smoothing and improve the layer segmented result from the EM procedure. MRF smoothing removes the spurious labeling in image regions and thus produces smooth, uniformly labeled layer map as shown in Fig. 7.7.

The image regions corresponding to the estimated scene layers are shown in Fig. 7.8. The four layers estimated in the image clearly represent the four distance planes in the given scene and these regions are the subsets of the first camera image, as illustrated in the introduction in Chapter 1. Although the images regions shown are not ideal and have occasional noise, they significantly capture the information in the scene. It should be recalled that these layers were extracted automatically from the images, with no initialization (or hints) from the user. Therefore these layers represent highly satisfactory results. With some little manual input, the noise in layers can be easily removed to represent the ideal image layers. Such fine tuning operations are left as a future work of the research.

Image compositing

From the image layers extracted in the previous step, the region corresponding to layer 2, corresponding to the object in focus (at 8m), is chosen for compositing. The matte (α) for this particular object is extracted by creating a binary image setting pixel locations corresponding to the object in the image with a value one (representing transparency). The pixel locations not present in layer 2 are set to zero (representing opaqueness). The binary matte α, thus extracted for layer 2 is shown in Fig. 7.9(a).

For compositing, an arbitrary background image I_B, of same size as the camera image is chosen and this image is shown in Fig. 7.9(b). Using the extracted matte with the camera image I_F, in Fig. 7.2, the composited result obtained is shown in Fig. 7.9(c). The composited image shows that the object from the camera image is successfully merged with the chosen background image.

With this example, we observe that the developed layer extraction methodology clearly works as desired and designed. Also the composited result obtained provides a clear proof of the validity and efficacy of the developed image compositing methodology using the moving-aperture lens.

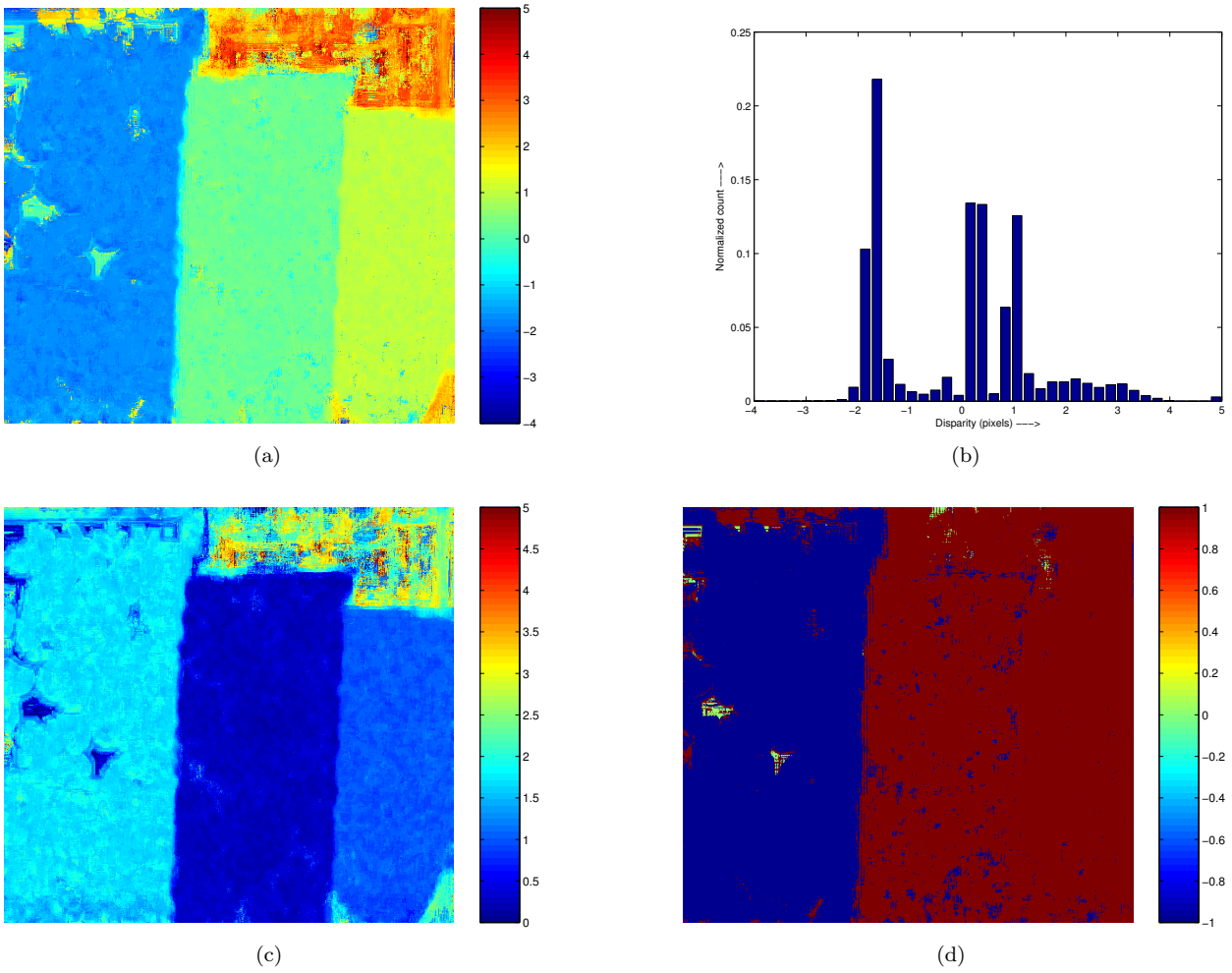
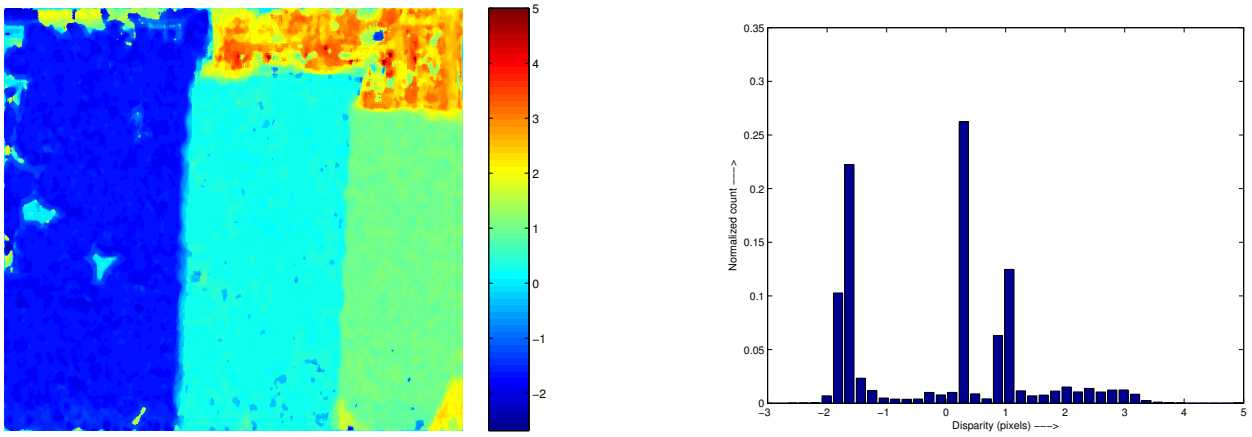


Figure 7.4: (a) Disparity map obtained from circle fitting using the optical flow vectors; (b) magnitude of the disparities (radius of circles fit), indicating the relative distances from the plane of focus; (c) sign of disparities, indicating the relative locations from the plane of focus.

Comparison with Lucas-Kanade method

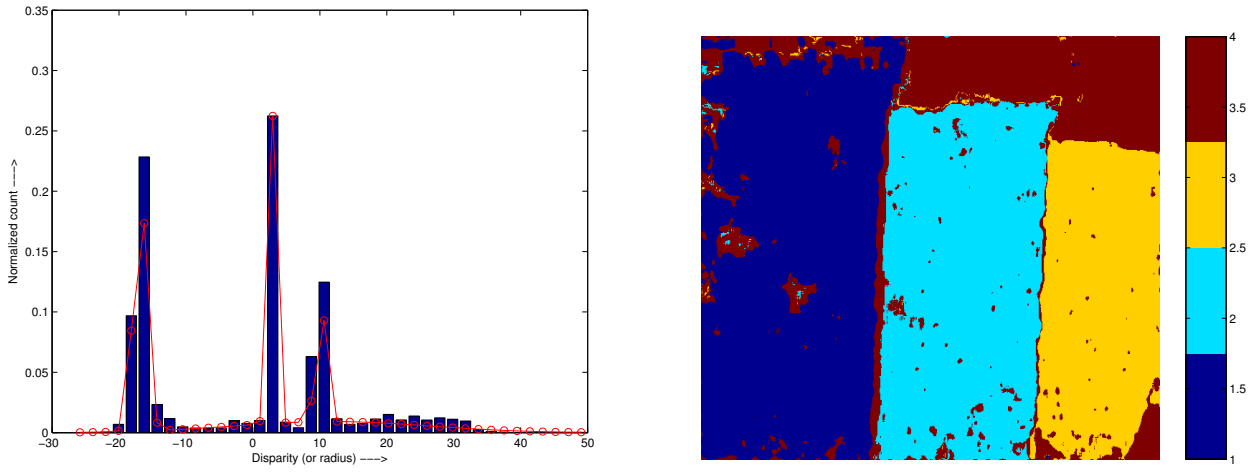
The methodology outlined in Chapter 5 stressed the importance of the epipolar line search, citing the need for improved computational speed and the advantage of exploiting the pattern of the moving-aperture. Therefore, for comparison purposes, optical flow was estimated from the same set of images in this example, using the pyramidal implementation of Lucas-Kanade method [65]. The corresponding flow maps obtained are shown in Fig. 7.10 and the corresponding disparity map with its distribution are shown in Fig. 7.11.

Comparing the flow maps of Fig. 7.10 with those in Fig. 7.4, it can be deduced that the one-dimensional epipolar line search method obtained similar results. This test confirms that the directions of optical flow



(a) (b)

Figure 7.5: (a) Disparity map obtained after noise removal in disparities using a median filter; (b) its corresponding distribution, showing the noise of +5 pixels removed.



(a) (b)

Figure 7.6: (a) Gaussian mixtures modeled on the distribution of disparities, after scaling the values; (b) segmented result of the disparity map using the EM procedure. The integer labels of the regions indicate layers in the scene.

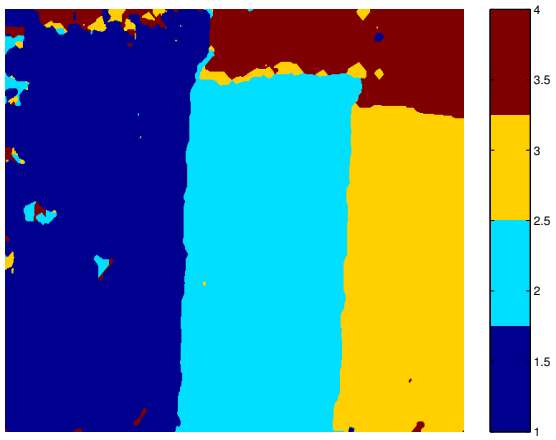


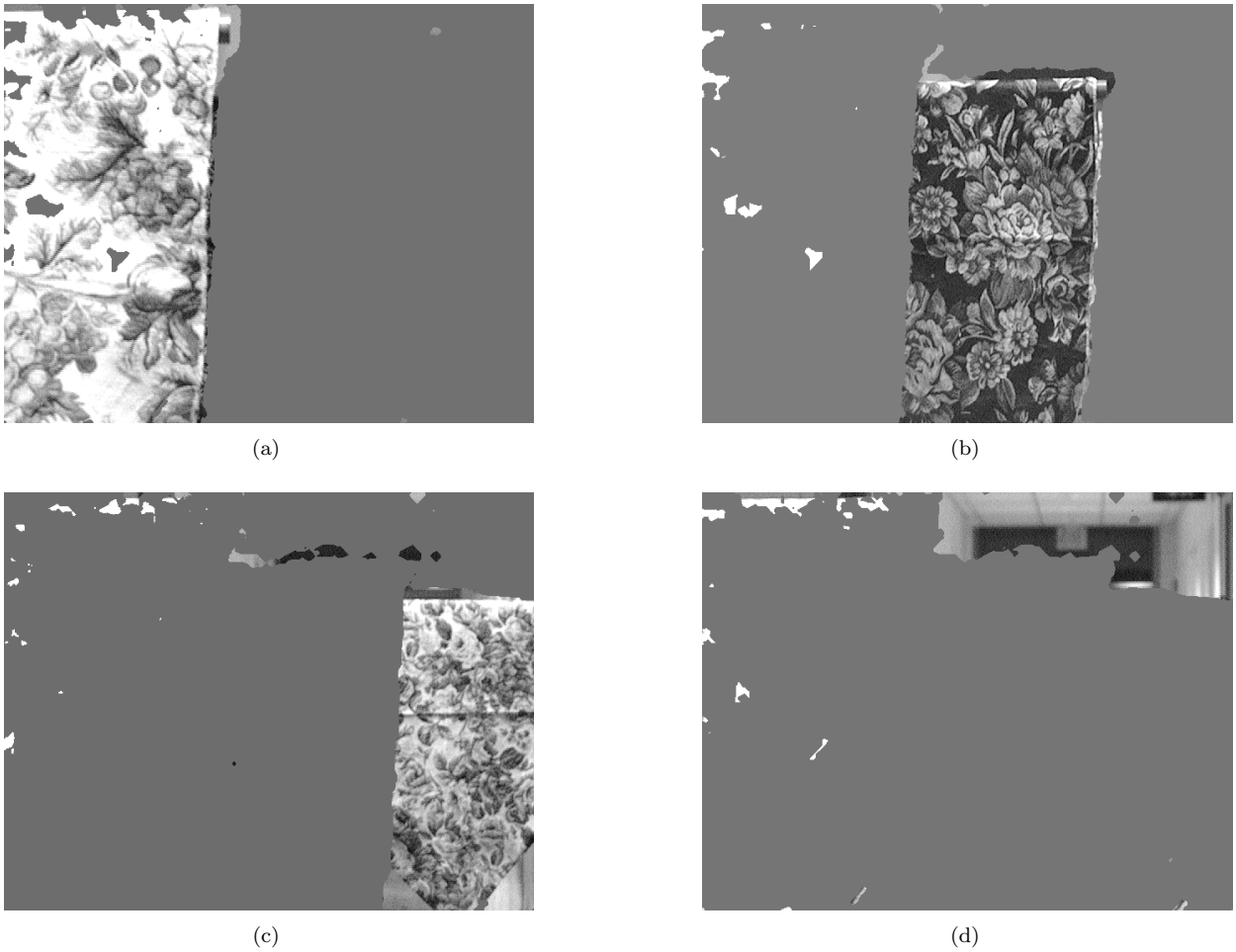
Figure 7.7: Smoothed result of the segmentation, using MRF. The image layers appear smooth, homogeneous regions compared to the result from EM procedure.

estimated in the new approach are almost the same as those really present in the image sequence and hence confirms the validity of the new optical flow method.

Also, the range of disparities estimated from the new approach results, shown in Fig. 7.4 are essentially the same as that shown in Fig. 7.11. Although the underlying circle fitting procedure is the same, if the optical flow vectors were significantly different between the two methods, then the disparities estimated would be different and hence would be clearly visible in the disparity maps. As this was not the case, it is sufficiently clear that both methods produced similar disparity maps.

Table 7.1 lists the various parameters used and time taken for estimating the optical flow, and the disparities, using the epipolar line search method and the Lucas-Kanade method (see Chapter 3). It shows that the new approach used in estimating optical flow reduced the computational cost by 66%. This significant reduction in time taken is a major advantage over the Lucas-Kanade method, with implications toward a real-time implementation of the developed methodology.

A similar flow estimation using 21×21 template window on the same image sequence took approximately 26 minutes using the block matching method described in Chapter 4. It should be recalled that in the block matching approach, the image size had to be doubled to estimate subpixel accurate optical flow and so the images used were twice their original size (640×480).



(a) (b) (c) (d)

Figure 7.8: Image layers estimated from the analysis; (a) layer 1, (b) layer 2, (c) layer3, and (d) layer 4. The image layers 1–3 represent a planar, distance plane from the camera and layer 4 corresponds to the far distant background in the scene. The layers exhibit

Table 7.1: Comparison of time taken to estimate the dense optical flow and disparity maps, in an image sequence.

	Lucas-Kanade	1-D epipolar search	Improvement
Window size (pixels)	9×9	9×9	-
Pyramid level	2	2	-
Search range (pixels)	n/a	4	-
Total image points (pixels)	307200	307200	-
Number of images analyzed	6	6	
Time taken for optical-flow estimation (secs)	311.11	105.68	66.03 %
Time taken for disparity estimation (secs)	120.47	130.6	-

(a)

(b)

(c)

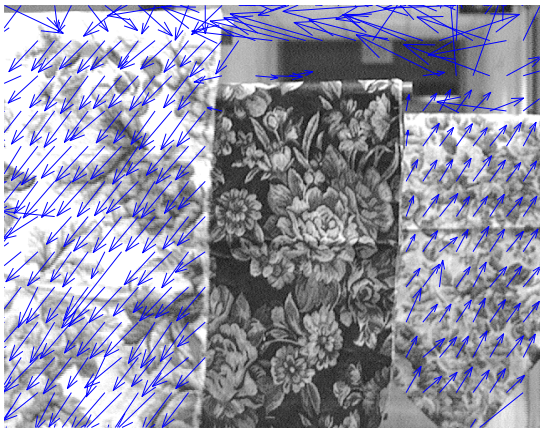
Figure 7.9: (a) Extracted matte of an image layer for an object in focus, (b) an arbitrary background image chosen for compositing[3], and (c) the composited result showing the object from the scene superimposed on the given background image.

7.3.2 Bear in aisle

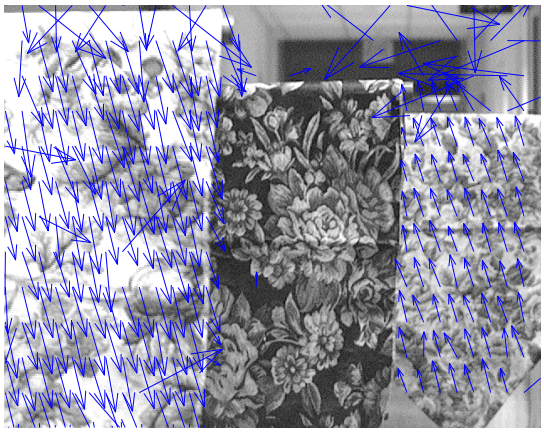
In this image sequence, a stuffed toy bear was placed at a distance, approximately 5.5m from the camera and was imaged in focus, using the f=50mm lens. Two objects were carefully placed in the scene, at distances 3m and 12.5m from the camera and so were located on either side of the focal plane. An image of the scene from the sequence used for analysis is shown in Fig. 7.12(a).

An optical flow map obtained in analyzing the image sequence, seen in Fig. 7.12(b) illustrates that the flow is opposite in either sides of the focal plane. The flow vectors in image regions corresponding to opposites sides of the focal distance clearly show that the underlying relation holds true. Also, the image regions of

(a)

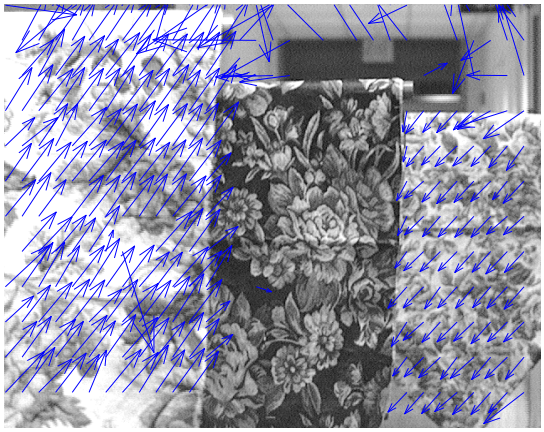


(b)



(c)

(d)



(e)

Figure 7.10: Optical flow maps estimated from the general purpose, Lucas-Kanade method. (a) v_{01}, estimated from image pair $I_0 - I_1$ in the sequence, and other maps: (b)v_{12}, (c)v_{23}, (d)v_{34}, (e)v_{45} from the consecutive image pairs. The vectors are displayed as a sparse version for clarity, although they are estimated for every point in the image.

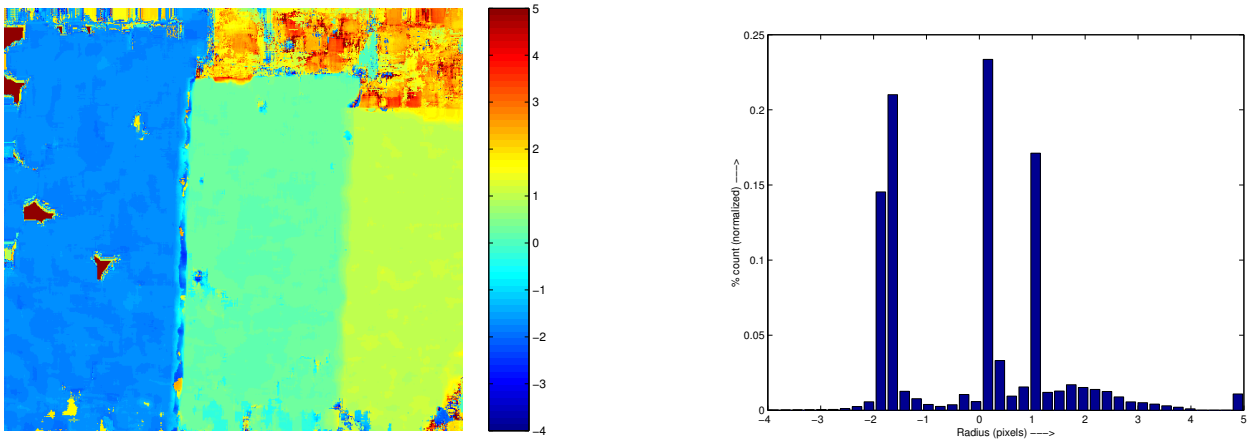


Figure 7.11: Disparity map and its histogram calculated from the optical flow vectors in Fig. 7.10, estimated using the Lucas-Kanade procedure.

the bear correspond to zero flow representing the distance of focus.

The median filtered result of the disparity map estimated from the five optical flow maps is shown in Fig. 7.12(c). Its histogram clearly shows three possible distance layers in the scene – a layer on the plane of focus (zero pixel disparity), a layer beyond the focal distance (at -2.5 pixels disparity) and another layer in front of the focal distance (at 2 pixels disparity). The small peak at 5 pixels in the histogram represents the error in the circle fitting procedure and it mostly corresponds to the plain, featureless regions of the image. The layers estimated from the EM procedure are shown in Fig. 7.12(d) and the corresponding MRF smoothed result is shown in Fig. 7.12(f).

For compositing, the object in focus was chosen and so the binary matte was extracted as given in Fig. 7.13(a). An arbitrary background selected for compositing and the composited result are shown in Fig. 7.13(b) and (c).

Because the layer extraction methodology is based on planar distances from the camera, the result also shows portions of the object holding the bear in the composite. This is not a limitation of the approach but is due to an inherent property of the moving-aperture lens.

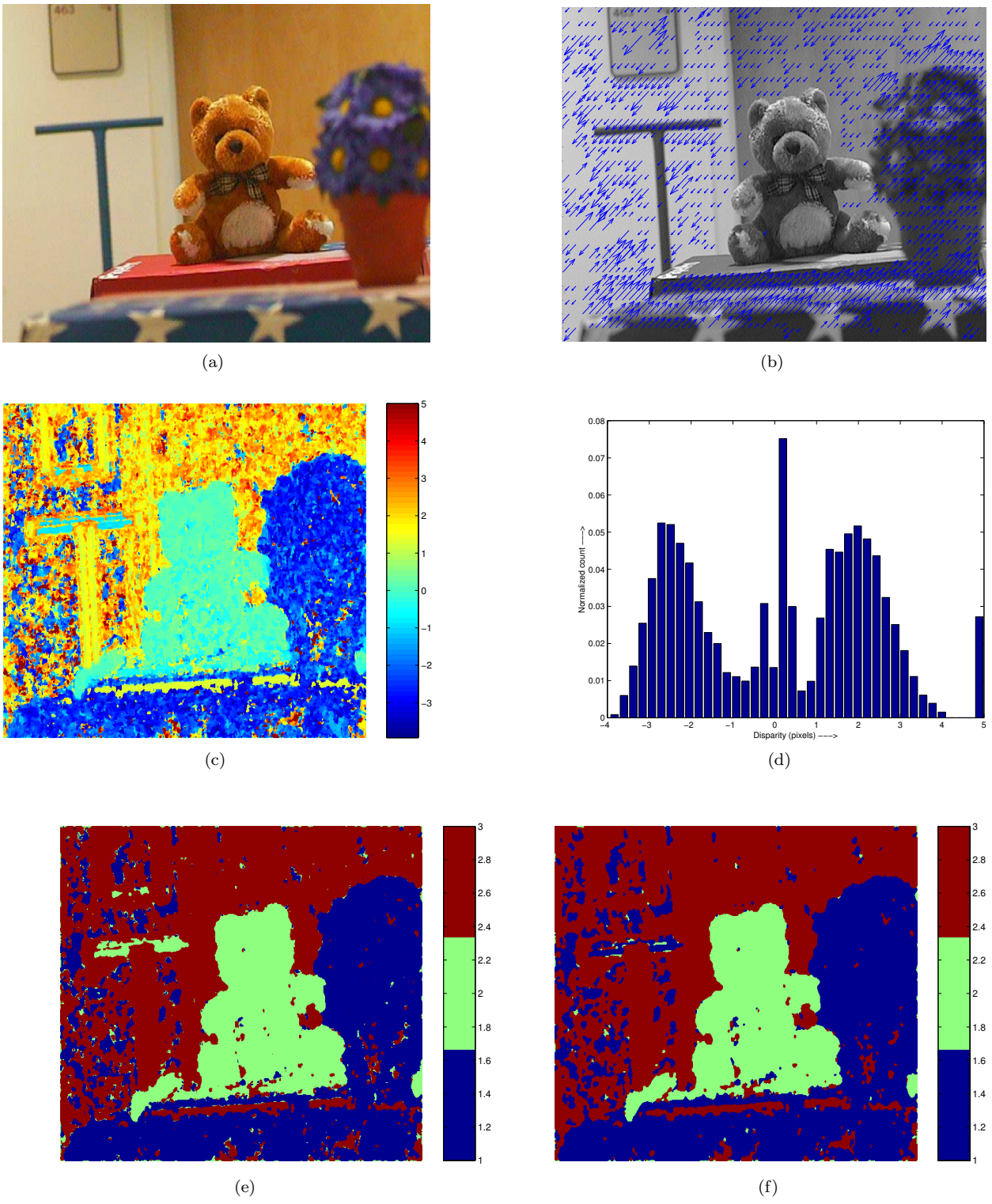
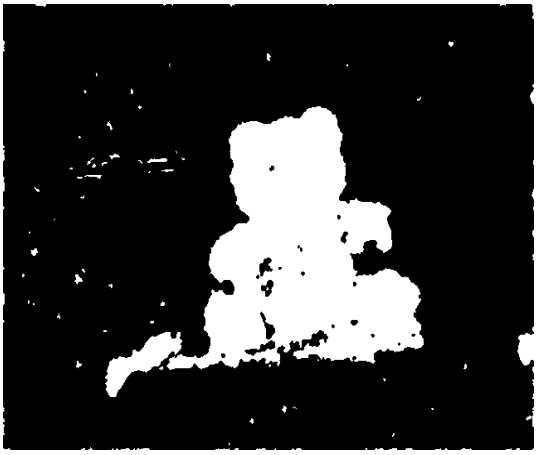


Figure 7.12: Results from the bear-in-aisle experiment: (a) an image from the sequence, (b) optical flow map v_{01}, (c) filtered disparity map, (d) distribution of filtered disparities, (e) layers estimated using the EM procedure, and (f) layers smoothed using the MRF technique.



(a)

(b)

(c)

Figure 7.13: (a) Extracted matte of an image layer for the object in focus, (b) an arbitrary background image chosen for compositing, and (c) the composited result showing an object in the scene superimposed on the given background image.

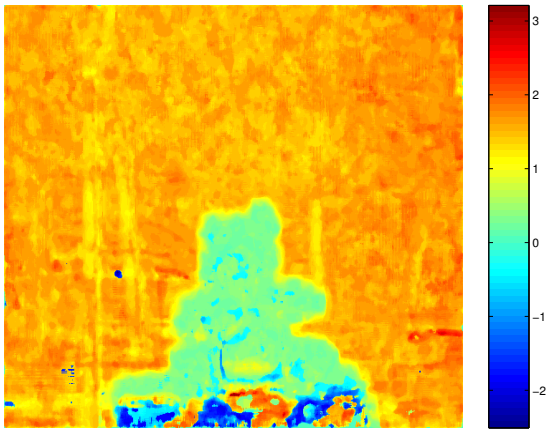
7.3.3 Bear in parking lot

An outdoor scene with the toy bear in focus, with the camera overlooking a parking lot, was chosen to test the compositing methodology, using the f=50mm lens. The representative results obtained from this image sequence are shown in Fig. 7.14 and its composited result is shown in Fig. 7.15.

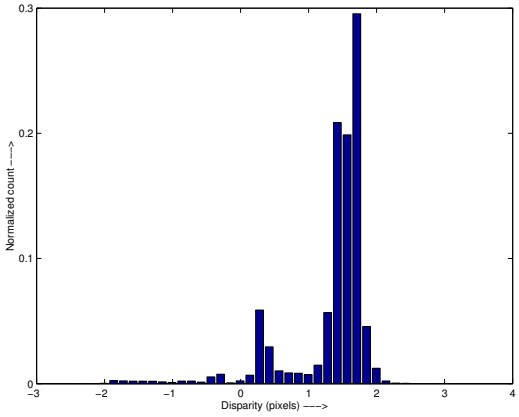
The composited result clearly shows that the bear was extracted from the camera image and overlaid on the chosen background image. It should again be emphasized that this compositing was performed automatically, with no manual input or no external props in the scene.

(a)

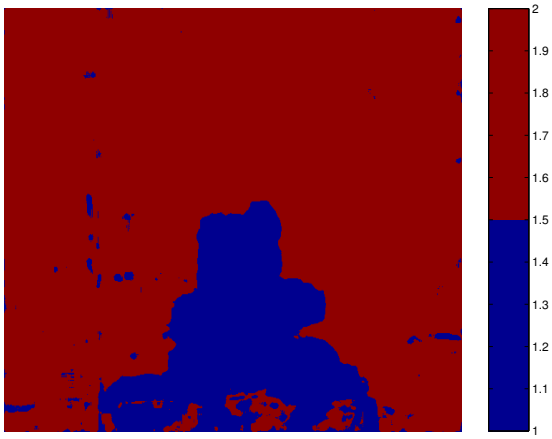
(b)



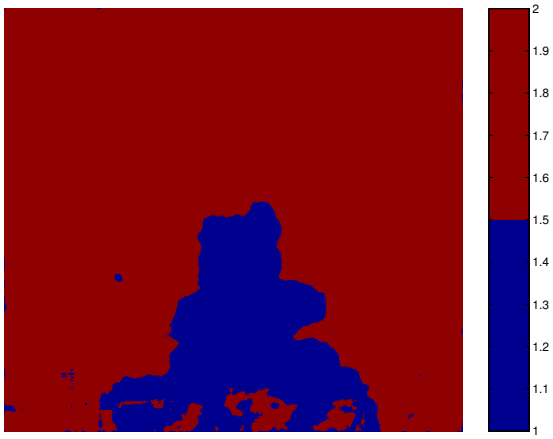
(c)



(d)



(e)



(f)

Figure 7.14: Results from the bear in parking lot experiment: (a) an image from the sequence, (b) optical flow map v_{01}, (c) filtered disparity map, (d) distribution of filtered disparities, (e) layers estimated using the EM procedure, and (f) layers smoothed using the MRF technique.

(a)

(b)

(c)

Figure 7.15: Compositing for the bear in parking lot experiment: (a) extracted matte of the image layer for the bear in focus, (b) an arbitrary background image chosen for compositing, and (c) the composited result showing the bear from the scene, superimposed on the chosen background image.

7.3.4 Person 1

An experiment was designed with a person in an indoor environment, to simulate the widely popular, typical blue-screen composing approach. In this sequence, the person in the scene was in focus with the textured background screen at a distance, slightly more than 2m from the person in the scene. The lens used in this experiment was the f=24mm moving-aperture lens.

The disparity analysis and layers extracted results are shown in Fig. 7.16. The corresponding composited result is shown in Fig. 7.17, which clearly demonstrates very good performance. The matte and the camera image can be scaled in compositing to achieve a desired visual effect. An example of such a scaled image

compositing result is shown in Fig. 7.17(d).

7.3.5 Bear in lab

Another experiment with the stuffed toy bear in an indoor environment was designed to study the layer extraction methodology, this time with the f=24mm lens. The camera was focussed on the bear and so everything else in the scene was beyond this focal distance. The results obtained in this image sequence are shown in Fig. 7.18.

The composited result shows that the bear was extracted from the camera image and was superimposed on the background image. However, as can be seen in the matte, the textureless regions in the camera image, again pose a problem creating "holes" in the matte. Therefore, the composited result shows the other image regions from the camera image "leaking" into the background image.

7.3.6 Person-2

An experiment was conducted with a person in a scene, with no textured background. The moving-aperture lens with f=24mm lens was used, with the person in the scene in focus. The objective was to extract the image of the person from the scene in a separate image layer.

The results obtained for this image sequence are shown in Fig 7.20. From the results, extracting layers proved extremely challenging, given that both the background and person in focus contain many featureless image points. Therefore the estimation of disparity in image regions corresponding to the person (approximately 0.5 pixels) within sufficient accuracy was difficult and hence the layer estimation in this image sequence was nearly impossible.

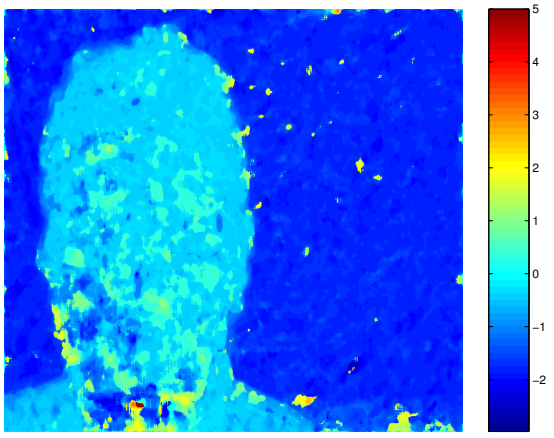
7.3.7 Multiple people

Another experiment with two people in a scene was imaged using the f=50mm lens. In this experiment, the person in front was focussed, with the second person at a farther distance from the focal plane and a textured object farther behind, in the scene.

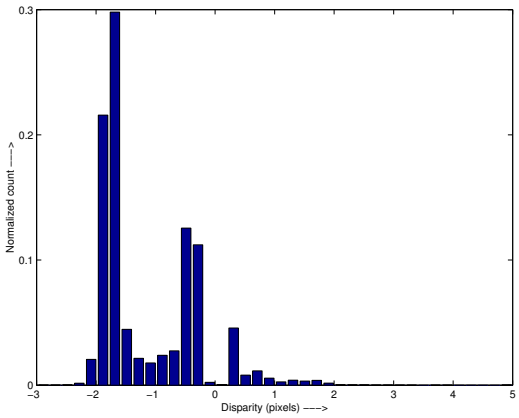
The results obtained from analyzing this image sequence are shown in Fig. 7.21. Again, in this case, extraction of layers proved difficult. A closer investigation revealed that the disparities between expected

(a)

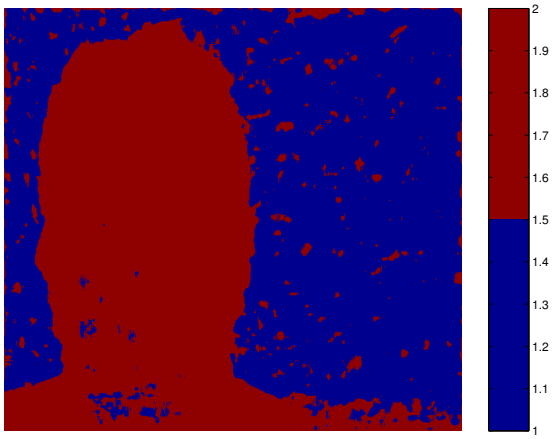
(b)



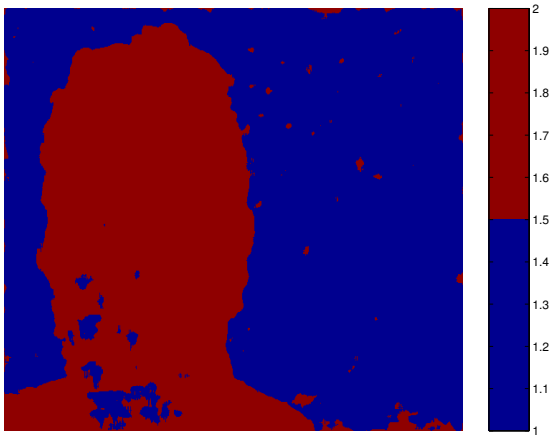
(c)



(d)



(e)



(f)

Figure 7.16: Results from the person-1 experiment: (a) an image from the sequence, (b) optical flow map v_{01}, (c) filtered disparity map, (d) distribution of filtered disparities, (e) layers estimated using the EM procedure, and (f) layers smoothed using the MRF technique.

(a)

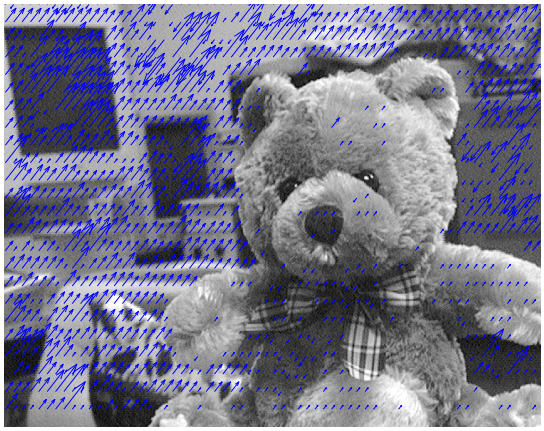
(b)

(c)

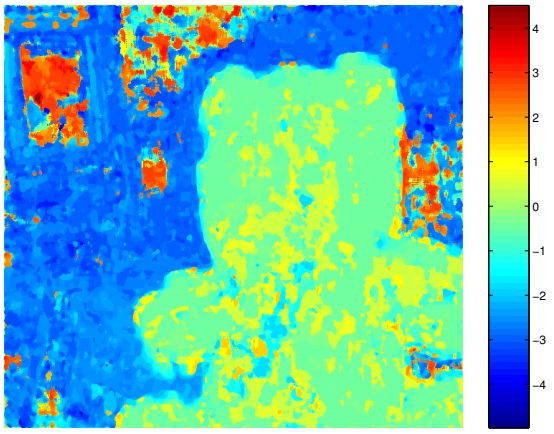
(d)

Figure 7.17: (a) Extracted matte of an image layer for the person in focus from the camera image, (b) an arbitrary background image (from an image sequence from NASA.) chosen for compositing, and (c) the composited result showing the person in the scene, superimposed on the given background image. (d) Image compositing with scaling. The scaling operation on the matte and the original image allows resizing of the superimposed object to achieve a desired visual effect.

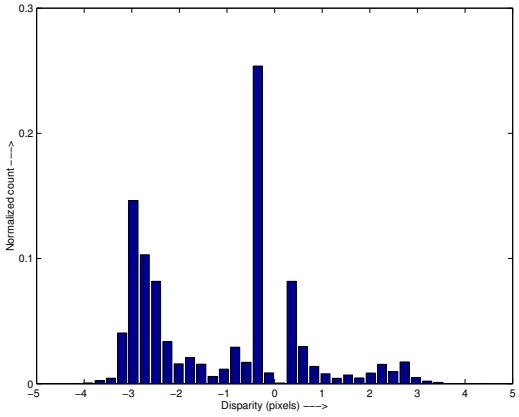
(a)



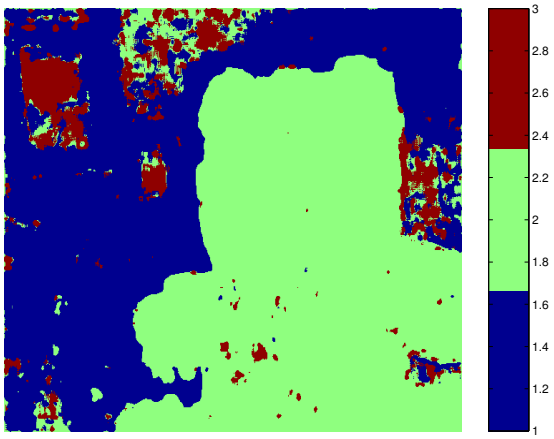
(b)



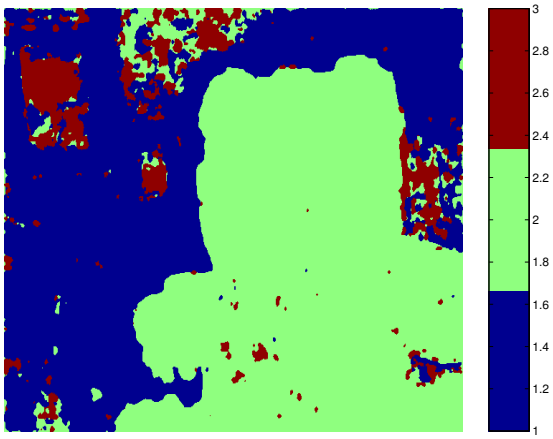
(c)



(d)



(e)



(f)

Figure 7.18: Results from the bear-in-lab experiment: (a) an image from the sequence, (b) optical flow map v_{01}, (c) filtered disparity map, (d) distribution of filtered disparities, (e) layers estimated using the EM procedure, and (f) layers smoothed using the MRF technique.

(a)

(b)

(c)

Figure 7.19: (a) Extracted matte of the image layer corresponding to the bear in focus, (b) an arbitrary background image chosen for compositing, and (c) the composited result showing an object in the scene superimposed on the given background image.

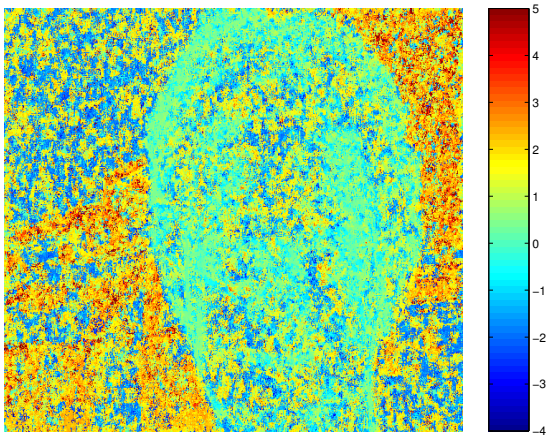
image layers were not sufficient enough to extract the layers.

7.3.8 Observations

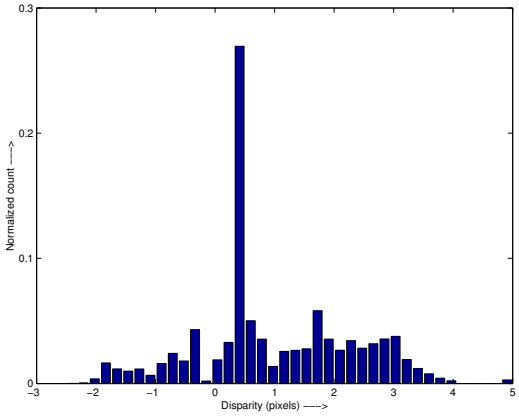
The various image sequences used in testing the developed methodology were captured using the moving-aperture lenses of focal lengths 50mm and 24mm. In particular, the indoor sequences identified as person-1, bear-in-lab, and person-2 were obtained using the f=24mm lens while the others were obtained using f=50mm lens.

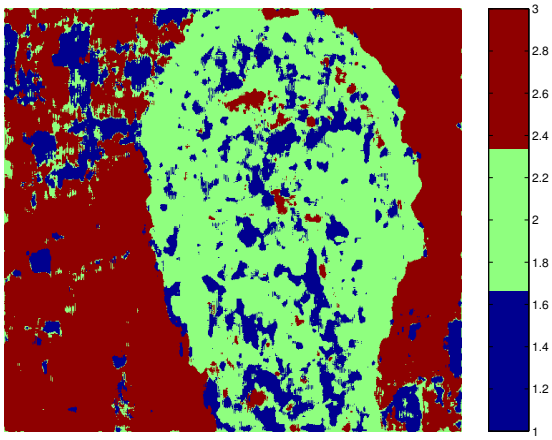
From the results obtained, it is generally observed that the image sequences captured with the f=50mm

(a)

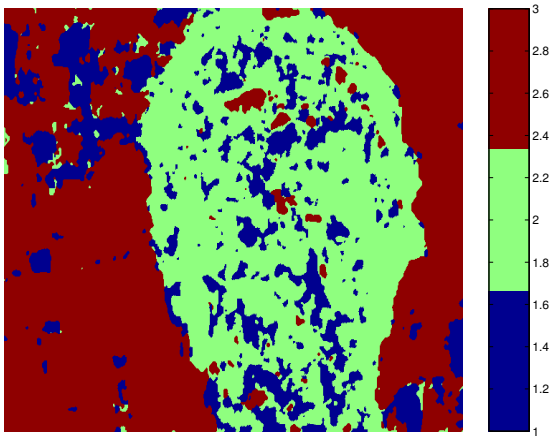


(b)



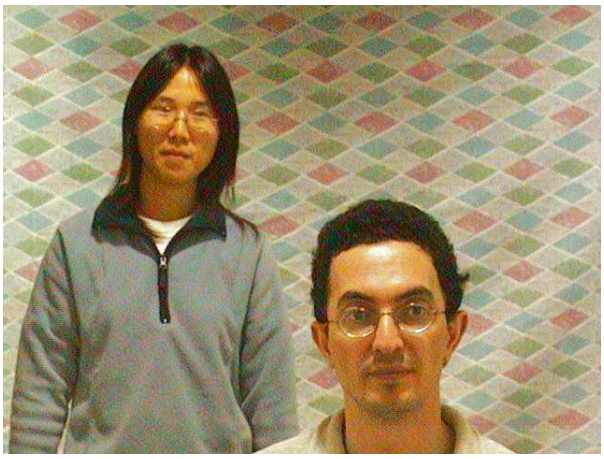


(d)

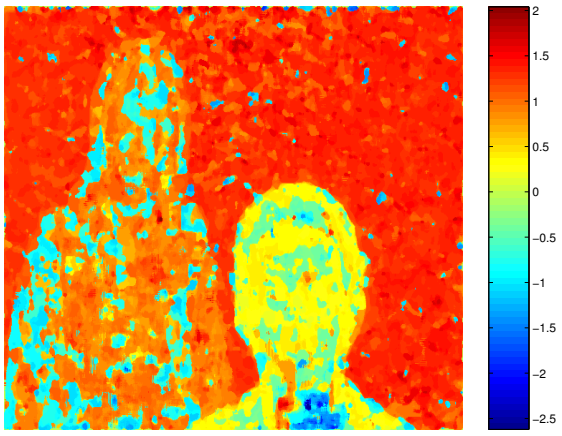


(e)

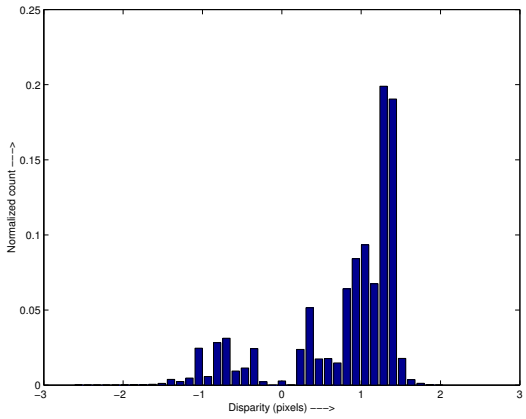
Figure 7.20: Results from the person-2 experiment: (a) an image from the sequence, (b) optical flow map v_{01}, (c) filtered disparity map, (d) distribution of filtered disparities, (e) layers estimated using the EM procedure, and (f) layers smoothed using the MRF technique.



(a)



(b)



(c)

(d)

(e)

Figure 7.21: Results from the multiple-people experiment: (a) an image from the sequence, (b) optical flow map v_{01}, (c) filtered disparity map, (d) distribution of filtered disparities, (e) layers estimated using the EM procedure, and (f) layers smoothed using the MRF technique.

lens were fairly easy to process compared to the sequences captured with f=24mm lens. A main reason is due to mechanical problems of the moving aperture in the f=24mm lens assembly, which has been damaged due to heavy use. Moreover, the maximum radius of the aperture motion possible using this lens was smaller than that with the f=50mm lens. This reduced value of R did not induce large enough disparities in the image sequences to enable extraction of layers easily.

In cases where the automatic layer separation was hard or challenging, the initial estimates of the average disparity in each layer were manually specified based on the values estimated from the automatic procedure. This produced better estimates of the layer and hence a better composited image.

Two parameters which determine the basic requirements for extracting layers from a scene are – sufficient distance (in Z) between objects in the scene, and large radius of aperture motion. The minimum distance separation between objects in 3D depends on the distance of focus of the scene (Z_0) and therefore is not a constant for all types of scenes. A large radius of aperture motion implies a large value of R which produces a large image plane motion. This corresponds to a large optical flow magnitude and hence enables easy distinction of layers in a scene.

Due to de-mosaicing in the camera, the images output from the camera are not exact representations of the sensor information. Therefore the optical flow estimated from the grayscale images (using the luminance portion of the camera output) also are not accurate. This inherent drawback from de-mosaicing can only be eliminated using a grayscale camera or a three-CCD color camera.

The 'holes' in the layer maps appear because of the quality of images output from the camera, inaccuracies in optical flow estimation and disparity estimation. These holes are present in the results shown because no manual or intelligent operation was performed on the layer maps and the entire sequence of operations were automatic. The holes can be removed using some form of manual input which will identify the holes; then a simple region filling operation can be used to eliminate the holes in a layer. With a better, single-chip or three-chip CCD camera, the image quality can be improved. This will lead to reduced errors in optical flow, and disparity estimation and therefore a result with no holes in layers.

The use of robust methods in the methodology needs an explanation. In estimating the aperture positions by analyzing the images, it was noticed that image feature tracking using the Lucas-Kanade method resulted in very large image displacements and so produced outliers in tracking. When this image displacements were used in estimating the angle of epipolar line in an image pair, the traditional least-squares technique did not

perform due to the large outliers; whereas a robust linear regression approach described earlier identified a line in the displacement data and hence the angle of the epipolar line. Therefore the use of robust method in this step proved to be a clear advantage. In the disparity estimation step, the performance of a standard least-squares technique in circle fitting was sufficiently accurate when compared with its robust counterpart. This was perhaps due to the use of epipolar line search in estimating the optical flow, which results in significantly less noise unlike a general purpose optical flow estimation method. A robust implementation of circle fitting involves a lot more computations (in the IRLS step) when compared with the least-squares approach and hence an increased run-time. Also, the accuracy gained in circle parameters with a robust circle fitting approach in this application is not very high or important. Therefore a robust circle fitting method was not utilized in the layer extraction methodology.

7.4 Range Estimation

The moving-aperture model formulated in Chapter 5 and the relations derived to express the real-world distances of objects, based on the image displacements, can be used in passive range estimation applications.

An experiment was designed with four objects at four distinct distances – 2m, 3m, 4m, and 10m from the camera. The camera was focussed on the 4m object. To enable easy calculation of displacements in the image, the objects were chosen as high-contrast image features, as shown in Fig. 7.22. The values on the axes indicate the row and column positions of the image.

From the image sequence, two images corresponding to the maximum distance ($2R$) between the aperture positions in the sequence were chosen for analysis. This ensures that the disparity (image displacement) in the image pair is maximum and hence allows in estimating the range of objects with reasonable accuracy.

Table 7.2 lists the image plane motion and the corresponding optical flow expected, with the focus at 4m. The CCD scaling constant was estimated from a camera calibration procedure (see Appendix B).

The optical flow vectors (image plane displacements) shown in Fig. 7.22 at specific rowa and column positions of the image are listed in Table 7.3. The last column in the table gives the distance of the corresponding object in the scene. From this table, it can be observed that the optical flow is approximately constant for each distance from the plane of focus. The optical flow corresponding to the object at 4m is not exactly zero, possibly because the camera was not perfectly focussed on the object and so a small amount

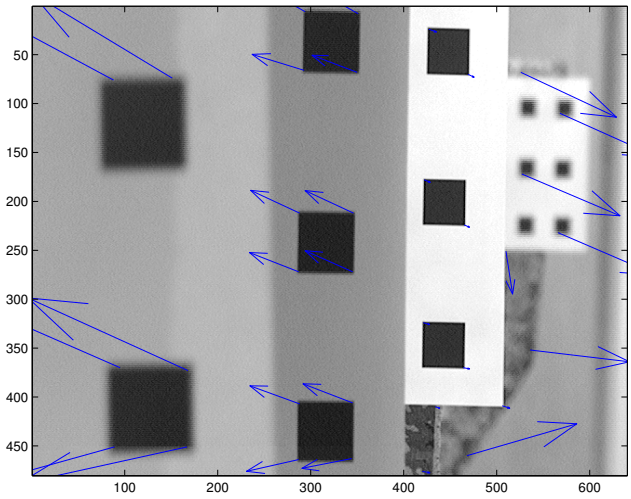


Figure 7.22: Range estimation experiment. Four objects at 2m, 3m, 4m, and 10m (shown left to right) were used and the camera was focussed at 4m. Optical flow values at sparse points in the image are shown in the image. The magnitude of vectors (scaled for display) tend to increase from the plane of focus at 4m, with increasing distance on either side. A few outliers in the optical flow vectors corresponding to featureless points in the image are ignored in the analysis.

of optical flow was estimated in these image regions.

The average optical flow values in Table 7.3 for corresponding image regions are calculated and listed in Table 7.4. By comparing these estimated average values with the expected optical flow values in Table 7.2, we clearly see that both values approximately match. Therefore, the possibility of estimating real world distances of objects, given the optical flow, is clearly evident. Moreover, the accuracy of estimated range is within a range of ± 0.5m, when the other parameters such as focal length f, radius of aperture motion R, distance of focal plane Z_0 are known.

Figure 7.23(a) shows the relation between image plane motion (hence optical flow) and the distances of objects in a scene, with the focus at $Z_0 = 4$m. The graph shows that the magnitudes are opposite in opposite directions of the plane of focus. Also, the graph in Fig. 7.23(b) shows that the magnitudes of motion are not same at a same distance from the plane of focus, on either side. This clearly explains the difference in disparities between objects at same distance from the plane of focus, yet with different magnitudes of optical flow.

Table 7.2: Expected image plane motion and optical flow with focus at 4m. (Parameters used are focal length, f=50mm, radius of aperture motion, R=4mm, and CCD scaling constant, κ=130000 pixels/m.)

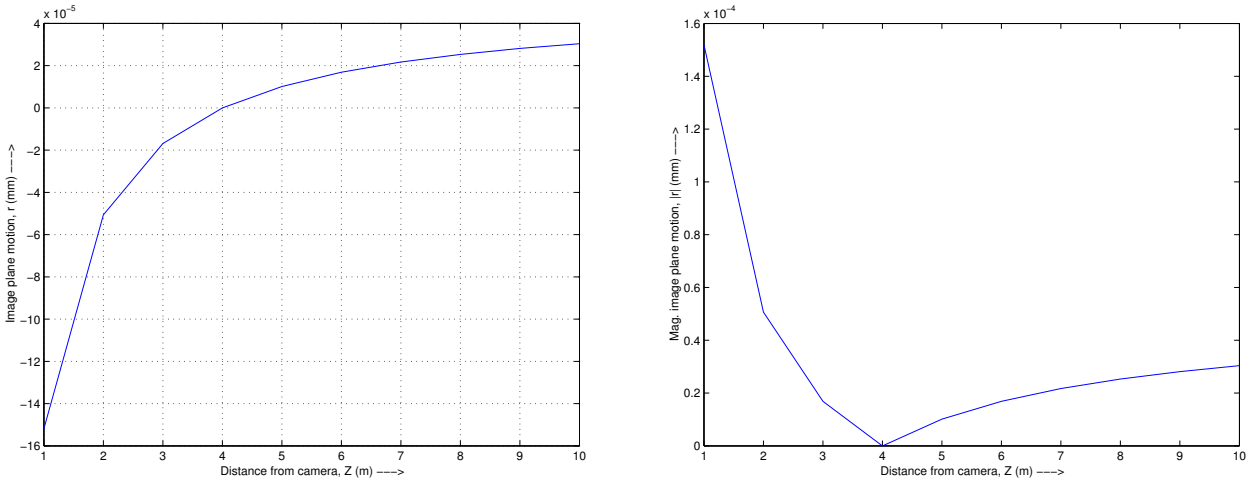
	Distance from the camera, Z_1 (focus, Z_0= 4m)			
	2m	3m	4m	10m
Image plane motion (m)	-0.51E-4	-0.17E-4	0	0.30E-4
Optical flow (pixels)	-6.58	-2.1	0	3.95

Table 7.3: Estimated optical flow from two images I_0 and I_3 in the moving-aperture sequence, using the Lucas-Kanade method.

Image row	Image column	Optical flow magnitude (pixels)	Distance of the corresponding object in scene, from the aperture plane (m)
76	88	-7.43	2
451	89	-6.44	2
370	95	-7.21	2
74	151	-6.25	2
451	167	-6.65	2
373	168	-6.71	2
272	287	-2.02	3
464	287	-2.06	3
212	288	-2.06	3
407	288	-2.03	3
66	293	-2.09	3
6	294	-2.17	3
463	344	-1.95	3
272	345	-1.99	3
406	345	-2.03	3
212	346	-2.04	3
68	350	-1.85	3
8	351	-1.95	3
476	420	0.30	4
324	421	0.26	4
178	422	0.25	4
24	427	0.31	4
410	434	0.18	4
370	464	0.25	4
224	465	0.22	4
70	469	0.27	4
68	526	4.04	10
172	527	4.09	10
352	536	3.85	10
232	566	4.10	10
110	568	4.16	10

Table 7.4: Average flow estimated from the optical flow given in Table 7.3. The averages tend to agree with the expected values in Table 7.2, within an allowable range possible from analyzing the images.

	Distance from the camera, Z_1 (focus, Z_0= 4m)			
	2m	3m	4m	10m
Average optical flow (pixels)	-6.78	-2.02	0.26	4.04



(a) (b)

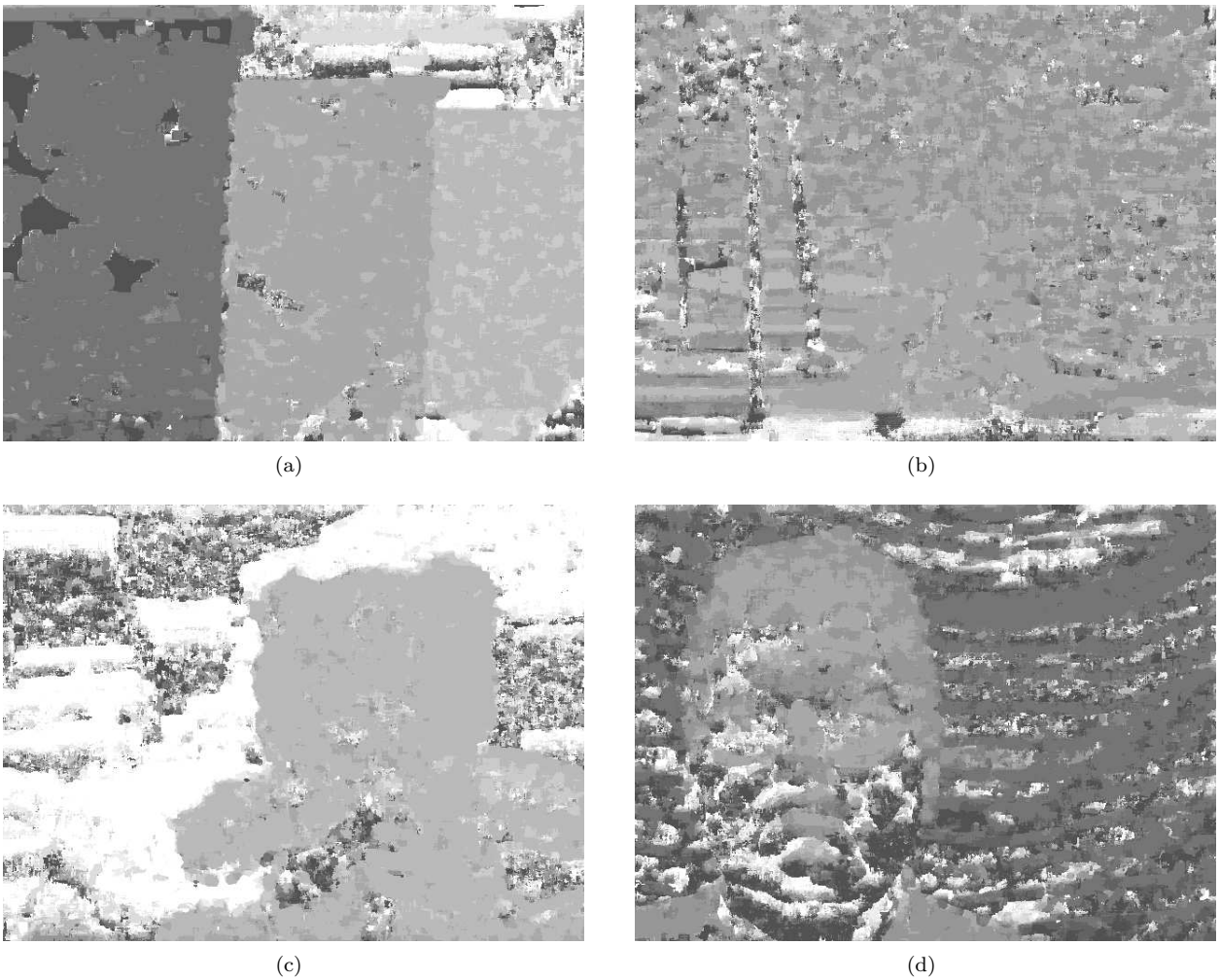
Figure 7.23: Graphs of (a) image plane motion vs. object distances, and (b) magnitude of image plane motion vs. object distances. The graphs are drawn for the case, when the camera is focussed at 4m. Note that the disparity is zero at the plane of focus. Also, the image plane motion (hence optical flow) are not symmetric at equal distances from the plane of focus.

7.5 Layer Extraction and Segmentation - Comparison

The layer extraction approach described and developed in this research can be compared against traditional image segmentation approaches. For this purpose, the readily available MATCH program for dense disparity map calculation, developed by Kolmogorov [108] was used.

Two images corresponding to the maximum possible disparity from each experiment were chosen and used in this analysis. The results of the dense disparity estimation obtained using correlation algorithm minimizing the L2 distance are shown in Fig. 7.24. Similarly, the dense disparity maps obtained using the graph-cut based Kolmogorov-Zabih [108] algorithm are shown in Fig. 7.25.

The correlation based algorithm [108] took 5.8 seconds to calculate the disparity map while the Kolmogorov-Zabih algorithm took 398.31 seconds for the same operation. However, as can be seen, the disparity maps calculated by both algorithms are not satisfactory compared to those of the results from the methodology



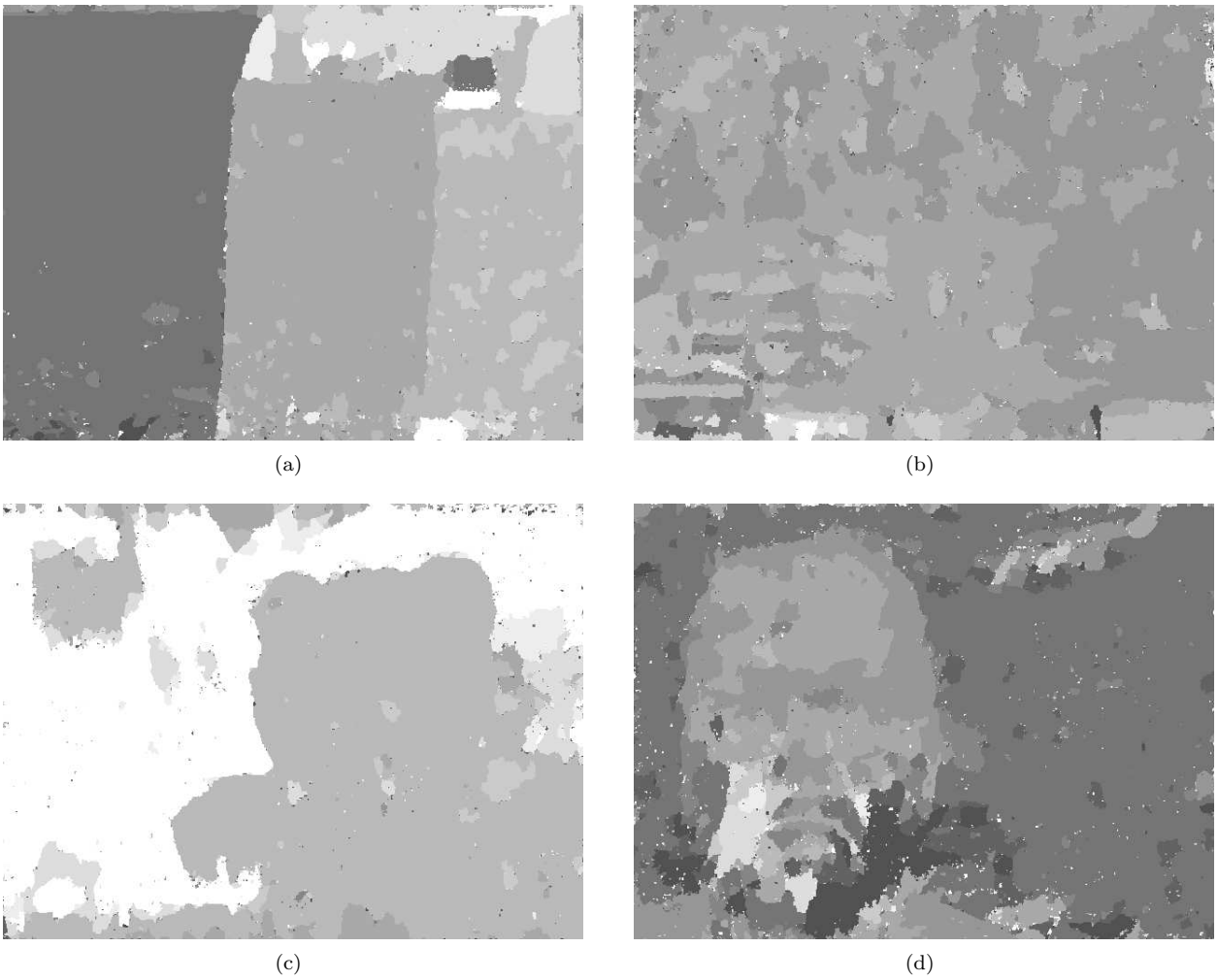
(a) (b)

(c) (d)

Figure 7.24: Dense disparity maps obtained using a correlation based algorithm [108] for (a) textured cloth sequence, (b) bear-in-parking-lot sequence, (c) bear-in-lab sequence, and (d) person-1 sequence. The maps show that these are significantly different from the ideal, expected results.

used in this research.

By observing these dense disparity maps with those from the research, it is clear that traditional stereo matching algorithms perform poorly or fail, when applied on the moving-aperture image sequences. Moreover, they cannot segment the scene based on the planar distance in the scene. Although standard stereo images assume increasing disparity with increasing distances, this is not true in the moving-aperture images where the image regions of an object in focus possesses no disparity. Therefore conventional stereo matching algorithms tend to fail when used to analyze the moving-aperture image sequences.



(a) (b)

(c) (d)

Figure 7.25: Dense disparity maps obtained using Kolmogorov-Zabih algorithm for (a) textured cloth sequence, (b) bear-in-parking-lot sequence, (c) bear-in-lab sequence, and (d) person-1 sequence. From the maps, it is clear that the disparity maps are highly noisy and so do not represent the ideal, expected results.

7.6 Implementation

The entire process of analyzing the the images and generating five optical flow maps took nearly 120 seconds. The circle fitting step for every point in the 640×480 image using five flow vectors took an additional 120 seconds. Both implementations are in C++ on a 1.2 GHz system running Linux. The later post processing was done using Matlab. The EM step converged in less than 10 seconds. The MRF smoothing was the only time consuming operation, which took approximately 5 minutes for ten iterations of smoothing.

With an improved moving-aperture lens design eliminating the mechanical inaccuracies in motion and

better quality of captured images, the estimate of the optical flow can be significantly increased. Therefore the error in disparity estimation can be reduced. This in turn will give better, smooth regions from the EM step and hence can eliminate the need for the MRF smoothing step.

7.7 Conclusion

This chapter presented various results obtained from the developed layer extraction and image compositing methodology. The composited results were obtained using an automatic procedure, with no manual input or initialization of the image layers. The results obtained were compared with the Lucas-Kanade approach and the advantages were highlighted. This chapter also discussed the range estimation application possible using the mathematical model formulated in this research. An experimental result was used to prove the validity and accuracy of this approach. Furthermore, the disparity maps obtained in this research were compared with a graph cut based stereo matching approach and the superiority of the approach in this research was highlighted. In short, the developed layer extraction and image compositing methodology works as desired and designed.

Chapter 8

Conclusion, Applications and Future Work

This chapter concludes the dissertation and discusses some related applications of the research, in addition to the intended image compositing application. Some future directions of the present research work are also suggested.

8.1 Conclusion

This research work has resulted in a novel approach to layer extraction and image compositing. The developed layer extraction method can extract layers automatically from a sequence of 2D images. It does not require any manual input or initialization of layers in an image. Scene layers are extracted based on relative distances of fronto-parallel planes from the focal distance in a scene. The image compositing technique is novel as it eliminates many well known limitations of the widely popular blue screen compositing method. It has been successfully demonstrated to work for several indoor and outdoor scenes, involving people and objects. The compositing method extracts a binary matte from the layers extracted by analyzing 2D images. The mathematical model of the moving-aperture lens and its characterization allowed the development of a passive-range estimation method. This technique was verified to work from a real experiment.

In summary, the following are some significant contributions of this dissertation:

1. a prototype of a novel image compositing system was developed,
2. a new, automatic layer extraction procedure was designed and developed,

3. a passive range estimation procedure was proposed and demonstrated to estimate distances of objects in a scene,
4. a mathematical model of the moving-aperture lens was formulated and verified using experimental data,
5. a new robust circle fitting method was developed, and
6. a comparative study of non-robust and robust ellipse fitting methods was made.

8.2 Applications

The developed image compositing method is a prototype that can be used for creating special visual effects in movies and television broadcasts, where often actors and objects from indoor and outdoor scenes are superimposed on to entirely different, unrelated background images. In addition to this intended application, the system can potentially be used for such tasks as: (1) passive range estimation, (2) surveillance, (3) autonomous navigation, (4) video content manipulation, and (5) video compression.

A secondary application of this research is passive range estimation. Using a camera whose intrinsic parameters are known in advance, the image disparity can be directly used to estimate the real world distance of objects in a scene, using the relations formulated in Chapter 4. An example of this application was given with results from an experiment in Chapter 7.

The layer extraction technique is also capable of segmenting images based on the focus setting of the lens. So in principle, it can selectively extract image regions of objects at the focal distance in a scene. This feature offers a good potential in surveillance applications. An example is a passive monitoring of a secure perimeter, where automatic alerts need to be issued if an object or person intrudes the perimeter. With the moving-aperture lens focussed at a distance of interest and using the developed technique, any object or person crossing the distance of interest will result in a clear peak in the disparity histogram and therefore can trigger an alert.

In a typical autonomous navigation application, a mobile robot follows a path and attempts to avoid obstacles on its path. The obstacle avoidance is generally achieved by using a camera pair or using a time-of-flight active range estimation method (such as radar or ultrasound) for estimating the distances of obstacles on the path. But with the combination of a moving-aperture lens and the developed work, it should be

possible to use only one camera in the mobile vehicle and achieve a similar functionality. The use of one camera also eliminates the complexities involved in a calibrated setup typically necessary with a camera pair and can potentially reduce the weight of the moving platform. Such mobile platforms with passive range sensing capabilities can be used in vehicles employed in hostile environments or for exploration.

The layer extraction approach developed in this research allows the manipulation of a video (image) sequence and hence offers a good control of its content. The approach offers the capability to selectively add or eliminate image objects in a scene and therefore modify the images, to suit the content for intended audience. This application is different from image compositing, as dynamic manipulation of video content based on the audience is the main emphasis here. This application is similar to video object manipulation in MPEG-4.

Another possible application of the developed layer extraction technique is video compression. Generally a stream of video (images) used in commercial applications is viewed by a variety of audiences, in different formats and different devices ranging from large format, high-definition television displays to miniature size, mobile LCD screens. Therefore it is desirable to effectively compress the stream of images for an efficient transmission, yet provide a pleasing experience for the viewer. Using the layer extraction approach, different layers in an image can be tagged to specify the degree of compression and hence the importance of a layer in the image. A compression scheme can use these importance values to provide an efficient way for compression and hence in transmission of video. The tagging approach can be ultimately automated based on the distance in a scene. This can be useful in such domains as sports broadcasting, where an action of interest is typically in a specified range of distances.

8.3 Future Work

The prototype image compositing system developed in this research can potentially evolve into commercial use, if further research is conducted on components of the system. Some of these are desirable modifications while some are highly challenging computer vision problems, which will require many years of active research.

8.3.1 Halo Removal

The composited image results shown in Chapter 6 shows clear artifacts in the result, near the edges of the foreground object. The foreground objects tend to include a small “halo” near the edges of the object.

While the image parallax of few pixels is one cause of this artifact, another possible reason is the error in flow estimation due to varying illumination (intensity) patterns created by the mechanical elements in the moving-aperture lens. A well defined foreground edge is necessary for better quality image compositing. Therefore it is essential to eliminate the "halo" effect and obtain a smooth, sharp edge of the foreground object.

8.3.2 Soft Matte

The composited results in Chapter 6 also show that the foreground and background are not composited seamlessly. Also the matte corresponding to the foreground shows many "holes", through which the foreground and background tend to leakand hence contaminate the result. This is due to the use of a hard matte or binary matte, where the α values in the matte are either 0 or 1. A better choice would be to use a soft matte, in which the α values can vary from 0 to 1. Many commercial, compositing applications provide this feature. The soft matte can be implemented by using a gradient of α values near the matte edges to produce a smooth composited result.

8.3.3 Closed Loop Control

In this research, the moving aperture and image capture in the camera were independent operations and so there was no control of one over the other. As an improvement, a closed loop control of the moving aperture in coordination with the capture mechanism of the camera (synchronization signal) can be implemented to obtain better results from the methodology. When the aperture position is controlled or set to a desired position on its plane and the camera captures at this position, then the aperture location information can be used in the epipolar line search effectively. This would eliminate the need to determine the aperture positions from the image and therefore the error introduced from this analysis. Furthermore, as the aperture position at the beginning of an image sequence will be known, the relative location of objects in 3D from the plane of focus can be determined unambiguously.

8.3.4 Moving Objects

Although this research was limited to stationary scenes, many scenes used for compositing involve moving objects. The presence of moving objects in a scene increases the complexity of the layer extraction process as well as its matte generation. In this case, the object motion must be estimated from the images and then

decoupled from the aperture motion in the lens. When the motion of objects in a scene is considered, it becomes possible for a single object to move across multiple distance planes. If an object undergoes a known motion, this valuable a priori information can be efficiently used in the process. When the object motion is unknown, a predictive approach using a Kalman filter can be utilized to estimate the motion of objects in the scene. This analysis can somewhat simplify the problem of layer extraction.

In a stationary scene, analysis of one complete sequence of frames is sufficient to extract the layers. Whereas in the moving objects case, a longer sequence of images must be analyzed. So the computation time of layer extraction also increases.

8.3.5 Moving Camera

This dissertation has considered a stationary camera only. In some special cases of compositing, a moving camera is used to capture images of a scene for compositing. Generally, the camera is mounted on a trolley and is moved to create some visual tricks on the foreground object. With controlled motion of the camera in the scene, such as a translation, a global motion is induced in the image sequence. The moving-aperture imparts multiple local motion in image regions depending on the distance. Therefore the separation of image motion components becomes difficult. Some existing techniques of layer extraction based on motion segmentation, using a moving camera, can prove to be more advantageous than using the moving-aperture lens. However the existing techniques cannot extract layers based on focus or simultaneously provide the range of objects in the scene.

8.3.6 Color

The developed work was based only on grayscale (luminance) information in image sequences. While this information is sufficient to perform basic compositing, using color information can be advantageous in some cases. An example is a scene where multiple objects are located at a same distance from the camera. In this case, the optical flow corresponding to these objects in the image will be of the same magnitude. An optical flow based segmentation alone can not separate the multiple objects into multiple layers. Therefore using the color information in images during the analysis can help in layer extraction and segmenting the multiple objects in the image.

8.3.7 Aperture Motion

The controller for the moving-aperture lens used in this research allowed changes in the magnitude of motion along the X and Y axes, and these two magnitudes were set equal to create a circular motion of the aperture. The aperture motion other than circular patterns like elliptical, square or other arbitrary patterns can be advantageous in some situations. For example, in a moving object case with an elliptical motion of aperture in a stationary camera, the stationary objects in the scene will also follow an elliptical patter similar to the aperture and hence this relationship can be exploited for layer extraction.

8.3.8 Real-time Compositing

The developed compositing method is presently designed as a off-line application on a PC and takes approximately 5–10 minutes to extract the layers in an image sequence. By the use of specialized image processing hardware, it is possible to significantly reduce the computational time and hence develop a real-time compositing system.

Appendix A

Circle Fitting

Linear regression methods of fitting a line or polynomial to data samples are often encountered in many applications. In general this is a linear problem and hence is easier. Every so often in applications like computer graphics, particle trajectories in high energy physics and lunar craters in astronomy, it is desired to fit a circle to the given data. The problem of fitting a circle to data is comparatively difficult due to the non-linearities involved [84]. In this problem, the goal is, for the given data, find the best estimates of the center and radius of a circle, that best fits the data. Various iterative and non-iterative techniques have been proposed to solve this problem [82, 84, 109]. The choice of a method to solve the problem is influenced by the merits and demerits of each method.

Most of the proposed methods are variations of least-squares approximation technique. A common approach, given a set of $m \geq 3$ points $\{(x_i, y_i) \,|\, i = 1...m\}$, is to determine the radius r and location of the

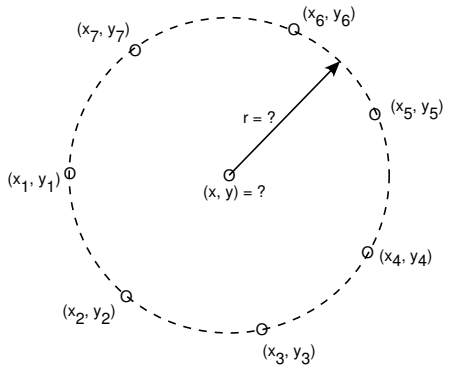


Figure A.1: Circle fitting problem - given seven two-dimensional points, find their center and radius.

center (x_c, y_c) that minimizes the error criterion

$$E = \sum_{i=1}^{m} [\hat{r}_i - \hat{r}]^2 \tag{A.1}$$

where

$$\hat{r}_i = \sqrt{(x_i - \hat{x}_c)^2 + (y_i - \hat{y}_c)^2} \tag{A.2}$$

and the hat notation implies an estimate of the value. It is assumed here that the points are non-collinear.

An iterative method for circle fitting is described in [91]. Taking partial derivatives of (A.1) and setting the resulting equations equal to zero, it is possible to derive the following relations.

$$\hat{r} = \frac{1}{m}\sum_{i=1}^{N} \hat{r}_i \tag{A.3}$$

$$\hat{x}_c = \frac{1}{m}\sum_{i=1}^{m} x_i + \hat{r}\frac{1}{m}\sum_{i=1}^{m}\frac{\partial \hat{r}_i}{\partial x_c} \tag{A.4}$$

$$\hat{y}_c = \frac{1}{m}\sum_{i=1}^{m} y_i + \hat{r}\frac{1}{m}\sum_{i=1}^{m}\frac{\partial \hat{r}_i}{\partial y_c} \tag{A.5}$$

This leads to the following update rules, where the superscript represents the iteration index:

$$\hat{r}^{(k+1)} = \frac{1}{m}\sum_{i=1}^{m} \hat{r}_i^{(k)} \tag{A.6}$$

$$\hat{x}_c^{(k+1)} = \frac{1}{m}\sum_{i=1}^{m} x_i + \left\{\frac{\hat{r}^{(k+1)}}{m}\sum_{i=1}^{m}\frac{\hat{x}_c^{(k)} - x_i}{\hat{r}_i}\right\} \tag{A.7}$$

$$\hat{y}_c^{(k+1)} = \frac{1}{m}\sum_{i=1}^{m} y_i + \left\{\frac{\hat{r}^{(k+1)}}{m}\sum_{i=1}^{m}\frac{\hat{y}_c^{(k)} - y_i}{\hat{r}_i}\right\} \tag{A.8}$$

The values $\hat{x}_c$ and $\hat{y}_c$ are updated until their change is very small. Experimentally, this procedure has resulted in rapid convergence. To assess the quality of fit, the measure

$$Q = \sum_{i=1}^{N} |\hat{r}_i^2 - \hat{r}^2| \tag{A.9}$$

can be used, where $\hat{r}$ is the final estimate of the radius. A low value of Q implies a good measure of fit (MOF), while a high value of Q indicates a poor fit. For a detailed discussion on alternative approaches to circle fitting, [84, 81] are suggested as good references.

Appendix B

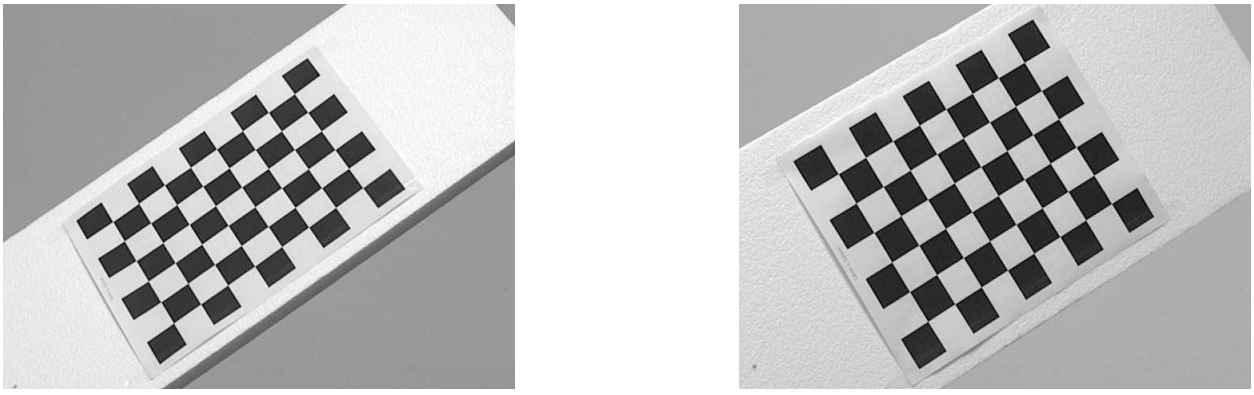
Camera Calibration

Two lenses were used in this research with focal lengths f=24mm and f=50mm. The camera was calibrated with the two lenses for range estimation experiments. The calibration was done using the Camera Calibration Toolbox for Matlab by Bouguet [76], only to estimate the CCD scaling constant (an intrinsic parameter) of the camera, for the two lenses. A sample image used for calibration is shown in Fig. B.1. The output from the calibration procedure for the two lenses is given in the following sections.

B.1 50mm moving-aperture lens

```
Calibration of 50mm MOE lens

Main calibration optimization procedure - Number of images: 7
```



(a) (b)

Figure B.1: Sample images used in the calibration procedure with the 24mm lens.

```
Gradient descent iterations: 1...2...3...4...5...6...7...8...9...10
...11...12...13...14...15...16...17...18...19...20...21...22...23...
24...25...26...27...28...29...30...done
Estimation of uncertainties...done


Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 6472.36438   6544.28654 ] +/- [ 380.42415   292.43467 ]
Principal point:       cc = [ 1152.14311   -760.40585 ] +/- [ 511.32078   529.81174 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]
                            => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ -0.90519   2.27384   0.04586   -0.02104  0.00000 ]
                          +/- [ 0.56603   3.69821   0.04096   0.03781  0.00000 ]
Pixel error:          err = [ 0.43488   0.47208 ]

Note: The numerical errors are approximately three times the standard deviations
(for reference).
```

B.2 24mm moving-aperture lens

```
Calibration of 24mm MOE lens
10 images

Calibration parameters after initialization:

Focal Length:          fc = [ 3040.72058   3040.72058 ]
Principal point:       cc = [ 319.50000   239.50000 ]
Skew:             alpha_c = [ 0.00000 ]   => angle of pixel = 90.00000 degrees
Distortion:            kc = [ 0.00000   0.00000   0.00000   0.00000   0.00000 ]

Main calibration optimization procedure - Number of images: 10
Gradient descent iterations: 1...2...3...4...5...6...7...8...9...10...11...12...
13...14...15...16...17...18...19...20...21...22...23...24...25...26...27...28...
29...30...done

Estimation of uncertainties...done


Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 2971.74170   3063.05024 ] +/- [ 63.40425   64.71360 ]
Principal point:       cc = [ 532.17713   307.07224 ] +/- [ 98.37807   55.90037 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]
                            => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ 0.04115   -0.06385   -0.00425   0.04239  0.00000 ]
                          +/- [ 0.30673   6.44230   0.00716   0.01761  0.00000 ]
Pixel error:          err = [ 0.37541   0.45963 ]
```

```
Note: The numerical errors are approximately three times the standard deviations
(for reference).
```

B.3 Intrinsic parameters

The focal length estimated from the calibration procedure gives the value in pixels has to be converted in for the range estimation application.

For the f=50mm lens, we find that the mean of focal lengths on both X and Y axes are approximately equal to 6500 pixels. Therefore we have the CCD scaling constant as follows:

$$\kappa = \frac{6500}{50 \times 10^{-3}} = 13 \times 10^4 \text{pixels/mm}$$

It should be noted that this value was used in estimating the image plane motion and optical flow in the discussion on range estimation application in Chapter 7.

Also, from the calibration results, it is clear that the skew on the image plane is zero. This indicates that there is effectively no pincushion or barrel distortion, and so there is no need to be concerned of lens distortions.

Bibliography

[1] A. Subramanian, "Image segmentation and range estimation using a moving-aperture lens," Master's thesis, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, Virginia, USA, May 2001.

[2] E. N. Mortensen, *Using Toboggan-based intelligent scissors for image and movie editing*. PhD thesis, Bringham Young University, 2000.

[3] E. N. Mortensen and W. A. Barrett, "Intelligent scissors for image composition," in *Proc. of ACM SIGGRAPH*, pp. 191–198, 1995.

[4] J. Wang and E. Adelson, "Representing moving images with layers," *IEEE Trans. on Image Processing*, vol. 3, pp. 625–638, September 1994.

[5] S. Baker, R. Szeliski, and P. Anandan, "A layered approach to stereo reconstruction," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 434–441, June 1998.

[6] P. Torr, R. Szeliski, and P. Anandan, "An integrated Bayesian approach to layer extraction from image sequences," *Proc. of the IEEE Intl. Conf. on Computer Vision*, vol. 2, pp. 983–991, 1999.

[7] Q. Ke and T. Kanade, "A robust subspace approach to layer extraction," in *Proc. of IEEE Workshop on Motion and Video Computing*, pp. 37–43, December 2002.

[8] H. Tao, H. Sawhney, and R. Kumar, "Object tracking with Bayesian estimation of dynamic layer representations," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 75–89, January 2002.

[9] Y. Wexler, A. Fitzgibbon, and A. Zisserman, "Bayesian estimation of layers from multiple images," in *Proc. of the European Conf. on Computer Vision*, vol. 3, pp. 487–501, 2002.

[10] J. Xiao and M. Shah, "Motion layer extraction in the presence of occlusion using graph cut," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 972–979, June 2004.

[11] Lessons from Innovations Past, "Blue screen on the silver screen." MIT Technology Review, September 2002.

[12] T. Porter and T. Duff, "Compositing digital images," in *Proc. of the 11th Annual Intl. Conf. on Computer Graphics and Interactive Techniques*, pp. 253–259, 1984.

[13] J. Blinn, "Jim Blinn's corner: Compositing part 1: Theory," *IEEE Computer Graphics and Applications*, pp. 83–87, September 1994.

[14] J. Blinn, "Jim Blinn's corner: Compositing part 2: Practice," *IEEE Computer Graphics and Applications*, pp. 78–82, November 1994.

[15] T. Mitsunaga, T. Yokoyama, and T. Totsuka, "AutoKey: Human assisted key extraction," in *Proc. of the 22nd Annual Conf. on Computer Graphics and Interactive Techniques*, pp. 265–272, 1994.

[16] A. R. Smith and J. F. Blinn, "Blue screen matting," in *Proc. of the 23rd Annual Conf. on Computer Graphics and Interactive Techniques*, pp. 259–268, ACM Press, New York, NY, USA, 1996.

[17] P. Vlahos, "Comprehensive electronic compositing system." U.S. Patent 4,625,231-A, November 1986.

[18] G. J. Brostow and I. A. Essa, "Motion based decompositing of video," in *Proc. of the IEEE Intl. Conf. on Computer Vision*, pp. (1) 8–13, September 1999.

[19] M. Ben-Ezra, "Segmentation with invisible keying signal," *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 32–37, June 2000.

[20] S. H. Or, K. H. Wong, K. S. Lee, and T. K. Lao, "Panoramic video segmentation using color mixture models," in *Proc. of the IEEE Intl. Conf. on Image Processing*, vol. 3, pp. 387–390, September 2000.

[21] Y. Y. Chuang, B. Curless, D. Salesin, and R. Szeliski, "A Bayesian approach to digital matting," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 264–271, December 2001.

[22] Y. Y. Chuang, A. Agarwala, B. Curless, D. H. Salesin, and R. Szeliski, "Video matting of complex scenes," *ACM Transactions on Graphics, Proc. of the 29th Annual Conf. on Computer Graphics and Interactive Technique*, vol. 21, pp. 243–248, July 2002.

[23] Y. Y. Chuang, D. B. Goldman, B. Curless, D. H. Salesin, and R. Szeliski, "Shadow matting and compositing," *ACM Trans. on Graphics*, vol. 22, pp. 494–500, July 2003.

[24] P. Peers and P. Dutré, "Wavelet environment matting," in *Proc. of the 13th Eurographics Workshop on Rendering*, pp. 157–166, June 2003.

[25] N. Apostoloff and A. Fitzgibbon, "Bayesian video matting using learnt image priors," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 407–414, June 2004.

[26] MPEG-4 Group, *Coding of moving pictures and video.* ISO/IEC JTC1/SC29/WG11 N4668, March 2002.

[27] B. Horn, *Robot vision.* MIT Press, 1987.

[28] R. Gonzalez and R. Woods, *Digital Image Processing.* Addison-Wesley, 1992.

[29] R. Jain, R. Kasturi, and B. Schunck, *Machine Vision*. McGrawHill, 1995.

[30] University of Southern California, "USC-SIPI image database." http://sipi.usc.edu/services/database/Database.html, 2001.

[31] R. Haralick and L. Shapiro, "Survey: image segmentation," *Computer Vision: Graphics and Image Processing*, vol. 29, pp. 100–132, 1985.

[32] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, pp. 1277–1294, September 1993.

[33] D. S. Zhang and G. Lu, "Segmentation of moving objects in image sequence: a review," *Circuits, Systems and Signal Processing (Special Issue on Multimedia Communication Services)*, vol. 20, no. 2, pp. 143–183, 2001.

[34] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[35] O. Faugeras, Q.-T. Luong, and T. Papadopoulo, *The geometry of multiple images: the laws that govern the formation of multiple images of a scene and some of their applications*. MIT Press, 2001.

[36] R. Jarvis, "A prespective on range finding techniques for computer vision," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 5, p. 122, 1983.

[37] F. Blais, "Review of 20 years of range sensor development," *Journal of Electronic Imaging*, vol. 13, pp. 231–240, January 2004.

[38] N. Yokoya, M. Kanbara, and T. Shakunaga, "Passive range sensing techniques: Depth from images," in *Image Processing Technologies: Algorithms, Sensors and Applications*, Marcel Dekker Inc., 2004.

[39] S. T. Barnard and M. A. Fischler, "Computational stereo," *ACM Computing Surveys*, vol. 14, pp. 553–572, December 1982.

[40] U. Dhond and J. Aggarwal, "Structure from stereo-a review," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 19, pp. 1489–1510, November 1989.

[41] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 993–1008, August 2003.

[42] R. Haralick and L. Shapiro, *Computer and Robot Vision*. Addison-Wesley, 1992.

[43] B. Horn and B. Schunck, "Determining optical flow," *Artifical Intelligence*, vol. 17, no. 1, pp. 185–203, 1981.

[44] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *Intl. Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[45] J. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images-a review," *Proc. of the IEEE*, vol. 76, pp. 917–935, August 1988.

[46] A. Verri and T. Poggio, "Motion field and optical flow: qualitative properties," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 490–498, May 1989.

[47] J. Weber and J. Malik, "Robust computation of optical flow in a multi-scale differential framework," in *Proc. of the IEEE Intl. Conf. on Computer Vision*, pp. 12–20, May 1993.

[48] M. Black and P. Anandan, "A framework for the robust estimation of optical flow," in *Proc. of the IEEE Intl. Conf. on Computer Vision*, pp. 231–236, May 1993.

[49] A. Kumar, A. Tannenbaum, and G. Balas, "Optical flow: a curve evolution approach," *IEEE Trans. on Image Processing*, vol. 5, pp. 598–610, April 1996.

[50] E. Adelson, "Layered representations for vision and video," *Proc. of the IEEE Workshop on Representation of Visual Scenes (in conjuction with ICCV '95)*, pp. 3–9, June 1995.

[51] Z. Oerusuc, *Visual Effects Cinematography*. Focal Press, 1999.

[52] T. S. Perry, "All in the game," *IEEE Spectrum*, vol. 40, pp. 31 – 35, November 2003.

[53] R. Brinkmann, *The Art and Science of Digital Copmositing*. Morgan Kaufmann, 1999.

[54] A. Subramanian, L. R. Iyer, A. L. Abbott, and A. E. Bell, "Segmentation and range sensing using a moving-aperture lens," in *Proc. of the IEEE Intl. Conf. on Computer Vision*, vol. 2, pp. 500–507, June 2001.

[55] A. Subramanian, L. R. Iyer, A. L. Abbott, and A. E. Bell, "Segmentation and range sensing using a moving-aperture lens," *Machine Vision and Applications*, vol. 15, pp. 46–53, October 2003.

[56] M. Halle, "Autostereoscopic displays and computer graphics," *ACM SIGGRAPH Computer Graphics*, vol. 31, pp. 58–62, May 1997.

[57] C. Mayhew, "Vision III single-camera auto-stereoscopic methods," *Society of Motion Picture and Television Engineers (SMPTE) Journal*, vol. 1001, pp. 416–422, June 1991.

[58] C. Mayhew, "Autostereoscopic imaging apparatus and method using suit scanning of parallax images." U.S. Patent - 5510831, 1996.

[59] A. Bacs Jr and C. Mayhew, "Autostereoscopic imaging apparatus and method using a parallax scanning lens aperture." U.S. Patent - 5991551, 1999.

[60] C. Mayhew and A. Bacs Jr., "Parallax scanning using a single lens," in *Proc. of the SPIE*, vol. 2653, pp. 154–160, 1996.

[61] H. Sawhney, "3d geometry from planar parallax," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 929–934, June 1994.

[62] R. Kumar, P. Anandan, and K. Hanna, "Direct recovery of shape from multiple views: a parallax based approach," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 685–688, October 1994.

[63] M. Irani and P. Anandan, "Parallax geometry of pairs of points for 3d scene analysis," in *Proc. of the European Conf. on Computer Vision*, pp. 17–30, 1996.

[64] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. of the Intl. Joint Conf. on Artificial Intelligence*, pp. 674–679, 1981.

[65] J.-Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker," tech. rep., OpenCV Documentation, Microprocessor Research Labs, Intel Corp., 2000.

[66] P. J. Burt and E. H. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, pp. 532–540, 1983.

[67] J. Shi and C. Tomasi, "Good features to track," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

[68] P. Huber, *Robust statistics*. John Wiley, 1982.

[69] P. Rousseeuw and A. Leroy, *Robust regression and outlier detection*. John Wiley, 1987.

[70] C. Tomasi, "Estimating Gaussian mixture densities with EM - a tutorial." Course Handout, 2005.

[71] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incompletet data via the EM algorithm," *Jour. of the Royal Stat. Society (Series B)*, vol. 39, no. 1, pp. 1–38, 1977.

[72] J. Bilmes, "A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," Tech. Rep. ICSI-TR-97-021, Intl. Computer Science Institute, 1997.

[73] R. Dubes, A. Jain, S. Nadabar, and C. Chen, "MRF model based algorithms for image segmentation," in *Proc. of the IEEE Intl. Conf. on Pattern Recognition*, vol. 1, pp. 808–814, 1990.

[74] S. Li, *Markov Random Field modeling in image analysis*. Springer, 2nd ed., 2001.

[75] J. Besag, "On the statistical analysis of dirty pictures," *Journal of Royal Stat. Soc. Ser. B (Methodological)*, vol. 48, no. 3, pp. 259–302, 1986.

[76] J.-Y. Bouguet, "Camera calibration toolbox for matlab," 2004.

[77] E. Adelson and J. Wang, "A stereoscopic camera employing a single main lens," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 619–624, June 1991.

[78] J. Rohály and D. Hart, "High resolution, ultra fast 3-D imaging," in *Proc. of the SPIE - Intl. Soc. for Optical Engg.*, vol. 3958, pp. 2–10, January 2000.

[79] A. Bacs Jr., "Method of using a parallax scanning lens aperture in a range-finding application." U.S. Patent - 5933664, 1999.

[80] I. Kasa, "A circle fitting procedure and its error analysis," *IEEE Transactions on Instrumentation and Measurement*, pp. 8–14, 1976.

[81] N. Chernov and G. Ososkov, "Effective algorithms for circle fitting," *Computer Physics Communications*, vol. 33, pp. 329–333, October 1984.

[82] I. D. Coope, "Circle fitting by linear and nonlinear least squares," *Journal of Optimization Theory and Applications*, vol. 75, pp. 381–388, February 1993.

[83] W. Gander, G. Golub, and R. Strebel, "Least-squares fitting of circles and ellipses," *BIT*, vol. 34, pp. 558–578, 1994.

[84] L. Moura and R. Kitney, "A direct method for least squares circle fitting," *Computer Physics Communications*, vol. 64, pp. 57–63, 1991.

[85] H. Späth, "Least-squares fitting by circles," *Computing*, vol. 57, pp. 179–185, 1996.

[86] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least squares fitting of ellipses," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 476–480, 1999.

[87] Y. Leedan and P. Meer, "Estimation with bilinear constraints in computer vision," in *Proc. of the IEEE Intl. Conf. on Computer Vision*, pp. 733–738, 1998.

[88] P. Rosin, "A note on the least squares fitting of ellipses," *Pattern Recognition Letters*, vol. 14, pp. 799–808, October 1993.

[89] Z. Zhang, "Parameter estimation techniques: a tutorial with application to conic fitting," *Image and Vision Computing*, vol. 15, no. 1, pp. 59–76, 1997.

[90] S. Ahn, W. Rauh, and H. Warnecke, "Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola and parabola," *Pattern Recognition*, vol. 34, pp. 2283–2303, 2001.

[91] D. Eberly, "Least squares fitting of data." http://www.geometrictools.com, 2005.

[92] C. Stewart, "Robust parameter estimation in computer vision," *SIAM Review*, vol. 41, no. 3, pp. 513–537, 1999.

[93] O. Nasraoui, "A brief overview of robust statistics." http://www.louisville.edu/~o0nasr01/Websites/tutorials/RobustStatistics/RobustStatistics.html, 2002.

[94] D. Donoho and J. Huber, "The notion of breakdown point," in *A Festschrift for Erich L. Lehmann* (P. Bickel, K. Doksum, J. Hodges, and B. Jr., eds.), pp. 157–184, 1983.

[95] P. J. Rousseeuw and B. C. van Zomeren, "Unmasking multivariate outliers and leverage points," *Journal of the American Statistical Association*, vol. 85, pp. 633–639, September 1990.

[96] P. Holland and R. Welsch, "Robust regression using iteratively reweighted least-squares," *Communications in Statistics - Theory and Methods*, vol. A6, pp. 813–827, June 1977.

[97] A. Beaton and J. Tukey, "The fitting of power series, meaning polynomials, illustrated in band-spectroscopic data," *Technometrics*, vol. 16, pp. 147–185, May 1974.

[98] P. Rousseeuw, "Least median of squares regression," *Journal of American Statistical Association*, vol. 79, pp. 871–880, 1984.

[99] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[100] P. Rosin, "Ellipse fitting by accumulating five-point fits," *Pattern Recognition Letters*, vol. 14, pp. 661–669, August 1993.

[101] W. A. Stahel, *Robuste schätzungen: infintestimale optimalität und schätzungen von kovarianzmatrizen.* PhD thesis, E.T.H Zürich, Switzerland, 1981.

[102] D. Donoho, *Breakdown properties of multivariate location estimators.* PhD thesis, Harvard University, Boston, MA, 1982.

[103] M. Gasko and D. Donoho, "Influential observation in data analysis," in *Proceedings of the Business and Economic Statistics Section*, pp. 104–110, American Statistical Association, 1982.

[104] L. Mili, M. Cheniae, N. Vichare, and P. Rousseeuw, "Robust state estimation based on projection statistics," *IEEE Transactions on Power Systems*, vol. 11, pp. 1118–1127, May 1996.

[105] R. Safaee-Rad, I. Tchoukanov, B. Benhabib, and K. Smith, "Accurate parameter estimation of quadric curves from grey-level images," *Computer Vision: Graphics and Image Processing*, vol. 54, no. 2, pp. 259–274, 1991.

[106] T. 1394 Trade Association, "Iidc 1394-based digital camera specification," July 25 2000.

[107] B. Bayer, "Color imaging array." U.S. Patent - 3,971,065, 1976.

[108] V. Kolmogorov and R. Zabih, "Multi-camera scene reconstruction via graph cuts," in *Lecture Notes in Computer Science (Proc. of European Conference on Computer Vision)*, vol. 2352, pp. 82–96, May 2002.

[109] W. Yi, "A fast finding and fitting algorithm to detect circles," in *IEEE Intl. Conf. on Geoscience*, 1998.

Vita

Anbumani Subramanian hails from the town of Tiruvannamalai, in Tamilnadu in South India. He got his undergraduate degree in Electrical and Electronics Engineering from Coimbatore Institute of Technology, Coimbatore, India, in 1996. After graduation, he joined IBM and worked as a Software Engineer until 1999. Later, he joined Virginia Tech and got his M.S. in Electrical Engineering in 2001. He continued his studies towards a Ph.D. and graduated in 2005. His research work helped in the creation of two intellectual properties for Virginia Tech. He was awarded an Honorable Distinction for Innovation in the Annual Research Symposium of Virginia Tech in 2005. He also received the Third place in Honor of Outstanding Research Contributions twice, in 2005 and 2001, during the Annual Research Symposium of Virginia Tech. His research interests include computer vision, statistical signal/image processing, biomedical image analysis, and pattern recognition.