ABSTRACT

MOTION ESTIMATION USING OPTICAL FLOW FIELD

by
Jingning Pan

Over the last decade, many low-level vision algorithms have been devised for extracting depth from intensity images. Most of them are based on motion of the rigid observer. Translation and rotation are constants with respect to space coordinates. When multi-objects move and/or the objects change shape, the algorithms cannot be used.

In this dissertation, we develop a new robust framework for the determination of dense 3-D position and motion fields from a stereo image sequence. The framework is based on unified optical flow field (UOFF). In the UOFF approach, a four frame mode is used to compute six dense 3-D position and velocity fields. Their accuracy depends on the accuracy of optical flow field computation. The approach can estimate rigid and/or nonrigid motion as well as observer and/or object(s) motion.

Here, a novel approach to optical flow field computation is developed. The approach is named as correlation-feedback approach. It has three different features from any other existing approaches. They are feedback, rubber window, and special refinement. With those three features, error is reduced, boundary is conserved, subpixel estimation accuracy is increased, and the system is robust. Convergence of the algorithm is proved in general.

Since the UOFF is based on each pixel, it is sensitive to noise or uncertainty at each pixel. In order to improve its performance, we applied two Kalman filters. Our analysis indicates that different image areas need different convergence rates,

for instance the areas along boundaries have faster convergence rate than an interior area. The first Kalman filter is developed to conserve moving boundary in optical flow determination by applying needed nonhomogeneous iterations. The second Kalman filter is devised to compute 3-D motion and structure based on a stereo image sequence. Since multi-object motion is allowed, newly visible areas may be exposed in images. How to detect and handle the newly visible areas is addressed. The system and measurement noise covariance matrices, Q and R, in the two Kalman filters are analyzed in detail. Numerous experiments demonstrate the efficiency of our approach.

MOTION ESTIMATION USING OPTICAL FLOW FIELD

by
Jingning Pan

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Department of Electrical and Computer Engineering

May 1994

APPROVAL PAGE

MOTION ESTIMATION USING OPTICAL FLOW FIELD

Jingning Pan

4/26/94

Dr. Yun-Qing Shi, Dissertation Advisor — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

4/9/94

Dr. Joseph Frank, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

4/7/94

Dr. Edwin Hou, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

4/7/94

Dr. Frank Shih, Committee Member — Date
Associate Professor of Computer and Information Science, NJIT

4/7/94

Dr. Chang-Qing Shu, Committee Member — Date
Member of Technical Staff, Image Business Systems, New York City

BIOGRAPHICAL SKETCH

Author: Jingning Pan

Degree: Doctor of Philosophy

Date: May 1994

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical and Computer Engineering,
  New Jersey Institute of Technology, Newark, New Jersey, 1994
- Master of Science in Electrical Engineering,
  Nanjing Institute of Technology, Nanjing, Jiangsu, China, 1983
- Bachelor of Science in Electrical Engineering,
  Xian Jiaotong University, Xian, Sanxi, China, 1982

Major: Electrical and Computer Engineering

Presentations and Publications:

J. N. Pan and Y. Q. Shi, "A Kalman filter-based Algorithm for 3-D motion estimation from image sequences using UOFF," (Preparing for journal and conference, May, 1994).

J. N. Pan and Y. Q. Shi, "A Kalman filter for improving optical flow accuracy along moving boundaries," (Preparing for journal and conference, May, 1994).

Y. Q. Shi, C. Q. Shu, and J. N. Pan, "Unified optical flow field approach to motion analysis from a sequence of stereo images," (Accepted to Pattern Recognition, April, 1994).

J. N. Pan, Y. Q. Shi and C. Q. Shu, "Correlation-feedback approach to computation of optical flow," (IEEE ISCAS'94, Accepted in December, 1993).

Y. Q. Shi, C. Q. Shu, and J. N. Pan, "Unified optical flow field approach to motion analysis from a sequence of stereo images," (Proceedings of IEEE Signal Processing Society 8th Workshop on Image and Multidimensional Signal Processing, pp. 230-231 Cannes, France, September 8-10, 1993).

C. Q. Shu, Y. Q. Shi, J. N. Pan, and L. Zhou, "A new approach to 3-D position estimation based on unified optical flow field," Proceedings of IEEE Workshop on Visual Signal Processing and Communications, pp. 83 - 86, National Chiao Tung University, Hsingchu, Taiwan. June 5 -7, 1991.

Y. Q. Shi and J. N. Pan, "A pair of stability theorems for complex polynomials," Proceedings of 24th Annual Conf. on Information Science and System, pp. 283, Princeton Uni. March 1990.

J. N. Pan, "Computer recognition of the neurilemoma tissue section image by statistical texture analysis," Proceedings of 5th national conf. on Pattern Recognition and Image Processing, Xian, China, May 1986.

J. N. Pan, "Computer recognition of nerve tumor tissue section images," Master thesis, South East Uni., China, 1986

This dissertation is dedicated to
my family

ACKNOWLEDGMENT

The author would like to thank the dissertation advisor, Dr. Yun-Qing Shi, for his ideas, encouragement and guidance. During the five year stay at New Jersey Institute of Technology, Dr. Shi have helped the author discover the joys of research, the necessity for clear and positive presentation, and the excitement of interaction with peers and colleagues.

Thank the dissertation committee member, Dr. Chang-Qing Shu, for his valuable advice and ideas.

Thank the other members of the dissertation committee, Dr. Joseph Frank, Dr. Edwin Hou, and Dr. Frank Shi, for their comments on the dissertation and their valuable insights.

The author is grateful to the graduate research environment which exists in the Electronics Imaging Center in Electrical and Computer Engineering Department at New Jersey Institute of Technology.

The author wishes to thank Dr. A. Singh for his early print of [44] and some helpful discussion in this research. The author also wishes to thank Professors J. L. Barron, D. J. Fleet and S. S. Beauchemin for their early print of [40] and providing the author with some testing image sequences.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

Over the last ten years, research on the analysis of visual motion has come to play a dominant role within the computer vision community [1]. The ultimate goals of this research are vision systems with the ability to navigate, recognize, and track objects and estimate their speed and direction. Such systems are required for tasks that are too expensive, too hazardous, or too inaccessible for people to perform them. For instances, when robots are used for repetitive manufacturing operations, handling hazardous materials, and exploring the oceans or other planets, the robots need vision systems to sense their environments, build internal models of those environments, and construct and execute plans for achieving their goals. Most of the operations must be performed in a dynamic world in which both the robots (or cameras) and objects move and change shape overtime.

Most of previous works are based on the motion of camera. Apparently, the camera motion is different from object(s) motion [2]. When a camera moves, the whole scene moves. One can use two constant vectors T and R to describe the motion because camera is rigid and its motion can only be translational and (or) rotational. This model cannot be used in object tracking. Nor can this model be used in multi-objects that have different motions. When the camera is stationary and objects move, background is stationary, and motion representation must be a few functions of different objects. When moving objects change their shape overtime, i.e., they are nonrigid, motion representation is more complicated. Such motion should be represented as a function of each point in 3-D world coordinate or 2-D image plane coordinate. In many literatures [2]-[8], the rotation and translation motion of camera are known, according to the motions, depth map, time-to-collision, and other three dimensional parameters are considered primarily.

In this research, we apply stereo vision to not only camera motion but also object motion. In other words, application of this work will enable robots to be able to not only travel through unknown environments but also to track moving objects. Actually, it can be seen that our model can also be used in nonrigid objects as well as rigid objects.

There are basically two different approaches to recovering the structure of object(s) and the relative motion between object(s) and camera(s): the optical flow field approach [9] and the feature correspondence approach [10]. This dissertation research is based on the optical flow approach because the approach can be used widely as long as its accuracy is improved. Since optical flow field describes motion of each pixel, the optical flow approach inherently is not restricted to rigid or nonrigid motion.

Although the research focuses on robot vision, the research of the optical flow field has much wider potentials. In [11], the possibility using optical flow in video coding application was discussed. Moreover, a significant amount of research has focussed on using the optical flow and computer vision techniques in Model-Based video compression [12]-[14]. The model-based video compression coding is to obtain a very low bit rate digital video (about 10 Kbits/s). Image motion analysis has also become active in modeling and analysis of heart wall motion [14].

In this thesis, a new approach to compute optical flow field and a new framework in 3-D motion estimation have been developed and presented. However, before describing our work, we will describe some background and motivation for the research.

1.1 Background and Motivation

The procedure of reconstruction of structure and motion of 3-D object(s) can be illustrated in Figure 1.1.

Firstly, images are taken by a camera and then are digitized by an A/D converter one by one to form an image sequence. With this digital image sequence, the object(s) are reconstructed. In this thesis, the reconstruction means that $X(x,y)$, $Y(x,y)$, $Z(x,y)$, $\dot{X}(x,y)$, $\dot{Y}(x,y)$, and $\dot{Z}(x,y)$ are computed, where X, Y, and Z are coordinates of 3-D object(s), $\dot{X}$, $\dot{Y}$, and $\dot{Z}$ are velocities of X, Y, and Z, and x and y are the coordinates in digital images.

How can these six 3-D fields be computed from the digital image sequence? Generally speaking, in order to recover the 3-D object(s), disparity between a frame and the next frame in the image sequence must be estimated. The disparity is defined as following: If a world point (P) in 3-D space is projected onto the first frame as a pixel with the coordinates (x_1, y_1), and because of motion of the world point (P) or motion of a camera the world point (P) is projected onto the second frame as another pixel (x_2, y_2) that is not equal to (x_1, y_1), then the disparity is the distance between these pixels. Once the disparity is known, 3-D motion can be computed using the perspective projection. We will describe the process next.

1.1.1 Optical Flow Computation

There are two basic approaches to compute disparity. As introduced in the beginning of this chapter, they are the feature correspondence approach and the optical flow approach.

In the feature correspondence approach, the motion of only features, such as boundaries, corners, and other interesting point sets, is computed. However, extracting and establishing the feature correspondence is difficult [15]. In addition, if an object is nonrigid, the approach usually can not be used.

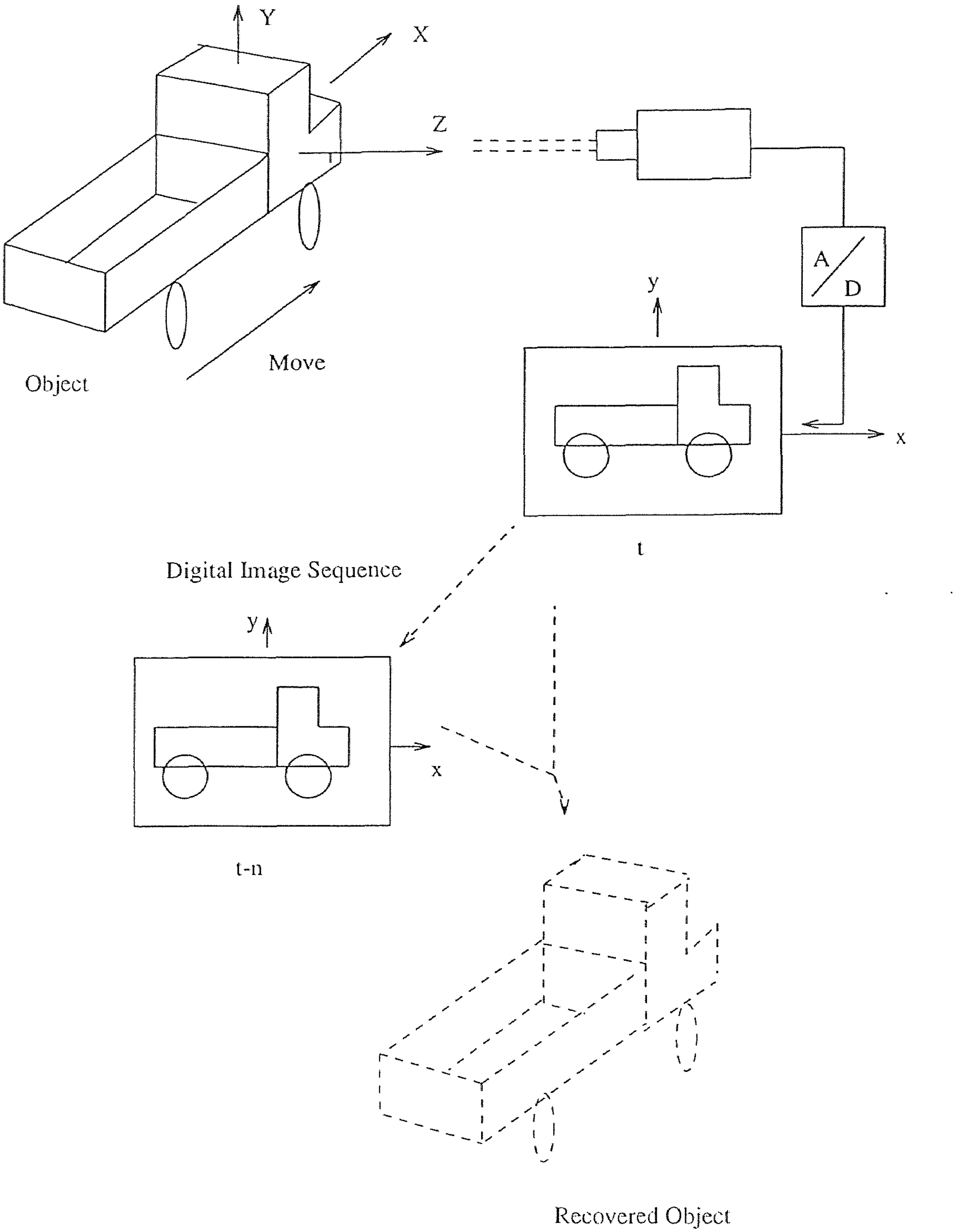


Figure 1.1 Reconstruction of 3-D object(s)

When optical flow approach is applied to compute disparities (or optical flow), the motion field for the entire image is estimated instead of a few interesting point sets. The motion field of images is expressed as

$$\vec{U}(x,y) = \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} \tag{1.1}$$

$\vec{U}(x,y)$ is called as optical flow field. If intensity of the first image is represented by $I_1(x,y,t)$, the second frame image $I_2(x,y,t+\delta t)$ should be equal to $I_1(x+u(x,y)\delta t, y+v(x,y)\delta t, t+\delta t)$. The optical flow field $\vec{U}(x,y)$ can be computed by minimize the difference between the first image and the second image. This technique to compute optical flow field is call "Correlation-based approach." This technique and another technique named as Gradient-based approach will be discussed in detail in Chapter 2.

Basically, the computation of optical flow field has four problems. They are noise, subpixel, aperture, and smoothness.

1. **Noise:** In image acquisition and digitization, noise may be generated. This noise can reduce accuracy of optical flow computation. Therefore some kind of filter is needed. In many literatures, image preprocessing is not considered. However, in those literatures, when a refinement process of optical flow computation is used to propagate neighborhood information, this refinement actually is a kind of filtering. When gradient-based approach is used to compute optical flow, derivative is substituted with difference because images are digital. This digitization is a source of noise.

2. **Subpixel:** We compute optical flow field from a digital image sequence. Digital image means that intensity and coordinates are digitized. If the correlation-based approach is used to compute optical flow field, matching measurement will cause error. Under the assumption that environmental light taking the first frame $I_1(x,y)$ and the second frame $I_2(x,y)$ does not change, this means that

only motion of the 3-D object makes two images different, therefore $I_2(x,y) = I_1(x+u(x,y), y+v(x,y))$. Around a pixel $P(x,y)$ in the first frame I_1, a small window $W_{p1}(x,y)$ is formed. Similarly, the same size window $W_{p2}(x,y)$ is formed around the pixel (x,y) in the second image I_2. When $W_{p1}(x+u, y+v)$ matches $W_{p2}(x,y)$ exactly, u and v at the pixel are obtained. However, if motion of the 3-D object causes that image shifts over subpixel, once the image is digitized, it is impossible to find a W_{p1} that matches exactly a W_{p2}. Therefore, estimation of subpixel image velocity is needed.

3. **Aperture:** Aperture is an inherent problem for every optical flow techniques. When an edge is looked at from an aperture, motion of the edge in some directions is impossible to know. For instance, Figure 1.2 shows an edge. When the edge moves upwards or downwards, the motion is not visible. In gradient-based approach, the aperture problem is represented as that two unknowns are to be solved from one equation. The aperture problem only can be overcome by means of neighborhood information. In the correlation-based approach, the ambiguity can be avoided near corners of an object's image and in textured areas [39].

4. **Smoothness:** When the neighborhood information is applied to refine the optical flow field, can results be improved? Sometimes, the answer is "no." Because this refinement actually is a low-pass filter, the moving boundary or the other point sets with great intensity variation are to be blurred.

Although optical flow computation has many problems, it does not rely on special features such as edge, corner, and the other interesting point set. It can be used widely in many computer vision tasks, for instance, 3-D structure and motion analysis.

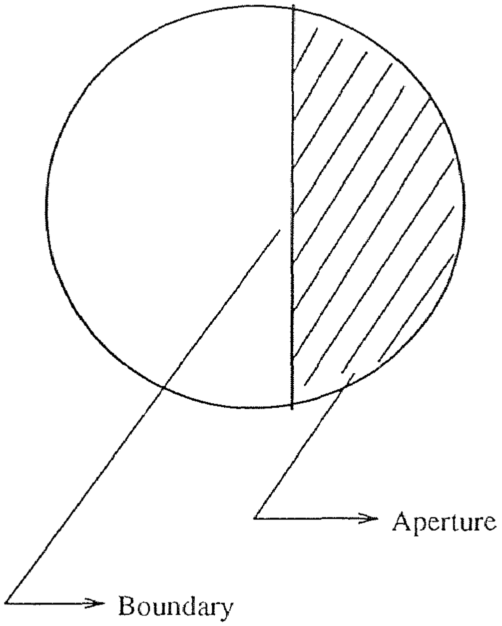


Figure 1.2 A boundary within aperture

Once optical flow field $\vec{U}(x,y)$ is known, 3-D parameters can be computed by perspective projection. That will be discussed next.

1.1.2 3-D Motion Computation

Let's take a look at Figure 1.3, which is to illustrate perspective projection. Assuming that distance D and focal length f are known, once the optical flow u is also known, the ΔX can be obtained from the perspective projection

$$\Delta X = -\frac{uD}{f}. \tag{1.2}$$

However, it is not easily to know D. Actually D is function of (x,y), e.g., $D = Z(x,y)$ that is called "depth." If all (x,y) are considered, it is referred to as depth map. In order to compute $Z(x,y)$, two or more cameras are needed. In other words, stereo images should be used. In many literatures [17], motion of a single camera is used to compute depth map. But, when the velocities of objects are considered,

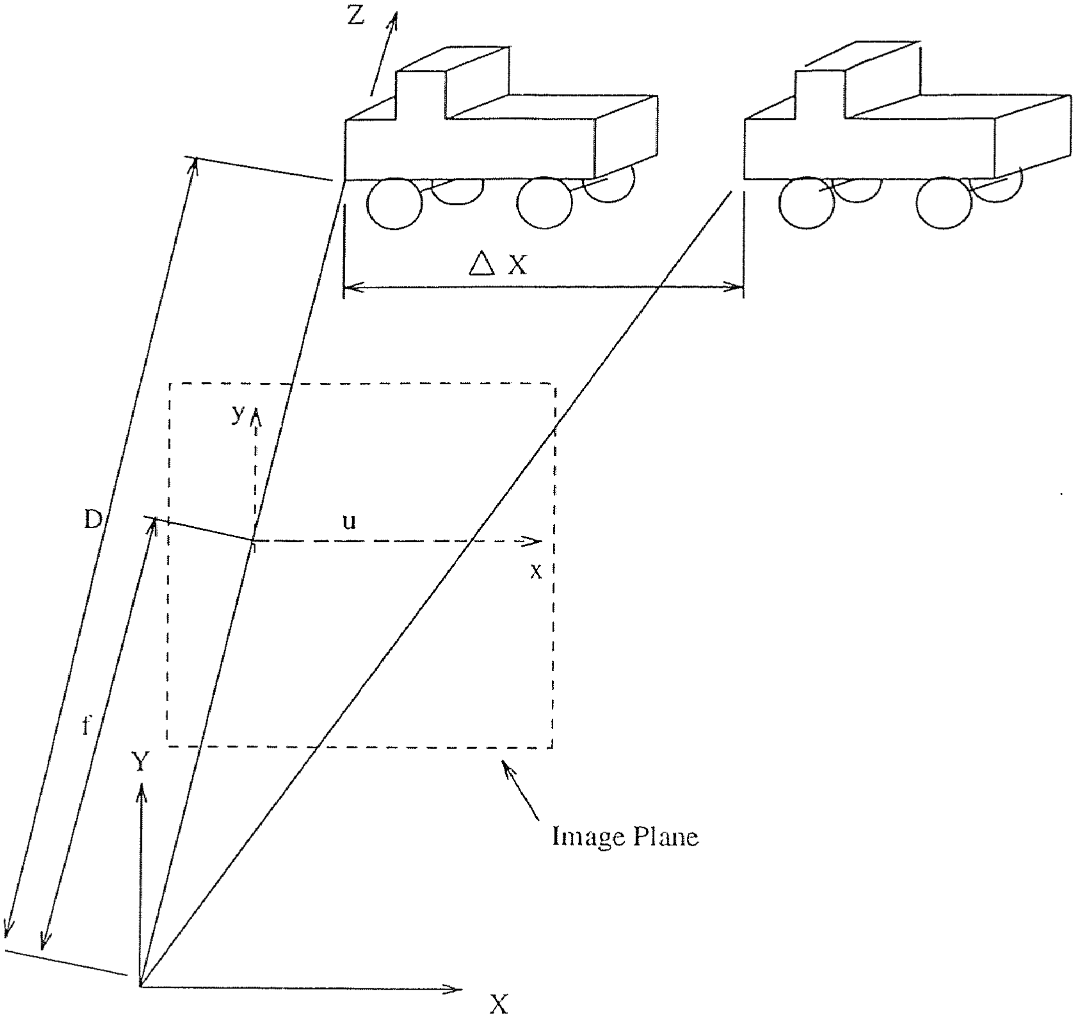


Figure 1.3 Perspective projection

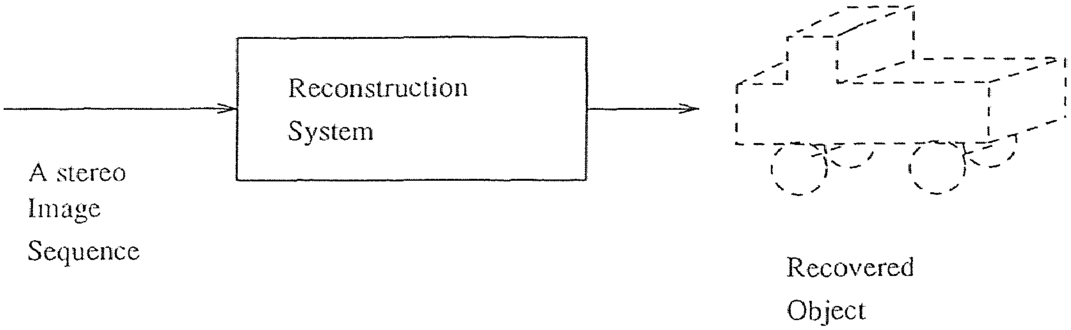


Figure 1.4 Reconstruction system

a stereo image sequence is needed. Therefore, the robot vision problem is to build up a reconstruction system whose input is a stereo image sequence and output is a recovered object. Refer to Figure 1.4. Most of previous works are based on this model. From control system theory point of view, this is an open loop system. It is difficult for such system to obtains accurate outputs, high stability, and high robustness.

In this thesis, we devise a novel system model that can be approximately represented by Figure 1.5.

Obviously, this is a close loop system. According to the recovered object, a virtual image sequence is estimated. By compensating the difference between the real and virtual stereo image sequences, the object can be estimated more accurately.

Dynamic behavior is inherent to the nature of our physical environment. When dynamic behavior of motion is needed, we must consider not only the current state but also a series of previous states. Kalman filtering is an optimal process to fuse the current state and the previous states. However, when objects move, newly visible area(s) may be exposed in image plane. In the new exposed area(s), the previous states can not be used, otherwise error will be caused.

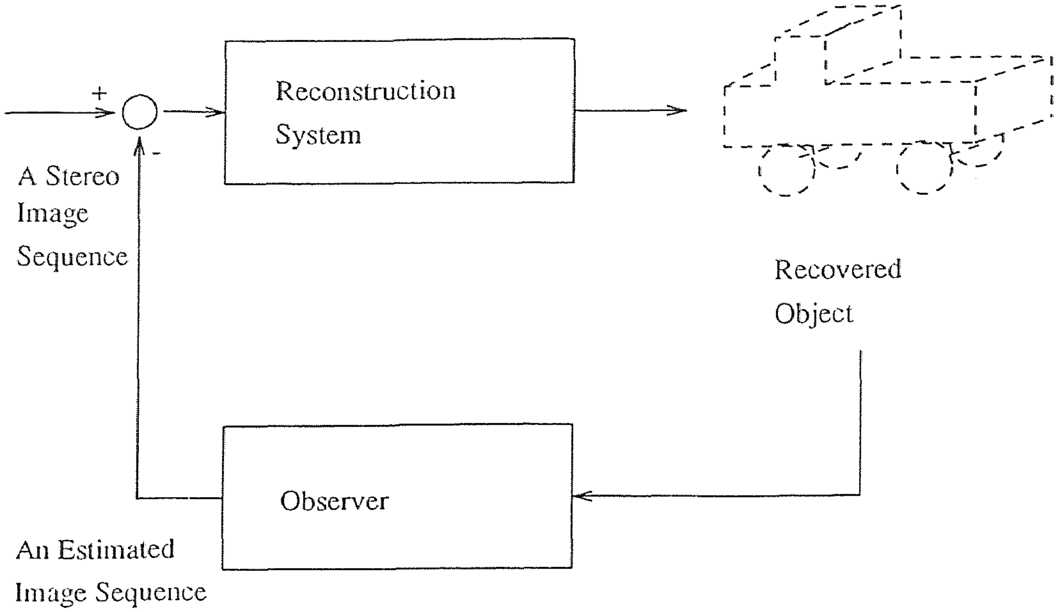


Figure 1.5 Feedback reconstruction system

1.2 Thesis Objective

The objective of this dissertation is to develop a theoretical framework that can estimate motion and position of 3-D objects in dynamic world from a stereo image sequence. Here, dynamic world means that both camera(s) and object(s) can move.

In order to reach the goal, the following four tasks must be fulfilled.

1. Enhancing the accuracy of optical flow field.
2. Improving moving boundaries of optical flow field.
3. Setting up relationship between 3-D motion and 2-D optical flow fields.
4. Using the Kalman filter to optimally fuse current and previous information so that 3-D motion and position can be refined over time when a stereo image sequences is applied.

1.3 Thesis Overview

Chapter 2 reviews related techniques described in the literatures and explains how they fit into our framework and what differences there are between them and our work. Chapter 3 describes our new optical flow field computation approach that is developed based on the correlation-based approach to optical flow computation and feedback technique widely used in control system theory. Accuracy of the optical flow field is considerably increased and the system is more robust. Chapter 4 presents UOFF framework and its application in 3-D motion analysis. A four frame image model is introduced. The model is used to compute position and motion of 3-D objects. Chapter 5 describes a new method to improve the moving boundaries of optical flow field. The method is based on a Kalman filter. Conventionally, Kalman filter is used in incremental fashion. But here, we use Kalman filter to refine optical flow filter based on a fixed set of images. Chapter 6 describes how to use another Kalman filter and a stereo image sequence to improve the estimates of 3-D position and motion, and how to handle newly visible area(s) explosion issue. Finally, Chapter 7 summarizes the major contribution of our research and indicates the directions for future research.

CHAPTER 2

MAJOR RELATED APPROACHES

As noted in Chapter 1, the recovery of 3-D motion and structure information from a sequence of images can be decomposed into two steps: 1. Compute image plane velocities from changes in image intensity values, i.e., compute optical flow. 2. Use optical flow fields to estimate 3-D motion and structure.

2.1 Optical Flow Computation

Most of the current approaches to the computation of optical flow can be divided according to their choice of a measurement.

1. Those that use the continuous variations of intensity over space and time to measure instantaneous image-velocities, e.g., the gradient-based techniques [9], [19]-[22].

2. Those that measure the displacement of points or primitive image tokens between successive frames of a sequence, e.g., the correlation- based matching techniques [25]-[28] and the symbolic-token based matching techniques [29]-[31].

3. Those which measure the spatio-temporal energy of the image intensity function in a small area during a small period of time to determine the direction (and possibly the speed) of motion image points i.e. optical flow [32]-[33].

2.1.1 The Gradient-based Approach

The gradient-based approach is based on the assumption that intensity of light reflected by a point on a surface of an object and recorded in the image remains constant during a short time interval, although the location of the image of that point may change due to motion. This can be mathematically stated as,

$$I(x,y,t) = I(x+u\Delta t, y+v\Delta t, t+\Delta t) \tag{2.1}$$

where $\vec{U} = (u,v)$ is an image velocity vector at the point (x,y) and it is assumed to be constant during the interval $(t, t+\Delta t)$. This equation is called intensity constant equation. $I(x,y,t)$ is the image intensity at point (x,y) in the image at time t. In the limit, when the time interval Δt tends to zero, the intensity-constancy assumption leads to the following equation:

$$I_x u + I_y v + I_t = 0 \tag{2.2}$$

because

$$\begin{aligned} I(x+u\Delta t, y+v\Delta t, t+\Delta t) &= I(x,y,t) + I_x u\Delta t + I_y v\Delta t + I_t\Delta t \\ &+ \mathit{higher\ order\ terms}. \end{aligned} \tag{2.3}$$

Ignoring the higher order terms in Equation (2.3), using Equation (2.1) in Equation (2.3) and taking the limit as $\Delta t \to 0$, Equation (2.2) can be obtained.

The collection of image velocity vectors $\vec{U}$ for the entire image constitutes the optical flow field for the image.

Equation (2.2) embodies two unknowns u and v, and is not sufficient by itself to specify the optical flow uniquely. But, it does constrain the solution. It is possible to compute optical flow for images using the optical flow constraint equation together with an additional assumptions. A popular assumption is smoothness constraint, i.e., motion field varies smoothly in most parts of the image. Horn and Schunck [9] imposed this constraint by minimizing the error in optical flow.

$$\int_D (\Delta I \cdot \vec{U} + I_t)^2 + \alpha^2(\| \Delta u \|_2^2 + \| \Delta v \|_2^2) dX \tag{2.4}$$

where D is integration domain, the magnitude of α reflects the influence of the smoothness term, $\Delta I = (I_x \ I_y)$, and $\| \Delta u \|_2^2 + \| \Delta v \|_2^2$ are the measure of the departure from smoothness in the optical flow. Here, $\| \Delta u \|_2^2$ and $\| \Delta v \|_2^2$ are the squares of the magnitude of the gradient of the optical flow velocity components u and v respectively:

$$(\frac{\partial u}{\partial x})^2 + (\frac{\partial u}{\partial y})^2 \ \& \ (\frac{\partial v}{\partial x})^2 + (\frac{\partial v}{\partial y})^2.$$

They derived an iterative method to calculate the optical flow.

$$\begin{aligned} u^{k+1} &= \bar{u}^k - \frac{I_x[I_x\bar{u}^k + I_y\bar{v}^k + I_t]}{\alpha^2 + I_x^2 + I_y^2} \quad (2.5) \\ v^{k+1} &= \bar{v}^k - \frac{I_y[I_x\bar{u}^k + I_y\bar{v}^k + I_t]}{\alpha^2 + I_x^2 + I_y^2} \quad (2.6) \end{aligned}$$

where k denotes the iteration number, u^0 and v^0 denote initial velocity estimates which are set to zero, and $\bar{u}^k$ and $\bar{v}^k$ denote neighborhood averages of u^k and v^k.

On the boundary the local smoothness of optical flow will not hold. A few people have made efforts to improve optical flow determination along the boundaries, but, it is still a problem.

The primary difficulty for the gradient-based approaches arises from the following facts: They are suitable when the displacements are small with respect to great scale of the image intensity variations. But, sometimes the temporal sampling rate cannot be high because limitation of digitization and resolution. On the other hand, when the scale of the image intensity variation is very small, if the displacements are not enough great, optical flow will not be obtained. Furthermore, above two requirements are not easy to be met simultaneously in real image sequences because the variations of intensities of real images may not be always uniform. About the quantitative error and reliability analysis of the gradient-based approach, Kearney et at. have discussed in detail [34].

2.1.2 The Correlation-based Approach

The correlation-based approach is also referred to the correlation matching approach. It is based on the following principles.

If the intensity functions of two images are $f(x,y)$ and $g(x,y)$ that are two successive frames. Then the cross correlation function between the two images is defined as

$$C_{fg}(\delta x, \delta y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x-\delta x, y-\delta y)g(x,y)dxdy \tag{2.7}$$

where $\vec{\delta} = (\delta x, \delta y)$ is the relative shift between the two images. C is the correlation function, and x and y vary over the two images. The best estimate of $\vec{\delta}$ is determined by maximizing the C over a set of candidate values for $\vec{\delta}$. When a finite sized window from one image is matched against identical windows from the second image, the definition given above is modified such that the integration is limited to the windows. If the image is represented by a discrete pixel array, the above description will still hold except that "point" is replaced by a "pixel" and the integral is replaced by discrete sum. Since our concern is primarily with digitized images, the following discussion uses the discrete formulations.

The discrete formulation of Equation (2.7) is called "direct correlation measure." There are also other related measures that can be used to determine the match. The most commonly used measures for matching are listed below:

1. Direct correlation, in which the image intensity values of the corresponding pixels in the two windows are multiplied and summed.

2. Mean normalized correlation, in which the average intensity of each window is subtracted from the intensity values of each pixel in that window before multiplication and summing.

3. Variance normalized correlation, in which the correlation sum is divided by the product of the variances of the intensities in each window.

4. Sum of squared differences, in which the sum of the square of the differences between the intensities at corresponding pixels is calculated.

5. Sum of absolute differences, which is similar to sum of squared differences, except that the absolute values of the differences are used instead of their squares.

In general, these measures can also be used as estimates of matching strength between two feature vectors; hence, they are useful as matching measures [44]-[45] for many different types of image events.

Based on sum-of-square-differences measure, the correlation matching process is discussed below. For each pixel $P(x,y)$ at location (x,y) in the first image I_1,a correlation - window W_p of size (2n+1)×(2n+1) is formed around the pixel. A search-window W_s of size (2N+1)×(2N+1) is established around the pixel at the same location (x,y) in the second image I_2. The (2N+1)×(2N+1) sample of error distribution is computed using the sum-of-squared-differences as

$$E(\Delta x, \Delta y) = \sum_{i=-n}^{n} \sum_{j=-n}^{n} (I_1(x+i, y+j) - I_2(x+i+\Delta x, y+j+\Delta y))^2 \qquad (2.8)$$

where $-N \leq \Delta x, \Delta y \leq +N$.

In order to handle subpixel issue, many algorithms are developed. Here, Singh's algorithm [44] is discussed. The (2N+1)×(2N+1) sample of response-distribution is computed as follows:

$$R_c(\Delta x, \Delta y) = e^{-kE(\Delta x, \Delta y)} \qquad (2.9)$$

where $-N \leq \Delta x, \Delta y \leq +N$.

One could compute an estimate of image velocity using, for instance, a weighted-least-squares approach. Under the assumption that noise is additive and zero-mean, one could also associate a covariance with this estimate. Quantitatively, the weighted-least-squares based estimate, denoted by $\vec{U} = (u, v)$, is given by

$$u(x,y) = \frac{\sum_{\Delta x} \sum_{\Delta y} R_c(\Delta x, \Delta y)\Delta x}{\sum_{\Delta x} \sum_{\Delta y} R_c(\Delta x, \Delta y)} \tag{2.10}$$

$$v(x,y) = \frac{\sum_{\Delta x} \sum_{\Delta y} R_c(\Delta x, \Delta y)\Delta y}{\sum_{\Delta x} \sum_{\Delta y} R_c(\Delta x, \Delta y)} \tag{2.11}$$

where the summation is carried out over$-N \leq u, v \leq +N$. Because of noise, digitization, and aperture problem, this approach can produce great errors in some pixels of the optical flow field. In order to reduce the errors, smooth constraint or other similar techniques have to be used.

The correlation-based approaches have the following two drawbacks. Firstly, it is inevitable that moving boundary is blurred because of use of smoothness. Secondly, when real image vectors are subpixels, it is difficult to accurately calculate image vectors in this approach. In Chapter 3, we discuss the problem in detail.

2.2 3-D Motion Computation

As noted in Chapter 1, most of the previous works are just to compute depth map and/or time-collision by assuming that the camera motion is known. Some of literatures consider computing depth map and camera translation together [4]. When optical flow field is known, the computation of the motion of the observer is not difficult because motion of observer can be represented by only six quantities, three instantaneous translational components (T_x, T_y, T_z) and three instantaneous rotational components (R_x, R_y, R_z). The (T_x, T_y, T_z) and (R_x, R_y, R_z) are constant with respect to space coordinates. However, when multi-objects and shape change exist, $\dot{Z}(x, y)$, $\dot{X}(x, y)$, and $\dot{Y}(x, y)$, three fields, should be computed. This is one of the differences between our work and the previous works. This section will introduce a few concepts and two examples of motion interpretation.

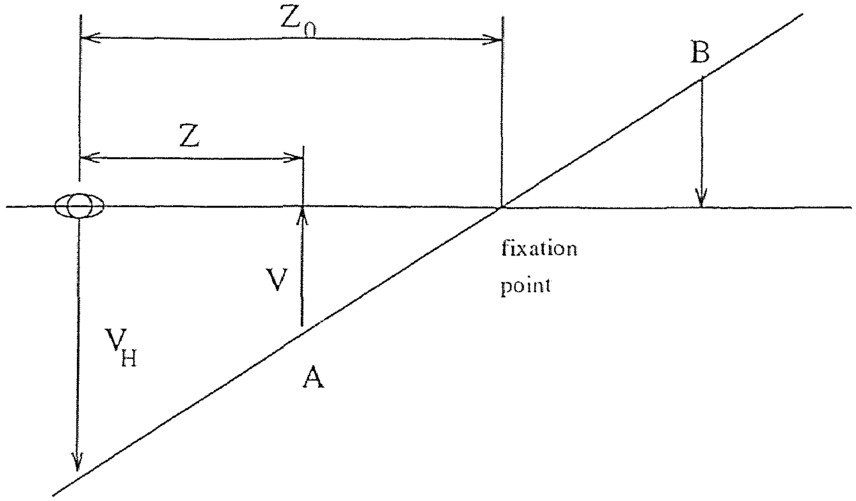


Figure 2.1 Kinetic depth

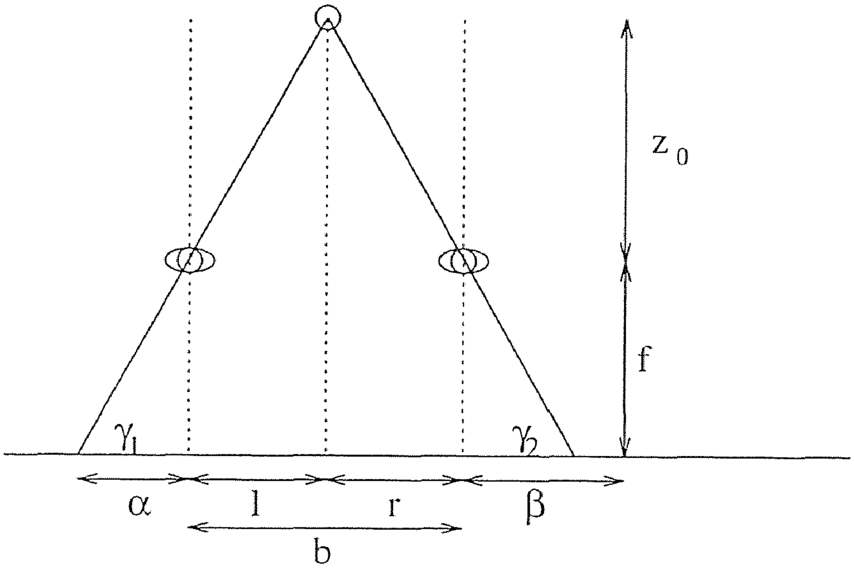


Figure 2.2 The geometry for stereopsis

2.2.1 Motion Parallax

In this section, only the benefits of being able to fixate a point in our 3-D environment is considered. Such a point is a point of reference and points (i.e., B) farther away ("behind") appear to move in the same direction as the viewer, while points (i.e., A) that are nearer ("in front") appear to move away in the opposite direction. Figure 2.1 shows the fixation point.

The motion description is that of motion parallax, kinetic depth is the sensation one gets when horizontally moving one's head while fixating a target. Ballard [35] shows how a fixation point facilitates the derivation of depth by providing an instantaneous origin at (0, 0, z_0), where z_0 is the distance of the fixation point from the viewer.

Vergence geometry is given in Figure 2.2. Given the focal length of the camera f and the base (separation) b, what remains to be established for deriving the depth z_0 is the displacement pair (α, β). The geometry for triangulation is then quite easy to use.

$$\frac{z_0}{l} = \frac{z_0 + f}{l + \alpha}; \ \frac{z_0}{r} = \frac{z_0 + f}{r + \beta}; \ b = l + r \tag{2.12}$$

Hence, one has

$$z_0 = \frac{fb}{\alpha + \beta} \tag{2.13}$$

Since

$$\alpha = f\frac{\cos\gamma_1}{\sin\gamma_1}; \ \beta = f\frac{\cos\gamma_2}{\sin\gamma_2}, \tag{2.14}$$

$$z_0 = \frac{b\sin\gamma_1\sin\gamma_2}{\sin(\gamma_1 + \gamma_2)}. \tag{2.15}$$

Using perspective projection, and the geometry of Figure 2.1, any point in the image plane of coordinates (x,y) is given by

$$x = \frac{-fX}{Z} \tag{2.16}$$

and

$$y = \frac{-fY}{Z}, \tag{2.17}$$

where (X, Y, Z) are the 3-D coordinates. Assume that $(\dot{X}, \dot{Y}, \dot{Z})$ and (u, v) stand for the 3-D (world) and 2-D (optical flow retinal) velocities, respectively. Then differentiating Equations (2.16-2.17) and assuming only translational velocity one obtains

$$
\begin{aligned}
-Zu &= f\dot{X} + x\dot{Z} \\
-Zv &= f\dot{Y} + y\dot{Z}
\end{aligned}
\tag{2.18}
$$

Assuming that we fixate the target, its foveal position implies that the retinal coordinates are approximately zero. Further assume that $\dot{X}$ and $\dot{Y}$ are comparable or greater than $\dot{Z}$, and that $f \gg x, y$. Then

$$\frac{u}{v} \simeq \frac{\dot{X}}{\dot{Y}}. \tag{2.19}$$

The 3-D velocity V can be obtained from Equations (2.18 and 2.19) as

$$|V| = (\dot{X}^2 + \dot{Y}^2)^{1/2} = \frac{Z}{f}(u^2 + v^2)^{1/2}. \tag{2.20}$$

When the head of the observer moves right, a vestibular command motion V_H is parallel and of opposite sign to V, and if one assumes the $\dot{Z}$ is small and $z_0 > Z$, from similar triangles one can then write

$$\frac{|V_H|}{V} = \frac{z_0}{z_0 - Z}. \tag{2.21}$$

Finally, from Equations (2.20) and (2.21) one readily obtains that

$$Z = z_0[1 + \frac{z_0(u^2 + v^2)^{1/2}}{V_H f}]^{-1}. \tag{2.22}$$

For $z_0 < Z$, the sign is negative, as one would expect when moving right and observing targets nearer than the fixation point.

If the approach is explained by single camera motion, we can say that the z_0 can be computed when the camera rotates counterclockwise over $108^o - \gamma_1 - \gamma_2$ with respect to the fixation point, then depth map can be computed when the camera translates over V_H. By the rotation and the translation of the camera, depth map is obtained.

2.2.2 Three-dimensional Motion

The more general case that [36] considers both translational velocity (T) and angular velocity (R) was treated by Waxman and Duncan [36]. For a 3-D pixel, P, the differential motion is defined as

$$\frac{dP}{dt} = -T - R \times P. \tag{2.23}$$

where $T = (T_x, T_y, T_z)$, $R = (R_x, R_y, R_z)$ and

$$\begin{aligned} \frac{dX}{dt} &= -T_x - R_y Z + R_z Y \\ \frac{dY}{dt} &= -T_y - R_z X + R_x Y \\ \frac{dZ}{dt} &= -T_z - R_x Y + R_y Y. \end{aligned} \tag{2.24}$$

If one were to project $P(X, Y, Z)$ onto a unit focal length image ($f = 1$), the retinal coordinates are again given as $(x, y) = (X/Z, Y/Z)$, and the optical flow $(u,\ v)$ is obtained as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + \begin{bmatrix} xy & -(1+x^2) & x \\ (1+y^2) & -xy & -x \end{bmatrix} \begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} \tag{2.25}$$

In many literatures, it is assumed that T and R are motion of a camera. When $R = 0$ and T is known, i.e., the camera does not rotate, but its translation is known, depth map $Z(x, y)$ can be obtained from following equations:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}. \tag{2.26}$$

In those literatures, T_y and T_z are always chosen to be zeroes ($T_y = T_z = 0$), thus

$$Z(x, y) = -\frac{T_x}{u(x, y)}. \tag{2.27}$$

This equation is typically used in the approach of depth from motion. Then, such system is of poor robustness. This will be discussed in Section 4.6.

If T and R are variables with respect to spatial coordinates, the existing approaches can not be used to estimate 3-D motion. In Chapter 4, we will introduce a new approach, that is, unified optical flow field (UOFF) approach. The UOFF can estimate 3-D motion $(\dot{X}(x,y)$, $\dot{Y}(x,y)$, $\dot{Z}(x,y))$.

CHAPTER 3

CORRELATION-FEEDBACK

In this chapter, a new approach using feedback technique is proposed to compute optical flow. It is well-know that feedback is a powerful technique widely used in the field of automatic control. Feedback can make system robust to noise and raise accuracy. In general, this feedback technique can be used for any existing optical flow determination algorithms, say either gradient-based or correlation-based optical flow techniques. As an illustration of our proposal, we choose to work with correlation-based technique because the recently developed correlation-based technique by Singh in [44] has some attractive merits and is suitable for us to apply the feedback technique. Therefore, we call this specific implementation the correlation-feedback algorithm.

The approach can be briefly represented as following. When the first given image that is taken by a camera and the optical flow velocities are known, the first image is shifted at the optical flow velocities, as a result, an estimate of the second given image is obtained. At beginning, the estimated second image is not accurate since the velocities is not accurate. As long as the system is built appropriately, the velocities will be iteratively compensated to make the difference between the recovered second image and the second given image. Since a true optical flow velocity may be a subpixel, a bilinear interpolation is applied to the digital image to obtain a virtual and continuous estimated second image, so that accuracy of estimated optical flow vectors can be increased considerably. In computation, the recovered image intensities are calculated only at the place needed instead of that entire continuous image. That is why it is called as the virtual and continuous image. We will show the approach is generally convergent. At least, when image intensity is a linear function

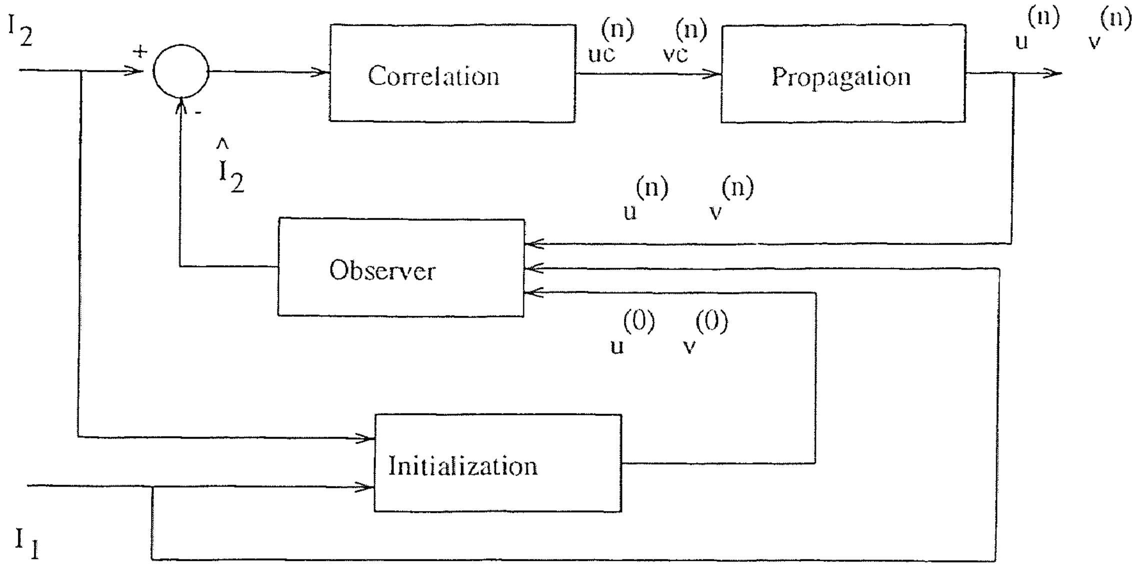


Figure 3.1 A schematic diagram.

of coordinates with free noise, the algorithm must converge to true values. In fact, if noise is reduced, the vicinity of a pixel can be thought of that the above condition is satisfied approximately except the moving boundaries. However, the condition is too strict. Actually, as long as extreme points of the response distributions (refer to the next section) are just true optical flow vectors, the true optical flow vectors can be recovered accurately. However, in the vicinity of a pixel, this condition is approximately satisfied.

Section 3.1 proposes framework and the algorithm. Section 3.2 and 3.3 carefully analyze the algorithm, compare it with previous works and discuss the convergence of it. Section 3.4 demonstrates implementation and experiment to verify the algorithm.

3.1 Proposed Framework

Figure 3.1 shows an overview of the proposed framework. Every block in the figure will be described below in detail.

3.1.1 Initialization

In order to have estimated optical flow field converge to true values and make the convergence faster, an early procedure can be used to compute an initial estimated optical flow field. It is not critical to select the early procedure. But, it is hoped that the procedure is as fast as possible and as accurate as possible. In our experiments, the algorithm of Horn and Schunck [9] is used. This is because that the algorithm is fast and the problem caused by the smoothness is not serious at the first ten to twenty iterations. Furthermore, our experiments show that even if the imposed smoothness constraint causes serious error, subsequent correlation feedback procedure can still make the error to a minimum. Our experiments also show that even if no any early procedure is used and initial optical flow are specified as any value except zero, the algorithm is still convergent, but the convergent speed may be reduced.

After this stage, an initial optical flow field ($\vec{U}^{(o)} = (u^{(o)}, v^{(o)})$) become available.

3.1.2 Observer

As discussed in Chapter 2, it is clear that the correlation-based approach cannot accurately recover subpixel optical flow vectors because image field has been digitized. If the corresponding continuous image field $f(x, y)$ could be known, the accuracy of the recovery would be improved.

The continuous image field $f(x, y)$ should have the following properties. Firstly,

$$f(i_1, j_1) = I_1(i_1, j_1), \tag{3.1}$$

where I_1 is the first digital image, i_1 and j_1 are integer indexes in the first image. That is, on the 2-D grid, the intensities in the continuous image field $f(i_1, j_1)$ are coincident with that of the digitized image $I_1(i_1, j_1)$. Secondly, if $\vec{U}^{(n)} = (u^{(n)}, v^{(n)})$ that is optical flow vector obtained in the nth iteration is equal to the true optical flow vector $\vec{U}^a = (u^a, v^a)$, it is not difficult to see that

$$f(i_2 - u^a(i_2, j_2), j_2 - v^a(i_2, j_2)) = I_2(i_2, j_2). \tag{3.2}$$

where I_2 is the second given digital image, i_2 and j_2 are integer indexes of I_2.

Now, define

$$\hat{I}_2(i_2, j_2) = f(i_2 - u^{(n)}(i_2, j_2), j_2 - v^{(n)}(i_2, j_2)). \tag{3.3}$$

where $\hat{I}_2(i_2, j_2)$ is an estimate of $I_2(i_2, j_2)$. The difference between $\hat{I}_2(i_2, j_2)$ and $I_2(i_2, j_2)$ is affected by the error of $\vec{U}^{(n)}$. In general, the greater the error, the greater the difference. In other words, the difference between $\hat{I}_2(i_2, j_2)$ and $I_2(i_2, j_2)$ contains the information about the accuracy of $\vec{U}^{(n)}$. It is noted that we do not actually estimate the entire continuous image field $f(x, y)$. Only those values needed for optical flow computation, say, $f(i_2 - u^{(n)}(i_2, j_2), j_2 - v^{(n)}(i_2, j_2))$ will be estimated from the first given digital image $I_1(x, y)$ by using a bilinear interpolation technique.

In our work, the bilinear interpolation discussed in [38] is adopted.

$$\begin{aligned} & f(i_2 - u^{(n)}(i_2, j_2), j_2 - v^{(n)}(i_2, j_2)) \\ = \; & (1 - a)[(1 - b)I_1(i, j) + b \times I_1(i, j + 1)] \\ & + a[(1 - b)I_1(i + 1, j) + b \times I_1(i + 1, j + 1)] \end{aligned} \tag{3.4}$$

where $i = int(i_2 - u^{(n)})$; $j = int(j_2 - v^{(n)})$; $a = i_2 - u^{(n)} - i$; $b = j_2 - v^{(n)} - j$; $int(x)$ means that only the integer part of x is retained.

3.1.3 Correlation

Once the interpolated image $f(i_2 - u^{(n)}(i_2, j_2), j_2 - v^{(n)}(i_2, j_2))$ and the second given digital image $I_2(i_2, j_2)$ are available, we can select a correlation measure to search for the best match for a given pixel of $I_2(i_2, j_2)$ in a search-area in the interpolated image $f(i_2 - u^{(n)}(i_2, j_2), j_2 - v^{(n)}(i_2, j_2))$. In this thesis, the sum-of-square-differences (SSD) [44] [39] is used. In essence, for each pixel $P(i_2, j_2)$ at location (i_2, j_2) in the

$I_2(i_2, j_2)$, a correlation window W_c of size $(2m+1) \times (2m+1)$ is formed around the pixel. The search window in our proposed approach is quite different from that used in the correlation-based approach, say, in [44] [39]. The search window is a "rubber" window with variable window size. The size of this window depends on $(u^{(n)}, v^{(n)})$. Let u be a quantity chosen from the following five quantities $(u^{(n)} - \frac{u^{(n)}}{2n}, u^{(n)} - \frac{u^{(n)}}{4n}, u^{(n)}, u^{(n)} + \frac{u^{(n)}}{4n}, u^{(n)} + \frac{u^{(n)}}{2n})$. Let v be a quantity chosen from the following five quantities $(v^{(n)} - \frac{v^{(n)}}{2n}, v^{(n)} - \frac{v^{(n)}}{4n}, v^{(n)}, v^{(n)} + \frac{v^{(n)}}{4n}, v^{(n)} + \frac{v^{(n)}}{2n})$, where n is the number of iterations. Hence, there are 5×5 possible combinations for (u, v). Each of them corresponds to a pixel in the continuous image plane $f(x, y)$, i.e., $(i_2 - u, j_2 - v)$. A correlation window is centered on this pixel. The 5×5 samples of error distribution can be computed by using the sum-of-square-differences. That is,

$$\begin{aligned} E(u,v) &= \sum_{x=-m}^{m} \sum_{y=-m}^{m} (I_2(i_2 + x, j_2 + y) \\ &- f(i_2 + x - u, j_2 + y - v))^2 \end{aligned} \tag{3.5}$$

$$\begin{aligned} u &\in \{u^{(n)} - \frac{u^{(n)}}{2n}, u^{(n)} - \frac{u^{(n)}}{4n}, u^{(n)}, u^{(n)} + \frac{u^{(n)}}{4n}, u^{(n)} + \frac{u^{(n)}}{2n}\} \\ v &\in \{v^{(n)} - \frac{v^{(n)}}{2n}, v^{(n)} - \frac{v^{(n)}}{4n}, v^{(n)}, v^{(n)} + \frac{v^{(n)}}{4n}, v^{(n)} + \frac{v^{(n)}}{2n}\}. \end{aligned} \tag{3.6}$$

The 5×5 samples of response distribution can be computed as follows:

$$R_c(u,v) = e^{-kE(u,v)} \tag{3.7}$$

where k is chosen so as to make R_c be a number close to unity. The optical flow vector derived at this correlation stage is then calculated as follows according to the weighted-least-squares estimation.

$$\begin{aligned} u_c^{(n)}(i_2, j_2) &= \frac{\sum_u \sum_v R_c(u,v)u}{\sum_u \sum_v R_c(u,v)} \\ v_c^{(n)}(i_2, j_2) &= \frac{\sum_u \sum_v R_c(u,v)v}{\sum_u \sum_v R_c(u,v)} \end{aligned} \tag{3.8}$$

It is noted that the equations similar to Equations (3.5, 3.7, 3.8) have been used in [44]. However, in our approach, since $f(i_2 + x - u,\ j_2 + y - v)$ is a 2-dimensional

	-1	0	1
-1	0.25*0.25	0.5*0.25	0.25*0.25
0	0.5*0.25	0.5*0.5	0.5*0.25
1	0.25*0.25	0.5*0.25	0.25*0.25

Figure 3.2 Gaussian mask.

continuous function, u and v are not necessary to be integers. Together with the feedback technique, the accuracy of computing optical flow field can be increased considerably.

3.1.4 Propagation

Except for the vicinity of a motion boundary, the optical flow velocity of neighborhood pixels should be similar to the velocity of central pixel. Therefore, neighborhood information should be used to improve computation of optical flow velocities. That is,

$$\begin{aligned} u^{(n+1)} &= \sum_{x=-N}^{N} \sum_{y=-N}^{N} w(x,y) * u_c^{(n)}(i_2 + x, j_2 + y) \\ v^{(n+1)} &= \sum_{x=-N}^{N} \sum_{y=-N}^{N} w(x,y) * v_c^{(n)}(i_2 + x, j_2 + y) \end{aligned} \tag{3.9}$$

where $w(x,y)$ is a weighting function. We choose a Gaussian mask [24] as the weighting function $w(x,y)$. When $N = 1$, $w(x,y)$ is depicted in Figure 3.2. Similar propagation stage was used in Singh's algorithm [44].

3.1.5 Feedback and Iterations

In each iteration, the output of the algorithm, i.e., the calculated optical flow is fed back into the algorithm. The observation, correlation and propagation discussed above are then carried out. This algorithm can be considered as a feedback control system. From the given digital image $I_1(i_1, j_1)$ and the calculated optical flow $U^{(n)}$ at the nth iteration, a virtual continuous image $f(i_2 - u, j_2 - v)$ is estimated. The difference between another given digital image $I_2(i_2, j_2)$ and the estimated virtual continuous image $f(i_2 - u, j_2 - v)$ is used to adjust optical flow $U^{(n)}$. Based on the updated $U^{(n)}$, the new iteration optical flow $U^{(n+1)}$ is generated in the propagation stage. If the system model is correct, $U^{(n)}$ will converge to the true optical flow. Two given images are repeatedly utilized. As shown next, the algorithm will converge to the true optical flow in general. When $|u^{(n+1)} - u^{(n)}|$ and $|v^{(n+1)} - v^{(n)}|$ are greater than a predefined threshold, we update $\hat{I}_2$ in the observer stage according to the values of $u^{(n+1)}(i,j)$, $v^{(n+1)}(i,j)$ and $I_1(i,j)$. Otherwise, it is said that the algorithm is convergent.

3.2 Convergence Proof

Here, we prove that the correlation-feedback algorithm converges to the true optical flow if the response distribution defined in Equation (3.7) is single-maximum-valued with the maximum being assumed at the true optical flow.

Let us neglect the propagation stage in the proof. Therefore, Equation(3.8) can be rewritten as

$$u^{(n+1)} = \frac{\sum_{i=0}^{4} \sum_{j=0}^{4} R_c(u_i^{(n)}, v_j^{(n)}) u_i^{(n)}}{\sum_{i=0}^{4} \sum_{j=0}^{4} R_c(u_i^{(n)}, v_j^{(n)})} \tag{3.10}$$

where

$$(u_0^{(n)}, u_1^{(n)}, u_2^{(n)}, u_3^{(n)}, u_4^{(n)}) = (u^{(n)} - \frac{u^{(n)}}{2n}, u^{(n)} - \frac{u^{(n)}}{4n}, u^{(n)}, u^{(n)} + \frac{u^{(n)}}{4n}, u^n + \frac{u^{(n)}}{2n}) \tag{3.11}$$

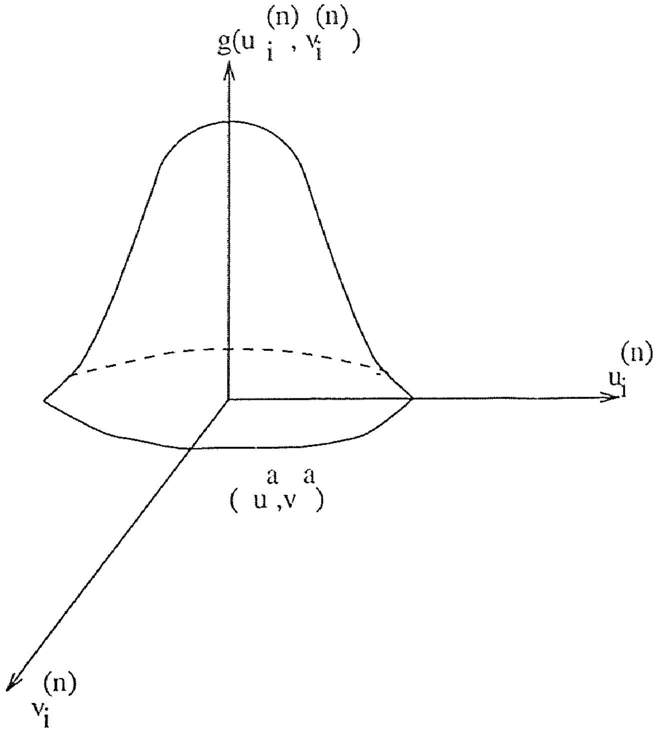


Figure 3.3 Ideal $g(u_i^{(n)}, v_i^{(n)})$.

Let normalized response distribution be denoted by $g(u_i^{(n)}, v_j^{(n)})$,

$$g(u_i^{(n)}, v_j^{(n)}) = \frac{R_c(u_i^{(n)}, v_j^{(n)})}{\sum_{i=0}^{4}\sum_{j=0}^{4} R_c(u_i^{(n)}, v_j^{(n)})} \tag{3.12}$$

Consider the case where $(u_i^{(n)}, v_j^{(n)})$ is in the vicinity of true image vector (u^a, v^a). Assume $g(u_i^{(n)}, v_j^{(n)})$ is a sample of a surface of revolution having only one extreme point (u^a, v^a). The ideal $g(u_i^{(n)}, v_j^{(n)})$ is shown in Figure 3.3. In fact, when $(u_i^{(n)}, v_j^{(n)})$ is the vicinity of (u^a, v^a), the assumption holds approximately.

Without loss of generality, only the positive $u^{(n)}$, u^a, $u^{(n+1)}$ values are considered in order to simplify the proof.

Let

$$h(u_i^{(n)}) = \frac{\sum_{j=0}^{4} R_c(u_i^{(n)}, v_j^{(n)})}{\sum_{i=0}^{4}\sum_{j=0}^{4} R_c(u_i^{(n)}, v_j^{(n)})} \tag{3.13}$$

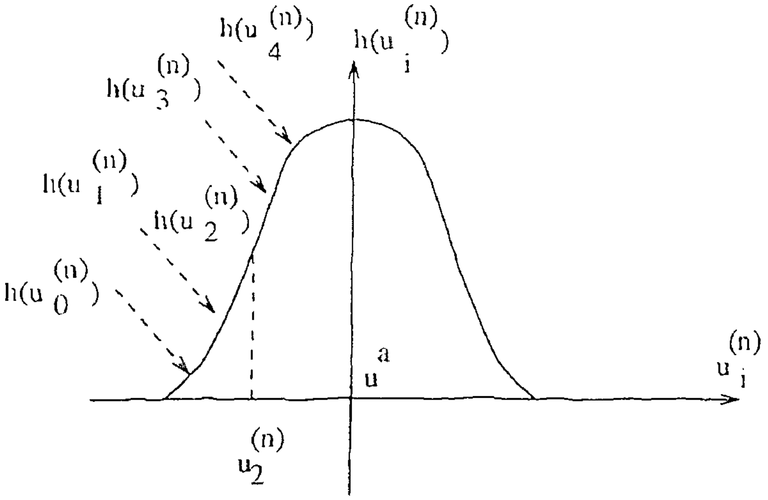


Figure 3.4 Curve of $h(u_i^{(n)})$ and case 1.

It is not difficult to see that $h(u_i^{(n)})$ is a sample of a symmetrical curve about axis $u_i^{(n)} = u^a$ with only one extreme value at $u_i^{(n)} = u^a$. The curve is shown in Figure 3.4.

The following two lemmas will be used for the convergence proof.

Lemma 1: $h(u_0^{(n)}), h(u_1^{(n)}), h(u_2^{(n)}), h(u_3^{(n)})$ and $h(u_4^{(n)})$ satisfy the next equation:

$$\sum_{i=0}^{4} h(u_i^{(n)}) = 1. \tag{3.14}$$

Lemma 2: $u^{(n+1)}$ and $u^{(n)}$ have the following relationships:

1. $u^{(n+1)} = u^{(n)} + u^{(n)}\left(\frac{h(u_4^{(n)}) - h(u_0^{(n)})}{2n} + \frac{h(u_3^{(n)}) - h(u_1^{(n)})}{4n}\right)$ (3.15)
2. $u_3^{(n)} - u^{(n+1)} = u^{(n)}\left(\frac{3h(u_0^{(n)})}{4n} + \frac{h(u_2^{(n)})}{4n} + \frac{2h(u_1^{(n)})}{4n} - \frac{h(u_4^{(n)})}{4n}\right)$ (3.16)
3. $u_4^{(n)} - u^{(n+1)} \geq 0$ (3.17)

Lemma 1 is apparent and its proof is therefore omitted. The proof of Lemma 2 is given below.

From Equations (3.10) and (3.12), the proof of the first part of Lemma 2 is not difficult to follow,

$$\begin{aligned}
u^{(n+1)} &= \sum_{i=0}^{4} u_i^{(n)} h(u_i^{(n)}) \\
&= h(u_0^{(n)})(u^{(n)} - \frac{u^{(n)}}{2n}) + h(u_1^{(n)})(u^{(n)} - \frac{u^{(n)}}{4n}) + h(u_2^{(n)})u^{(n)} \\
&+ h(u_3^{(n)})(u^{(n)} + \frac{u^{(n)}}{4n}) + h(u_4^{(n)})(u^{(n)} + \frac{u^{(n)}}{2n}) \\
&= (h(u_0^{(n)}) + h(u_1^{(n)}) + h(u_2^{(n)}) + h(u_3^{(n)}) + h(u_4^{(n)}))u^{(n)} \\
&+ u^{(n)}(\frac{f(u_4^{(n)}) - h(u_0^{(n)})}{2n} + \frac{h(u_3^{(n)}) - h(u_1^{(n)})}{4n}) \\
&= u^{(n)} + u^{(n)}(\frac{h(u_4^{(n)}) - h(u_0^{(n)})}{2n} + \frac{h(u_3^{(n)}) - h(u_1^{(n)})}{4n})
\end{aligned}$$

Similarly, the second part of Lemma 2 follows from Equations (3.10), (3.11) and (3.13) straightforward.

$$\begin{aligned}
u_3^{(n)} - u^{(n+1)} &= u^{(n)} + \frac{u^{(n)}}{4n} - u^{(n)} - u^{(n)}(\frac{h(u_4^{(n)}) - h(u_0^{(n)})}{2n} + \frac{h(u^{(n)}) - h(u^{(n)})}{4n}) \\
&= \frac{u^{(n)}}{4n} - u^{(n)}(\frac{h(u^{(n)}) - h(u_0^{(n)})}{2n} + \frac{h(u_3^{(n)}) - h(u_1^{(n)})}{4n}) \\
&= \frac{h(u_0^{(n)}) + h(u_1^{(n)}) + h(u_{n2}) + h(u_3^{(n)}) + h(u_4^{(n)})}{4n} u^{(n)} \\
&- (\frac{2h(u_4^{(n)}) - 2h(u_0^{(n)})}{4n} + \frac{h(u_3^{(n)}) - h(u_1^{(n)})}{4n})u^{(n)} \\
&= u^{(n)}(\frac{3h(u_0^{(n)})}{4n} + \frac{h(u_2^{(n)})}{4n} + \frac{2h(u_1^{(n)})}{4n} - \frac{h(u_4^{(n)})}{4n})
\end{aligned}$$

With Equations (3.10), (3.11) and (3.13), and some algebraic manipulation, the third part of Lemma 2 can be proved as follows.

$$\begin{aligned}
u_4^{(n)} - u^{(n+1)} &= u^{(n)} + \frac{u^{(n)}}{2n} - u^{(n)} \\
&- u^{(n)}(\frac{h(u_4^{(n)}) - h(u_0^{(n)})}{2n} + \frac{h(u_3^{(n)}) - h(u_1^{(n)})}{4n}) \\
&= \frac{h(u_0^{(n)}) + h(u_1^{(n)}) + h(u_2^{(n)}) + h(u_3^{(n)}) + h(u_4^{(n)})}{2n} u^{(n)} \\
&- \frac{h(u_4^{(n)}) - h(u_0^{(n)})}{2n} u^{(n)} - \frac{h(u_3^{(n)}) - h(u_1^{(n)})}{4n} u^{(n)} \\
&= u^{(n)}(\frac{h(u_0^{(n)})}{n} + \frac{3h(u_1^{(n)})}{4n} + \frac{h(u_2^{(n)})}{2n} + \frac{h(u_3^{(n)})}{4n})
\end{aligned}$$

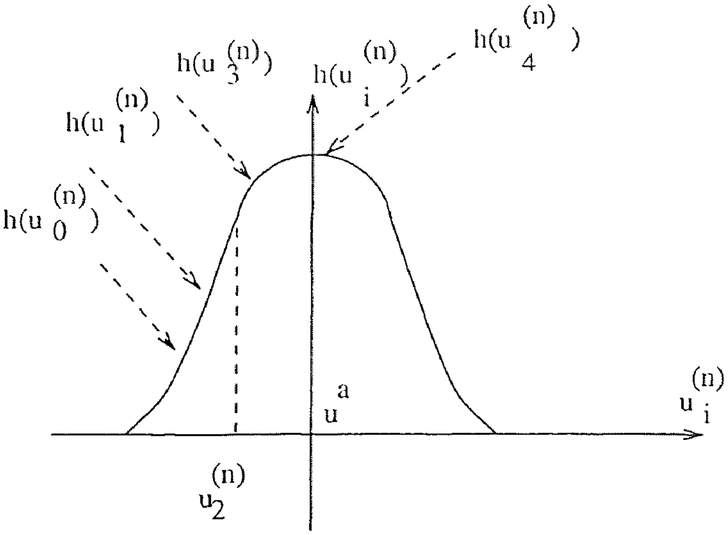


Figure 3.5 Case 2.

$u_4^{(n)} - u^{(n+1)} > 0$, since $h(u_i^{(n)}) > 0 \;\; i = 0, 1, 2, 3, 4$. Now, let's prove the convergence of the algorithm.

Case 1: $u_4^{(n)} \leq u^a$.

Because $h(u_4^{(n)}) > h(u_0^{(n)})$ and $h(u_3^{(n)}) > h(u_1^{(n)})$, and from Equations (3.15) and (3.17), we have

$u_4^{(n)} > u^{(n+1)} > u^{(n)}$.

According to Figure 3.4, we know

$$\left| \frac{u^{(n+1)} - u^a}{u^{(n)} - u^a} \right| \leq 1 \tag{3.18}$$

Case 2: $u_4^{(n)} > u^a$ and $u_3^{(n)} \leq u^a$ as shown in Figure 3.5.

From Equations (3.15), (3.17) and Figure 3.5, one has

$u_4^{(n)} > u^{(n+1)} > u^{(n)}$ and $|u_4^{(n)} - u^a| \leq |u^{(n)} - u^a|$.

Thus

$$|u^{(n+1)} - u^a| \leq |u^{(n)} - u^a|. \tag{3.19}$$

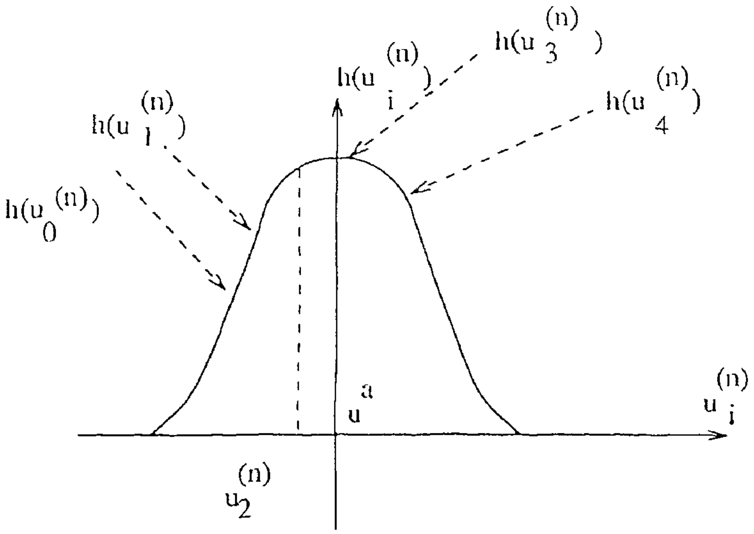


Figure 3.6 Case 3.

Case 3: $u_3^{(n)} > u^a$, $u_2^{(n)} \leq u^a$ and $h(u_4^{(n)}) \leq h(u_2^{(n)}) \leq h(u_3^{(n)})$ as shown in Figure 3.6.

Because $h(u_4^{(n)}) \leq h(u_2^{(n)})$, and $u_3^{(n)} - u^{(n+1)} > 0$ according to Equation (3.16), one has

$u_3^{(n)} > u^{(n+1)} > u^{(n)}$.

From Figure 3.6, we know

$$|u^{(n+1)} - u^a| \leq |u^{(n)} - u^a| \textit{ and } |u^{(n+1)} - u^a| \leq \frac{|u^{(n)}|}{4n}.$$

Case 4: $u_3^{(n)} > u^a$, $u_2^{(n)} \leq u^a$ and $h(u_2^{(n)}) > h(u_3^{(n)}) \geq h(u_4^{(n)})$.

Because $h(u_4^{(n)}) \leq h(u_2^{(n)})$, and $u_3^{(n)} - u^{(n+1)} > 0$ according to Equation (3.16), one has

$u_3^{(n)} > u^{(n+1)} > u^{(n)}$.

As shown in Figure 3.7, $|u^{(n+1)} - u^a| \leq \frac{|u^{(n)}|}{4n}$.

When $n \to \infty$, $|u^{(n)} - u^a| \to 0$.

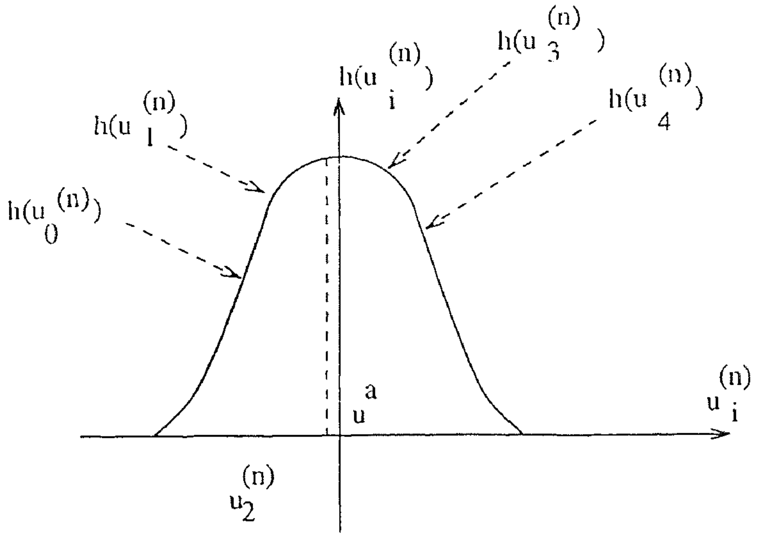


Figure 3.7 Case 4.

Similarly, $|v^{(n)} - v^a|$ eventually tends to zero when $n \to \infty$.

From the above discussion about these four cases and the known symmetry, we conclude the proof of the convergence.

In our experiments, three successive image frames are used to computer R_c so that assumption about the existence of only one extreme point holds much better. In the Section 3.4, the computation of R_c will be described in detail. Our experiments have shown that the algorithm is convergent indeed.

3.3 Analysis

3.3.1 Subpixel

This approach uses feedback technique to iteratively estimate and refine optical flow field. The correlation-based approach can not accurately estimate subpixel optical flow velocity because they don't sufficiently make use of information that is provided by images.

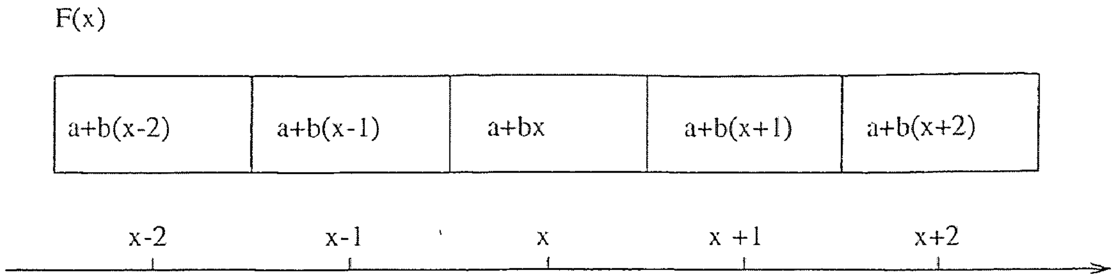


Figure 3.8 Intensities of the first image.

In the correlation-based approaches, the basic procedures are represented as follows. First, a correlation window W_c of size (2m+1) by (2m+1) is formed around the (x, y) pixel in the first image. A search window W_s of size (2N+1) by (2N+1) is established around the pixel at the same location of (x, y) in the second image. The (mis)match measure M(Δx, Δy) between W_c and a window of the same size (2m+1) by (2m+1) around each pixel in W_s, displaced from (x, y) by an amount (Δx, Δy), is computed. According to these measurements, we can compute the image vector (u, v). But, since images are digital, usually we cannot recover true optical flow vectors by the measurements accurately. Therefore, subpixel estimation must be adopted. Are previous subpixel estimations efficient? Let us take a look at the following example.

For the simplicity, a one-dimension case is considered. Intensities of the first image is shown in Figure 3.8. A correlation window W_c of size 3 is formed around the central pixel in the first image. Assume that the intensity is a continuous linear function. That is, $F(x) = a + bx$, where x is the location of pixel. Therefore, when the true image vector is Δx pixel, the second image can be represented as $F(x - \Delta x) = a + b(x - \Delta x)$. When the sum-square-difference is used, we have the error

$$E^2(x_1) \;=\; [F(x-1) - F(x-1-\Delta x + x_1)]^2 + [F(x) - F(x - \Delta x + x_1)]^2$$

$$+ \quad [F(x+1) - F(x+1-\Delta x + x_1)]^2$$
$$= \quad 3b^2(\Delta x - x_1)^2 \tag{3.20}$$

where x_1 is the center pixel in the correlation window in the second image. According to [44], we denote the response distribution by R_{x_1}. We then have

$$R_{x_1} = e^{-E^2(x_1)k} = e^{-3b^2(\Delta x - x_1)^2 k} \tag{3.21}$$

where k is a parameter appropriately chosen. For a discussion of the selection of parameter k, refer to Section 3.4. Here it is chosen as:

$$k = \frac{ln(0.95)}{-3b^2 min^2_{x_1=-2}((\Delta x - x_1)^2)} \tag{3.22}$$

Using the weighted-least-squares, one can estimate the velocity u as follows.

$$u = \frac{\sum_{x_1=-2}^{2} x_1 R_{x_1}}{\sum_{x_1=-2}^{2} R_{x_1}} \tag{3.23}$$

The relative error is then

$$\frac{\Delta x - u}{\Delta x} 100\% \tag{3.24}$$

By using computer simulation, Figures 3.9 and 3.10, the diagrams depicting the relative error and the absolute error versus true displacement, respectively, are obtained. From these figures, we see that when the true disparities are -1, 0 and 1, the associated errors are zero. Usually the relative errors exceed easily beyond much greater than 10%. The Equation (3.23) is a kind of optimal estimation to handle the subpixel problem. Clearly it cannot accurately recover optical flow velocity vectors whose components involve subpixels.

This example indicates that even if image intensity is a linear function and noise free, the correlation-based approach do not work efficiently. This is because they do not sufficiently use information that images provide. They use the information only one time. In fact, if true velocity is not an integer, it is impossible to at once recover the true velocity accurately.

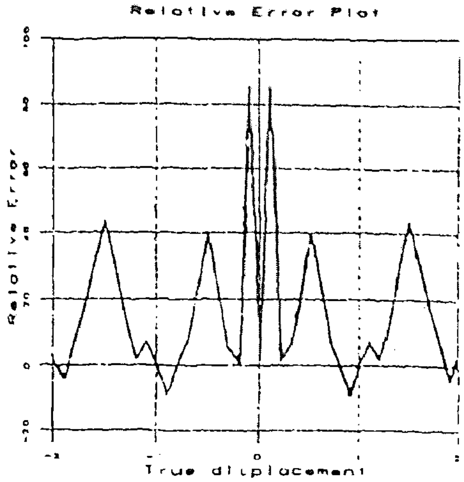


Figure 3.9 Relative error.

Although many refinement processes are used to improve the situation, the effect of the improvement is not always remarkable. Our experiments have shown that when the true image velocity vectors are integer pixels, some correlation-based algorithm can recover some of the vectors accurately, but when the true velocity vectors are subpixels, error is increased.

In correlation-feedback approach, "rubber" window is used, i.e., SSD search is based on the variable size instead of fixed grid. Therefore, it is not necessary that all of true image velocities are integer pixels. Feedback iteration is utilized to adjust the optical flow field so that the field becomes more and more accurate.

3.3.2 Refinement

Most of previous refinement methods are based on some sort of optimal objective criteria. Any criterion may have limitation and weakness. Most of these refinements make use of the neighborhood optical flow velocity vectors to improve the velocity vector. However, if the neighborhood velocity vectors are not recovered very well, using them is not helpful and not reliable. Also, use of the neighborhood image velocities violate the local discontinuity of optical flow field. Under the circum-

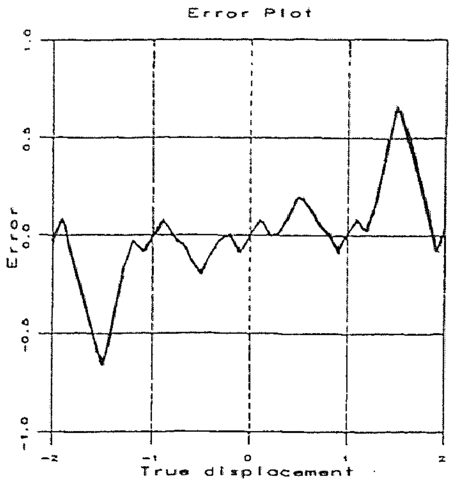


Figure 3.10 Absolute error.

stances that only image sequence is known, it is best and most reliable to make use of matching relationship between the images to refine optical flow. Our optimal objective criterion is just based on the idea. The criterion is to make the difference between the second given image $I_2(i,j)$ and the image $f(i_2-u, j_2-v)$ estimated from the first given image $I_1(i,j)$ minimal. Our experiments verified that the optimal objective criterion make not only the effect of noise reduced but also the boundary information conserved.

3.3.3 Convergence

In Section 3.2, it is proved that the correlation-feedback algorithm is convergent, if the normalized response distribution

$$g(u_i^{(n)}, v_j^{(n)}) = \frac{R_c(u_i^{(n)}, v_j^{(n)})}{\sum_{i=0}^{4}\sum_{j=0}^{4} R_c(u_i^{(n)}, v_j^{(n)})} \tag{3.25}$$

is symmetrical and has only one peak which is assumed by true optical flow velocity.

When image intensity is a linear function of coordinates, it can be shown that this condition must be satisfied. Without loss of generality, a one dimension case is considered. From Equation (3.21), we have

$$\begin{aligned} g(u_i^{(n)}) &= \frac{R_c(u_i^{(n)})}{\sum_{u_i^{(n)}} R_c(u_i^{(u)})}, \\ &= \frac{e^{-3b^2(\Delta x - u_i^{(n)})^2 k}}{\sum_{u_i^{(n)}} e^{-3b^2(\Delta x - u_i^{(n)})^2 k}} \end{aligned} \tag{3.26}$$

It is not difficult to see from the above equation that only one peak of the response distribution exists and the peak value is achieved when $u^{(n)}$ is equal to Δx.

In fact, in the vicinity of a pixel, the intensity can usually be considered as such a linear intensity, especially, bilinear interpolation is used. Therefore, we can claim that mostly the true optical flow vectors having subpixel or integer pixel cômponents can be recovered by using the correlation-feedback approach.

3.4 Implementation and experiments

3.4.1 Implementation

Three images: the first, second and third frames in an image sequence, denoted by I_1, I_2, and I_3, respectively, are used to compute optical flow. Each is of 256 × 256. Assume the time interval between I_1 and I_2 is the same as that between I_2 and I_3. Also assume approximate uniform motion during these two intervals. From images I_1 and I_2, $(u^{(o)}(i,j), v^{(o)}(i,j))$ can be computed using the Horn and Schunck's algorithm [9]. From $u^{(n)}(i,j)$, $v^{(n)}(i,j)$, the optical flow estimated during the nth iteration, and I_1 and I_2, the response distribution $R_c^+(u^{(n)}, v^{(n)})$ can be calculated using Equations (3.5) and (3.7). Similarly, from images I_3 and I_2, $-u^{(n)}(i,j)$, $-v^{(n)}(i,j)$, $R_c^-(-u^{(n)}, -v^{(n)})$ can be calculated. The response distribution $R_c(u^{(n)}, v^{(n)})$ can then be determined as follows.

$$R_c(u^{(n)}, v^{(n)}) = R_c^+(u^{(n)}, v^{(n)}) + R_c^-(-u^{(n)}, -v^{(n)}) \tag{3.27}$$

$u_c^{(n)}(i,j)$ and $v_c^{(n)}(i,j)$ can be obtained by using Equation (3.8). We adopt 3 × 3 correlation windows. The Gaussian mask shown in Figure 3.2 is used. The choice of

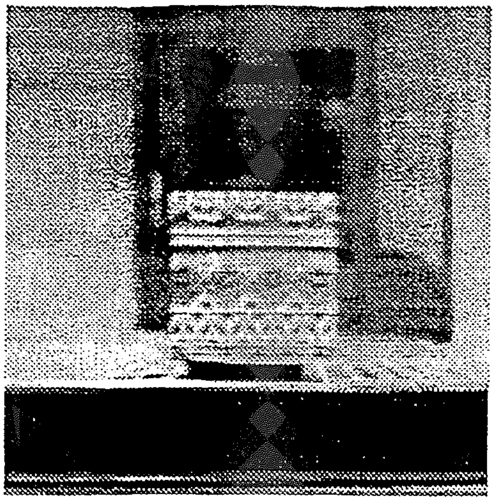


Figure 3.11 Texture square (a).

parameter k in Equation (3.7)is adaptive [44]. In each search window, k is chosen so as to make R_c^+ and R_c^- be a number close to unity. In the observer stage, the bilinear interpolation is used. Our experiments show that the bilinear interpolation is faster and better than $B - spline$. The u and v used in the implementation are a little different from that in Equation (3.6), and are shown below.

$$u \in \{\frac{u^{(n)}}{2}, \frac{3u^{(n)}}{4}, u^{(n)}, \frac{5u^{(n)}}{4}, \frac{3u^{(n)}}{2}\}$$
$$v \in \{\frac{v^{(n)}}{2}, \frac{3v^{(n)}}{4}, v^{(n)}, \frac{5v^{(n)}}{4}, \frac{3v^{(n)}}{2}\}$$

Our experiment proves that this kind of rubber window has better convergence and higher accuracy than the rubber window described in Equation (3.6).

3.4.2 Experiment I

Figures 3.11, 3.12 and 3.13 show the three successive image frames: I_1, I_2 and I_3. They were taken by a CCD video camera and a DATACUBE system that is a real time image processing system and is supported by a Sun workstation. The experiment setting is shown in Figure 3.14. The focal length of the camera is 12.5 mm. The distance between the texture square post and the camera is 1300 mm. The

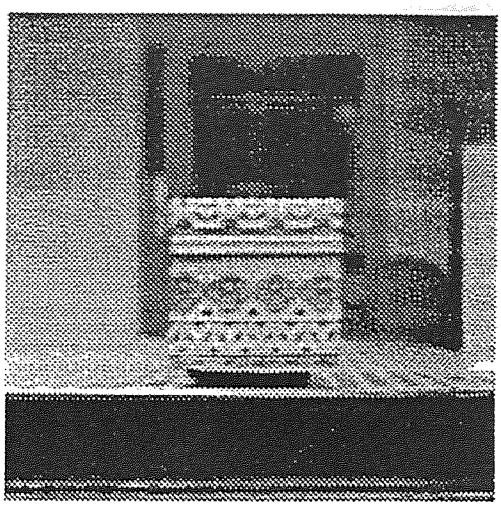


Figure 3.12 Texture square (b).

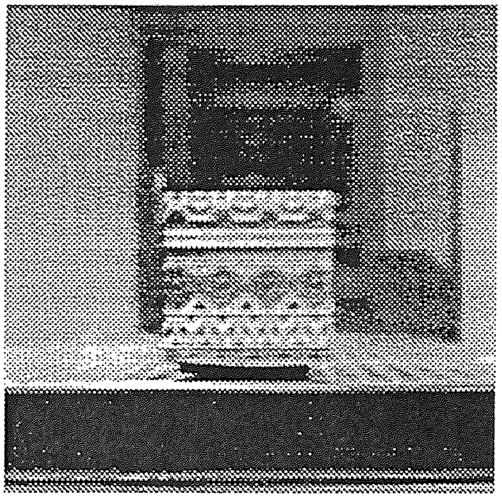


Figure 3.13 Texture square (c).

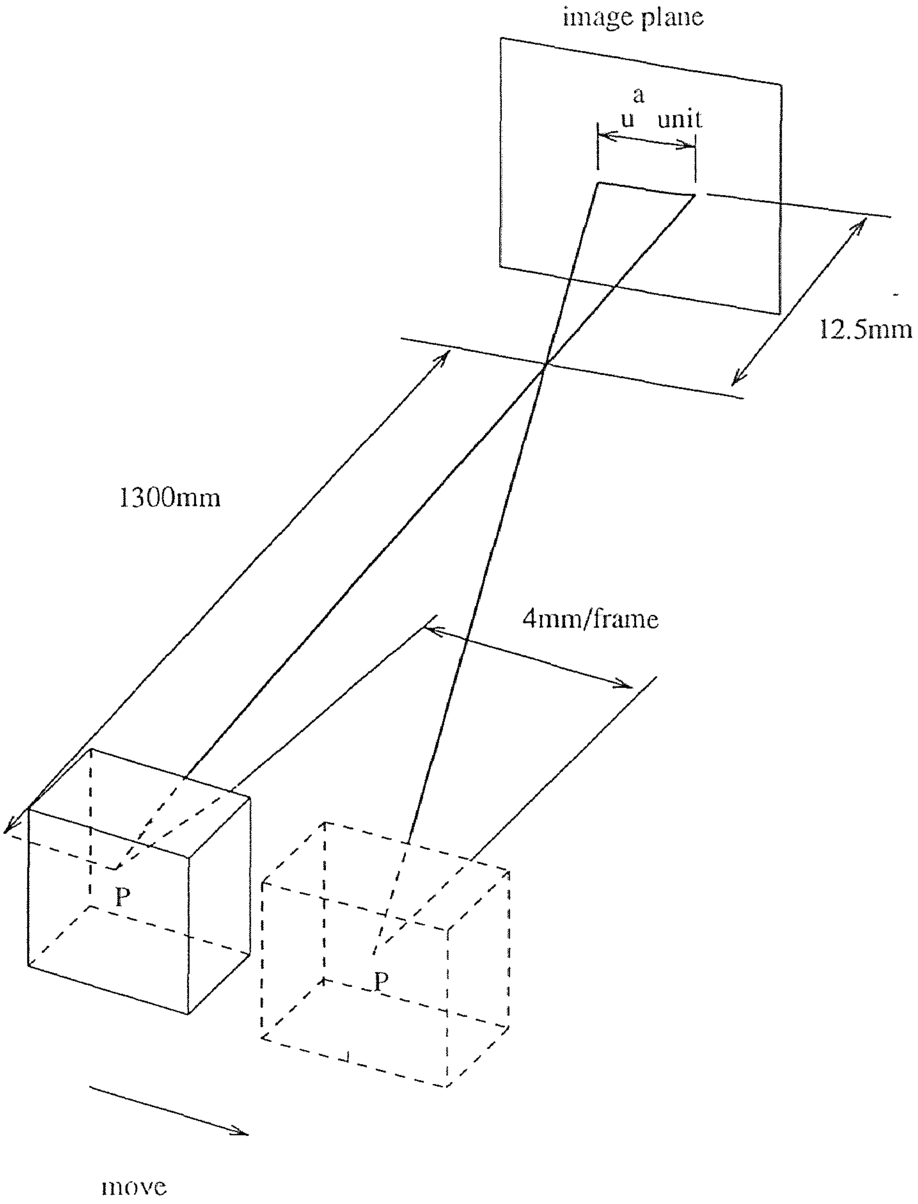


Figure 3.14 Imaging geometry in Experiment I.

texture square moves normal to the line of sight, with a velocity of 4 mm/frame. According to the perspective projection relationship shown in Figure 3.14, the true optical-flow (u^a, v^a) is as follows.

$$u^a = -\frac{12.5 \times 4}{1300 \times unit} \tag{3.28}$$

$$v^a = 0. \tag{3.29}$$

where the *unit* is a constant that represents the distance between two consecutive pixels along x direction in a digital image. From camera calibration, we know $unit =$ 0.014 mm/ pixel. It will be discussed in Section 4.6 in detail. Thus

$$u^a = -2.747 \ (pixels). \tag{3.30}$$

The true optical flow field is shown in Figure 3.15. Since in our setting the object is in motion while the camera is static, and the light of our laboratory is not uniform, the different environmental light produces non-negligible noise. Before computing the optical flow field, these three 256 × 256 images are compressed into three 64 × 64 images by sub-sampling. The sub-sampling algorithm is extremely simple. That is, a 256 × 256 image is uniformly segmented into 64 × 64 blocks. Each block having 4 × 4 pixels corresponds to a pixel in the 64 × 64 image. The average intensity of these 4 × 4 pixels is used as the intensity of the corresponding pixel in the 64 × 64 image, thus, generating three 64 × 64 images. Because of the sub-sampling, the true optical flow values become: $u^a = -0.6868$; $v^a = 0$.

In order to compare the performance of correlation-feedback approach with that of the gradient-based approach and correlation-based approach. We choose the Horn and Schunck's [9] algorithm to represent the gradient-based approach and Singh's framework [44] the correlation-based approach. Table 3.1 shows the results of comparison. In Table 3.1, $n = 2$ means that the correlation window is 5×5; $w = 2$ means that the Gaussian mask size is 5×5; $N = 4$ means that the search window is 9×9. The program that implements Singh's algorithm is provided by Barron [40].

In the correlation-feedback algorithm, Horn and Schunck's algorithm with $\alpha = 5$ is used in initialization. Finally, central 40×40 optical vector array is used to compute u_{error} as follows.

$$u_{error} = \frac{\sqrt{\sum_{i=0}^{39} \sum_{j=0}^{39} ((u^a(i,j) - u^r(i,j))^2 + (v^a(i,j) - v^r(i,j))^2)}}{\sqrt{\sum_{i=0}^{39} \sum_{j=0}^{39} (u^a(i,j)^2 + v^a(i,j)^2)}} \tag{3.31}$$

where (u^a, v^a) is a true optical flow vector and (u^r, v^r) is a calculated optical flow vector. These apply to all of the experiments reported in this chapter. The optical flow fields that are calculated by using the gradient-based algorithm, the correlation-based algorithm and the correlation-feedback algorithm are shown in Figures 3.16, 3.17 and 3.18 respectively.

From Figures 3.11, 3.12 and 3.13, we know that many true optical flow vectors are zeros because of the static background. It is why when Equation (3.31) is used to calculate relative error, the denominator is too small due to the fact: many true optical flow vectors are zeros. Consequently, the relative error calculated in Experiment I is greater than 10%. It is not easy to choose an appropriate formula to compute relative error especially when many true optical flow vectors are zeros. Relatively speaking, our correlation-feedback algorithm performs best in determining optical flow for a texture post in translation.

In this experiment the correlation-feedback algorithm does converge. Its convergence process is demonstrated in Figure 3.19.

3.4.3 Experiment II

Figures 3.20, 3.21 and 3.22 are obtained by rotating a CCD camera with respect to the center of a ball. The imaging geometry is shown at Figure 3.23. The rotating velocity is $2.5^o/frame$. The focal length of the camera is 12.5 mm. The distance between the center of the ball and the camera is 1120 mm. In Figures 3.20, 3.21 and 3.22, the background seems static because it moves together with the camera. Before computing optical flow field, three 256×256 images are compressed into three

Table 3.1 The comparison in Experiment I

Techniques	Gradient-based approach	Correlation-based approach	Correlation-feedback approach
Conditions	$Iteration = 128$ $\alpha = 5$	$Iteration = 25$ $n = 2, w = 2$ $N = 4$	$Iteration = 10$ $Horn - iteration = 10$ $n = 1, w = 1$
u_{error}	56.37%	80.97%	44.56%

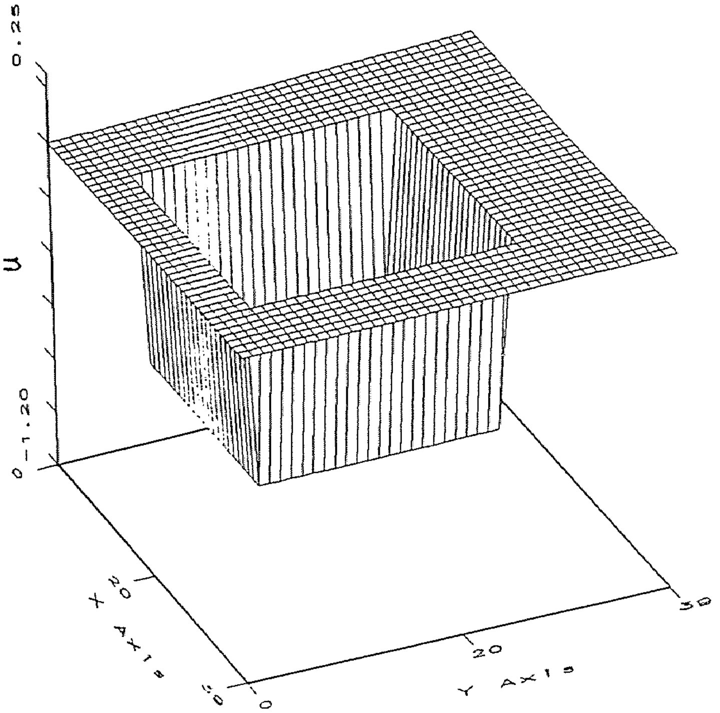


Figure 3.15 $u^a(x, y)$ of correct optical flow field.

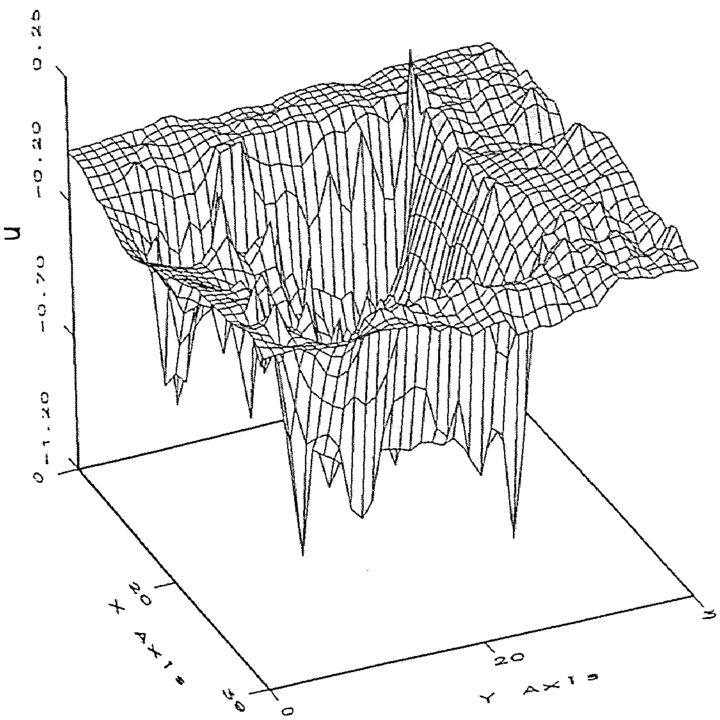


Figure 3.16 $u^r(x,y)$ computed using the gradient-based approach.

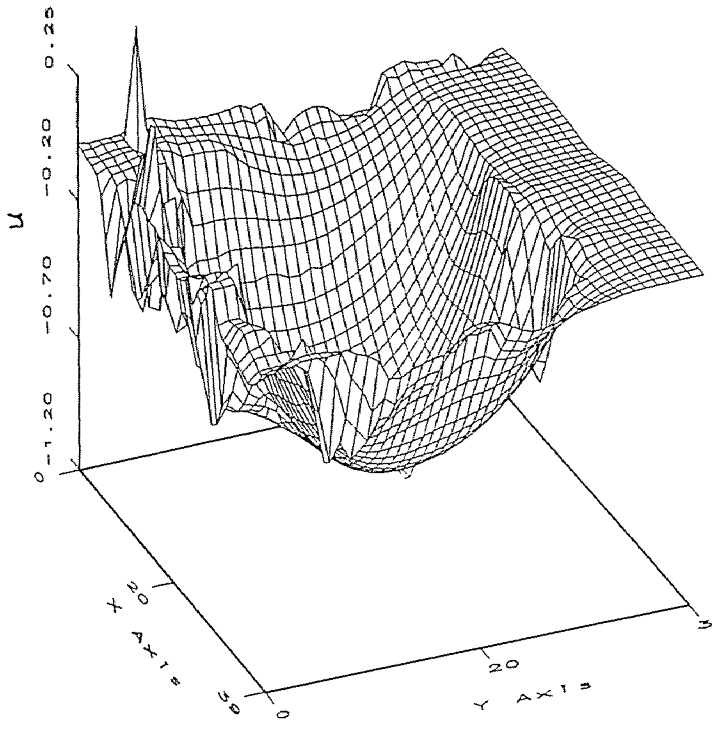


Figure 3.17 $u^r(x,y)$ computed using the correlation-based approach.

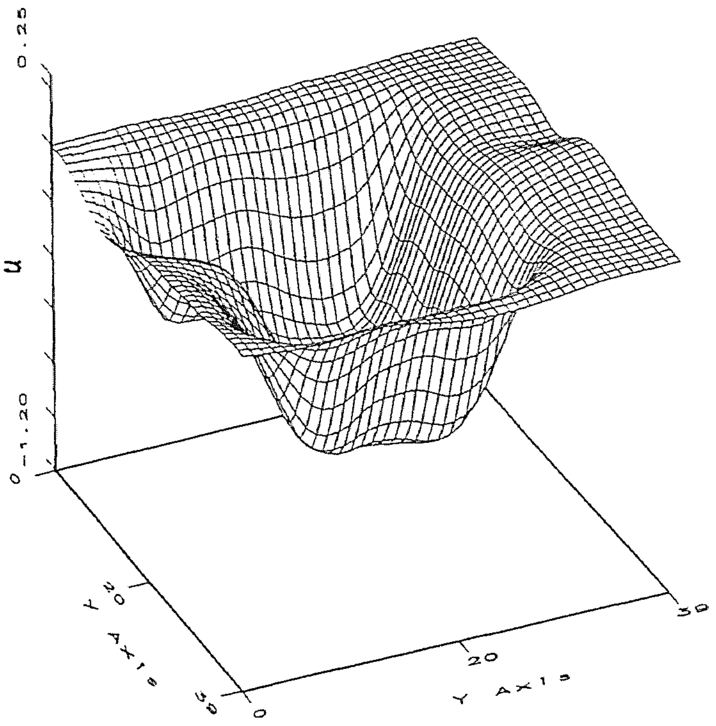


Figure 3.18 $u^r(x, y)$ computed using the correlation-feedback approach.

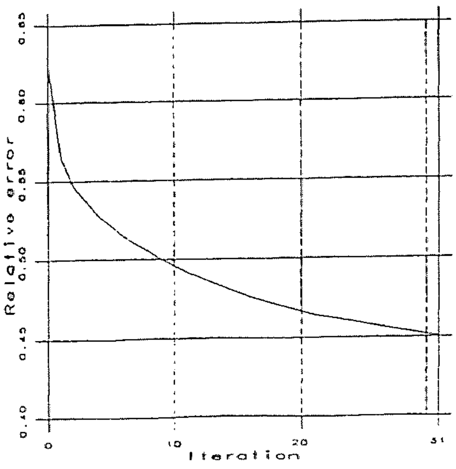


Figure 3.19 The convergence process of the correlation-feedback approach.

Table 3.2 The comparison in Experiment II

Techniques	Gradient-based approach	Correlation-based approach	Correlation-feedback approach
Conditions	$Iteration = 128$ $\alpha = 5$	$Iteration = 25$ $n = 2, w = 2$ $N = 4$	$Iteration = 10$ $Horn - iteration = 10$ $n = 1, w = 1$
u_{error}	65.67%	55.29%	49.80%

64×64 images by using the same sub-sampling technique as that used in Experiment I. Finally, central 40×40 optical vector arrays are used to compute u_{error}. Table 3.2 reports the results for this experiment. There u_{error}, (u^a, v^a), (u^r, v^r), n, w, and N are of the same meaning as defined in Experiment I. It is obvious that our correlation-feedback algorithm performs best in determining optical flow for this rotating ball case.

3.4.4 Experiment III

As mentioned before, a very comprehensive study of various optical flow techniques and comparison of their performance mainly in terms of accuracy have been conducted recently in [40]. In order to compare our correlation-feedback algorithm with other techniques in a more objective, quantitative manner, we cite some results reported in [40] which were obtained by applying some typical optical flow techniques to some image sequences chosen with deliberation. On the other hand, we apply our feedback technique to the identical image sequences and then report the results with the same criterion as used in [40].

Three kinds of image sequence used in [40], kindly provided by the authors of [40], are utilized here. They are named Translation Tree 2-D, Diverging Tree 2-D, and Yosemite. The first two simulate translational camera motion with respect to a textured planar surface (see Figure 3.24). In the Translating Tree 2-D sequence, the

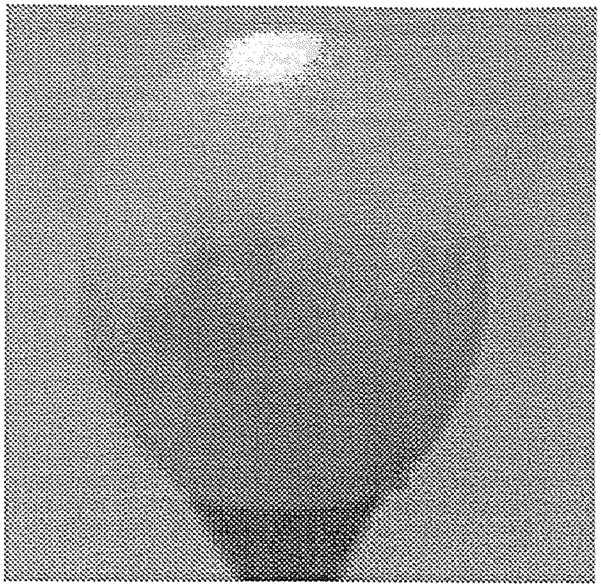


Figure 3.20 Ball (a).

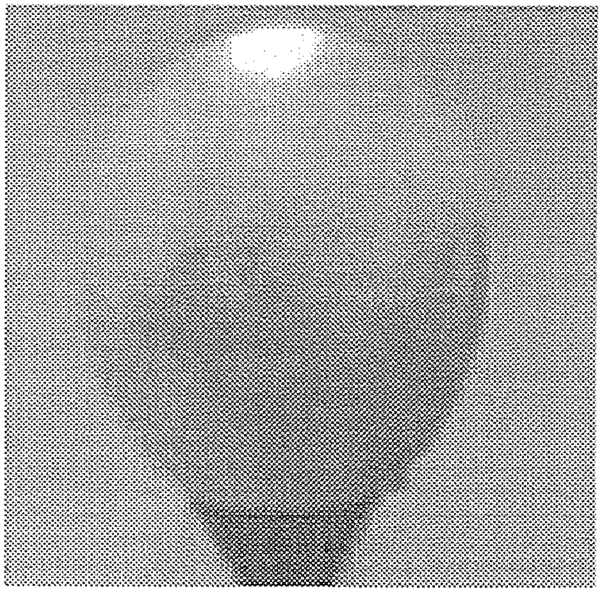


Figure 3.21 Ball (b).

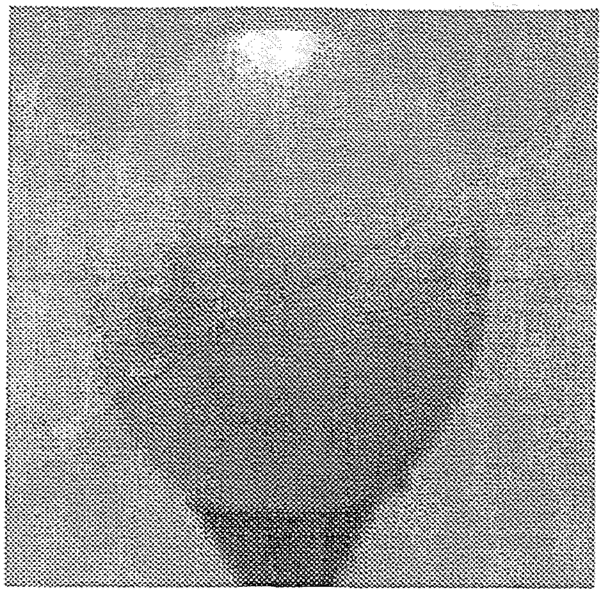


Figure 3.22 Ball (c).

camera moves normal to its line of sight along its X-axis, with velocities between 1.73 and 2.26 pixels/frame parallel to the x-axis in the image plane. In the Diverging Tree 2-D sequence, the camera moves along its line of sight. The focus of expansion is at the center of the image. The speeds vary from 1.29 pixels/frame on left side to 1.86 pixels/frame on the right. The Yosemite sequence is a more complex test case (see Figure 3.25). The motion in the upper right is mainly divergent, the clouds translate to the right with a speed of 1 pixel/frame, while velocities in the lower left are about 4 pixels/frame. Barron, Fleet Beauchemin pointed out in [40] that "this sequence is challenging because of the range of velocities and the occluding edges between the mountains and at the horizon. There is severe aliasing in the lower portion of the images however, causing most methods to produce poorer velocity measurements."

The same way to express the error occurred in the optical flow determination as that used in [40], i.e., the angular measure of the error is utilized here. As Barron, Fleet and Beauchemin pointed out in [40] that optical flow vectors may be written

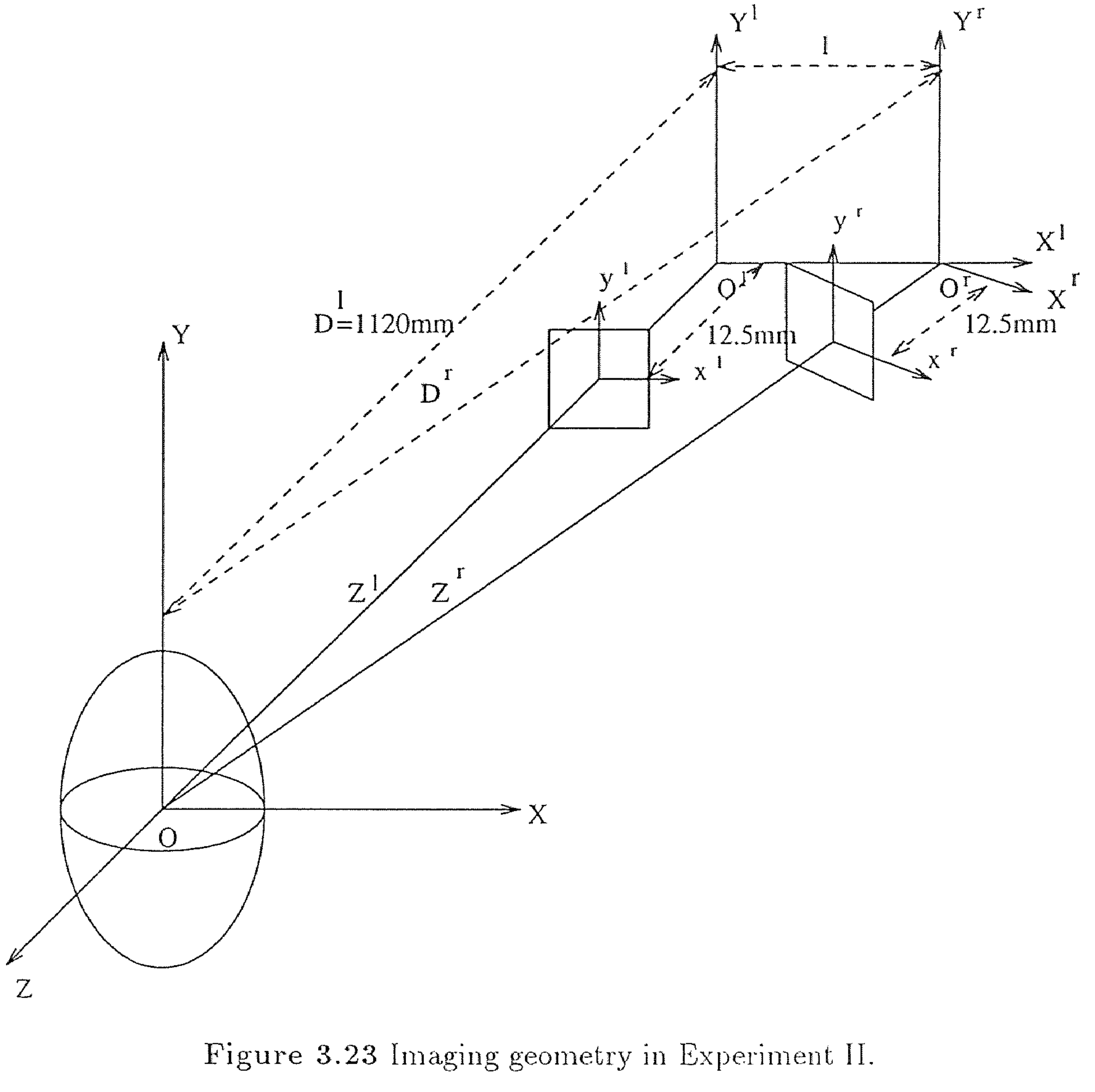


Figure 3.23 Imaging geometry in Experiment II.

as displacements per time unit as in $\vec{U} = (u, v)$ pixels/frame, or as a space-time direction vector $(u, v, 1)$ in units of (pixel, pixel, frame). When image velocity is viewed (and measured) as orientation in space-time, it is natural to measure errors as angular deviations from the correct space-time orientation. Therefore, let image velocity $\vec{U} = (u, v)^T$ be represented as 3-D direction vectors, $\vec{V} \equiv \frac{1}{\sqrt{u^2+v^2+1}}(u, v, 1)^T$. The angular error between the correct image velocity $\vec{V_c}$ and an estimate $\vec{V_e}$ is

$$\psi_E = arccos(\vec{V_c} \cdot \vec{V_e}). \tag{3.32}$$

It is obvious that the smaller the angular error ψ_E is the more accurate the estimation of optical flow field will be. Despite of the fact that the confidence measurement can be used in our correlation-feedback algorithm, in this chapter, we don't consider the usage of the confidence measurement. Therefore only the results that have 100% density in Table 4.6, Table 4.7 and Table 4.10 in [40] are used in Tables 3.3 - 3.5 in this chapter.

In the experiments, our correlation-feedback algorithm is applied to the above-mentioned three kinds of image sequences. The results are then compared with that obtained by using other techniques, reported in [40] in terms of angular error. Therefore, we consider this comparison is objective and reliable.

Prior to computation of optical flow field, Yosemite images are compressed from 316 × 252 to 79 × 63, Translating Tree 2-D and Diverging Tree 2-D images are compressed from 150 × 150 to 75 × 75 by using the same sub-sampling technique as that used in Experiment I. In fact, this type of compression is of of function of low-pass filtering.

As mentioned in [40] the optical flow field for Yosemite image sequence is very complex and Table 3.5 indicates evidently that our correlation-feedback algorithm performs best. These lead to that our feedback technique is very suitable for complex optical flow field determination. This is because the powerful feedback technique is applied repeatedly and its refinement is based on the optimal local matching

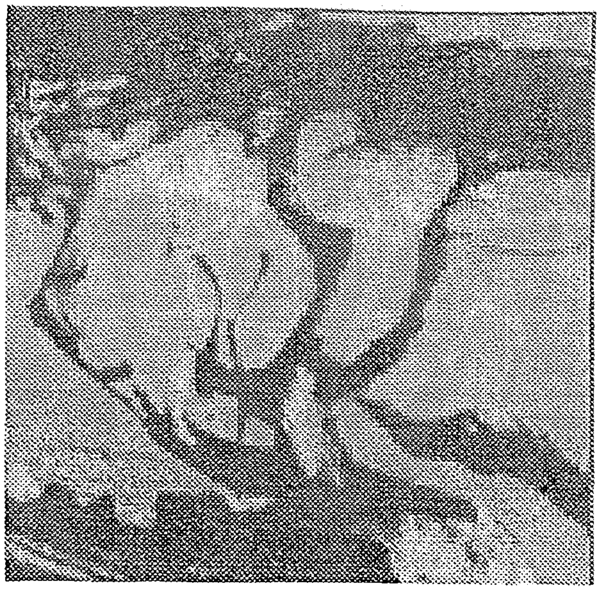


Figure 3.24 TREE.

between the interpolated image, generated with estimated optical flow, and the given digital images instead of other types of optimization. Tables 3.3 and 3.4 indicate the feedback technique also performs very well in the case of translating and diverging texture post cases.

3.5 Summary

It is well-known that feedback is a powerful technique widely used in the field of automatic control field. It can make systems robust against noise and improve performance drastically. In this chapter, we propose for the first time in the literature on optical flow techniques a feedback approach to determining optical flow. The numerous experiments in our laboratory demonstrate that it performs very well in general. Usually better than the gradient-based and correlation-based approaches. In the very complicated optical flow cases, specifically in the case of the Yosemite

Figure 3.25 YOSEMITE.

Table 3.3 Summary of the TRANSLATING TREE 2D velocity results.

Techniques	Average Error	Standard Deviation	Density
Horn and Schunck (original)	38.72°	27.67°	100%
Horn and Schunck (modified)	2.02°	2.27°	100%
Uras et al. (unthresholded)	0.62°	0.52°	100%
Anandan	4.54°	3.10°	100%
Singh (step 1, n=2, w=2)	1.64°	2.44°	100%
Singh (step 2, n=2, w=2)	1.25°	3.29°	100%
Correlation feedback (n=1, w=1)	1.07°	0.48°	100%

Table 3.4 Summary of the DIVERGING TREE 2D velocity results.

Techniques	Average Error	Standard Deviation	Density
Horn and Schunck (original)	12.02°	11.72°	100%
Horn and Schunck (modified)	2.55°	3.67°	100%
Uras et al. (unthresholded)	0.62°	0.52°	100%
Nagel	2.94°	3.23°	100%
Anandan (frames 19 and 21)	7.64°	4.96°	100%
Singh (step 1, n=2, w=2)	17.66°	14.25°	100%
Singh (step 2, n=2, w=2)	8.60°	5.60°	100%
Correlation feedback (n=1, w=1)	5.12°	2.16°	100%

Table 3.5 Summary of the YOSEMITE 2D velocity results.

Techniques	Average Error	Standard Deviation	Density
Horn and Schunck (original)	32.43°	30.28°	100%
Horn and Schunck (modified)	11.26°	16.41°	100%
Uras et al. (unthresholded)	10.44°	15.00°	100%
Nagel	11.71°	10.59°	100%
Anandan (frames 19 and 21)	15.84°	13.46°	100%
Singh (step 1, n=2, w=2)	18.24°	17.02°	100%
Singh (step 2, n=2, w=2)	13.16°	12.07°	100%
Correlation feedback (n=1, w=1)	7.93°	6.72°	100%

image sequence, considered as a most challenging one in a recent comprehensive study of optical flow techniques [40], it performs better than all other techniques.

A concrete implementation of our feedback approach is the correlation-feedback algorithm. It is an iterative procedure. In each iteration, the estimated optical flow vectors $(u^{(n)}, v^{(n)})$ and their vicinity values (25 combinations in our algorithm) are fed back to the observer stage. There they together with the first given digital image are used to estimate, via a bilinear interpolation, certain portion (25 windows in our implementation) of a virtual continuous image which is supposed to be coincident with the second given digital image after a translation by the true optical flow. (It is worth noting that there is no need to estimate the whole continuous image field in the procedure. Only those values needed are estimated by using the bilinear interpolation). In the correlation stage of the correlation-feedback approach, the difference between the estimated portion of the continuous image and the corresponding portion in the second given digital image is calculated in terms of sum-square-difference (SSD). A weighted-least-square technique is then used to calculate $(u_c^{(n)}, v_c^{(n)})$, the output of the correlation stage at the new iteration. In the propagation stage, the optical flow vectors of pixels at a neighborhood make contribution to the optical flow vector of the central pixel via the use of a Gaussian mask.

Our correlation stage is quite different from that used in the correlation-based approaches in the following four aspects. Firstly, the search window in our approach is a "rubber" window in the sense that it has variable size depending on $(u^{(n)}, v^{(n)})$. In the correlation-based approaches, the search window has fixed size. Secondly, the weighted-least-squares technique is used to calculate the output of the correlation stage from 25 different weighted combination of $(u^{(n)}, v^{(n)})$ instead of from all possible pixel positions in the fix-sized search window in the second digital image as in the correlation-based approaches. Thirdly, a bilinear interpolation technique used in the observation stage provides the correlation stage with a virtual continuous image field for a more accurate SSD computation. Fourthly, the refinement objective function is based on the minimal difference between the estimated continuous image and the second given digital image. Therefore, this correlation-feedback algorithm can reduce error considerably.

Since the observation, correlation and propagation stages are used in each iteration, in other words, since the special objective function (the best matching between interpolated continuous image by using estimated optical flow and the given second digital image) is adopted, not only effect of noise is reduced but also the boundary information is conserved. From Figures 3.16 -3.18, it is evident that our feedback approach not only make optical flow field smoother than the gradient-based and correlation-based approaches do but also preserve flow discontinuity better than the other two techniques do.

When the response distribution has single maximum which is assumed by the true optical flow, the convergence of our algorithm to the true optical flow has been proven. This condition is satisfied in the most practical cases. Specifically, consider the ideal case where image intensity is a linear function of the image coordinates and without noise. It is shown in Section 3.3.1 that, except the optical flow has only integer multiple of pixels as its components, the correlation-based approach will easily have error much greater than 10%. On contrary, this chapter shows that the feedback technique proposed can converge to the true optical flow in this case.

As discussed, the correlation-feedback algorithm is quite robust against the choice of the initialization. Even arbitrary non-zero constant initialization works.

CHAPTER 4

COMPUTING 3-D MOTION

Once optical flow fields are known, one can compute 3-D motion and structure. As noted in Chapter 2, previous works mostly are based on motion of camera. However, the motion of camera can be represented by six constants. Assuming that the six constants are known, depth map can be computed from optical flow field. When objects are allowed to move, depth map and 3-D motion are all variables with respect to space coordinates, those approaches can not be used.

In this chapter, the Unified Optical Flow Field (UOFF) is discussed [41] - [43]. Based on the approach, the objects position $(X(x,y),\ Y(x,y),\ Z(x,y))$ and velocity $(\dot{X}(x,y), \dot{Y}(x,y), \dot{Z}(x,y))$ can be computed, where x and y are coordinates of image plane. The approach does not require any condition such as objects should be rigid, motion or depth map should be known.

4.1 Discussion of Unified Optical Flow Field (UOFF)

According to UOFF, an image brightness function should be described as below :

$$g(x, y, t, \vec{s})$$

where x and y are coordinates on image plane, t is time, $\vec{s}$ indicates the sensor's i.e. camera position in 3-D world space, i.e., the coordinates of the sensor center and the orientation of the optical axis of the sensor, that is, $\vec{s}$ is a 5 - D vector.

$$\vec{s} = (\tilde{x}, \tilde{y}, \tilde{z}, \beta, \gamma) \tag{4.1}$$

where $\tilde{x}$, $\tilde{y}$ and $\tilde{z}$ represent the coordinate of the optical center of the sensor in 3 -D world space; β and γ represent the orientation of the optical axis of the sensor in 3-D world space.

Assume now a world point P in 3-D space that is projected onto the image plane as a pixel with the coordinates x_P and y_P. Then, x_P and y_P are also dependent on t and $\vec{s}$. That is, the coordinates of the pixel can be denoted by $x_P = x_P(t, \vec{s})$ and $y_P = y_P(t, \vec{s})$. Therefore, generally speaking, we have

$$g = g(x_P(t, \vec{s}), y_P(t, \vec{s}), t, \vec{s}) \tag{4.2}$$

Due to the assumption of the time-and-space-invariance of brightness, one can get

$$g(x(t, \vec{s}), y(t, \vec{s}), t, \vec{s}) = g(x(t + \Delta t, \vec{s} + \Delta\vec{s}), y(t + \Delta t, \vec{s} + \Delta\vec{s}), t + \Delta t, \vec{s} + \Delta\vec{s}) \tag{4.3}$$

The expansion of the right-hand side of the above equation in the Taylor series and the use of Equation (4.3) lead to

$$(\frac{\partial g}{\partial x}u + \frac{\partial g}{\partial y}v + \frac{\partial g}{\partial t})\Delta t + (\frac{\partial g}{\partial x}\vec{u^s} + \frac{\partial g}{\partial y}\vec{v^s} + \frac{\partial g}{\partial \vec{s}})\Delta\vec{s} + \epsilon = 0 \tag{4.4}$$

where $u \triangleq \frac{\partial x}{\partial t}$, $v \triangleq \frac{\partial y}{\partial t}$, $\vec{u^s} \triangleq \frac{\partial x}{\partial \vec{s}}$, $\vec{v^s} \triangleq \frac{\partial y}{\partial \vec{s}}$.

We now discuss some special cases about the brightness time-and-space invariant equation, i.e., Equation (4.4):

Case 1: If $\Delta\vec{s} = 0$, i.e., sensor is static in a fixed spatial position, in other words both the coordinate of the optical center of sensor and its optical axis direction remain unchanged, dividing both sides of the equation by Δt and evaluating the limit as $\Delta t \to 0$ degenerate Equation (4.4) into:

$$\frac{\partial g}{\partial x}u + \frac{\partial g}{\partial y}v + \frac{\partial g}{\partial t} = 0 \tag{4.5}$$

It is the result derived by Horn and Schunck [9].

Case 2: If $\Delta t = 0$, its both sides are divided by $\Delta\vec{s}$ and $\Delta\vec{s} \to 0$ is examined, Equation (4.4) then reduces to:

$$\frac{\partial g}{\partial x}\vec{u^s} + \frac{\partial g}{\partial y}\vec{v^s} + \frac{\partial g}{\partial \vec{s}} = 0 \tag{4.6}$$

When $\Delta t = 0$, i.e., at a specific time moment, the images generated with sensors at different spatial positions can be viewed as a spatial sequence of images. Equation(4.6) is then the equation for the spatial sequence of images. A simplified version of this equation is discussed in the next section.

Case 3: If $\frac{\Delta \vec{s}}{\Delta t}$ is a constant, Equation (4.4) is the equation for a sequence of images taken by a sensor experiencing a uniform motion.

Case 4: If $\frac{\Delta \vec{s}}{\Delta t} = f(\vec{s}, t)$ and the function $f(\vec{s}, t)$ is given, Equation (4.4) can then be utilized to treat the case when the sensor is experiencing a known movement.

4.2 Further Discussion of UOFF

In stereo imagery, two sensors are usually fixed in two positions in 3-D world space. Hence, we can simplify discussion by considering two points in $t - \vec{s}$ space, i.e., $(t_1, \vec{s}_1)$ and $(t_2, \vec{s}_2)$ and choosing a straight line in $t - \vec{s}$ space which passes through points $(t_1, \vec{s}_1)$ and $(t_2, \vec{s}_2)$. Along this line, we use ρ as a parameter to indicate the distance between another point in the line and the point $(t_1, \vec{s}_1)$. Different ρ is therefore corresponding to different point along the line in $t - \vec{s}$ space. Along this line we then have

$$g = g(x(\rho), y(\rho), \rho) \tag{4.7}$$

Thus, Equation (4.3), i.e., the expression for the brightness time-and-space-invariance equation, becomes

$$g(x(\rho), y(\rho), \rho) = g(x(\rho + \Delta\rho), y(\rho + \Delta\rho), \rho + \Delta\rho) \tag{4.8}$$

It is noted that the variation of ρ implies the variation of both time and space along the line. Similar derivation to that presented in Section 4.1 can lead to

$$g_x \mu + g_y \nu + g_\rho = 0 \tag{4.9}$$

where

$$\mu \triangleq \frac{\partial x}{\partial \rho} \tag{4.10}$$

$$\nu \triangleq \frac{\partial y}{\partial \rho} \tag{4.11}$$

$$g_x \triangleq \frac{\partial g}{\partial x} \quad g_y \triangleq \frac{\partial g}{\partial y} \quad g_\rho \triangleq \frac{\partial g}{\partial \rho} \tag{4.12}$$

Here, μ and ν are, respectively, the variation rates of the pixel coordinates, x and y, with respect to the distance parameter ρ in $t-\vec{s}$ space. They can also be referred to as pixel velocities with the following fact kept in mind, that is, ρ relates to both t and $\vec{s}$. Equation (4.9) is a governing equation which the optical flow quantities, i.e., the pixel velocities μ and ν have to obey. After assuming the straight line, passing through points $(t_1, \vec{s}_1)$ and $(t_2, \vec{s}_2)$ in $t - \vec{s}$ space, and introducing parameter ρ, representing point along the line from point $(t_1, \vec{s}_1)$, hence simplifying the derivation, we are now going to discuss how to solve unified optical flow field quantities. Similarly to Section 4.1, the discussion will be conducted with respect to several cases.

Case 1: If $\Delta\vec{s} = 0$, i.e., the sensor is static in a fixed spatial position. As analyzed in Case 1 in Section 4.1, the brightness "time"-invariance equation, i.e., Equation (4.5) can be rewritten below

$$\frac{\partial g}{\partial x}u + \frac{\partial g}{\partial y}v + \frac{\partial g}{\partial t} = 0 \tag{4.13}$$

In comparison of Equation (4.13) with Equation (4.9), it is noted that μ and ν become u and v, respectively, while $\frac{\partial g}{\partial \rho}$ changes to $\frac{\partial g}{\partial t}$. Obviously, with only one equation available, it is impossible to solve the two unknowns. Hence, some constraint must be imposed. Among several choices [15], the most popular one is the smoothness constraint which is first utilized by Horn and Schunck. Thus the next two equations are derived

$$g_x^2 u + g_x g_y v = \alpha_1^2 \nabla^2 u - g_x g_t \tag{4.14}$$

$$g_x g_y u + g_y^2 v = \alpha_1^2 \nabla^2 v - g_y g_t \tag{4.15}$$

where α_1 is a constant, g_x and g_y are defined in Equations (4.12), $g_t \triangleq \frac{\partial g}{\partial t}$. It is known that g_x, g_y and g_t can be estimated from the image sequence [9]. The pixel velocities u and v can then be solved by using the iterative algorithm developed in [9].

Case 2: In order to discuss the case when $\Delta t = 0$ more specifically, see Figure 4.1. In Figure 4.1, an imaging setting is shown. There the optical centers of the left and the right sensors are denoted by O and O^R, respectively. The axis OZ is the optical axis of the left sensor, while $O^R Z^R$ is that of the right sensor. These two optical axes are coplanar. The $O - XYZ$ and $O^R - X^R Y^R Z^R$ are, respectively, the Cartesian coordinate systems with axes OX and $O^R X^R$ being on the same plane as that of $OZO^R Z^R$ and axes OY and $O^R Y^R$ perpendicular to the plane of $OZO^R Z^R$. For simplicity it is further assumed that the left sensor is fixed, while the right sensor is only allowed to be translated along the straight line OO^R and to be rotated around $O^R Y^R$ axis. The distance between OO^R is denoted by l, while the angle between $O^R Z^R$ and OZ is by θ. Because $\Delta t = 0$, ρ is only a function of $\vec{s}$ at the specific moment t. Clearly, $\rho = \sqrt{l^2 + \chi^2\theta^2}$ where χ is a characteristic length that is determined according to the imaging setting parameters. We define $s \triangleq \sqrt{l^2 + \chi^2\theta^2}$. Equation (4.9) becomes

$$g_x u^s + g_y v^s + g_s = 0 \tag{4.16}$$

where g_x and g_y are defined in Equations (4.12), and

$$g_s \triangleq \frac{\partial g}{\partial s} \tag{4.17}$$

u^s and v^s are, respectively, the spatial variation rates of Δx and Δy with respect to Δs, i.e., $u^s \triangleq \frac{\partial x}{\partial s}$, $v^s \triangleq \frac{\partial y}{\partial s}$. The Δx and Δy are defined as

$$\Delta x \triangleq x^R - x^L \qquad \Delta y \triangleq y^R - y^L \tag{4.18}$$

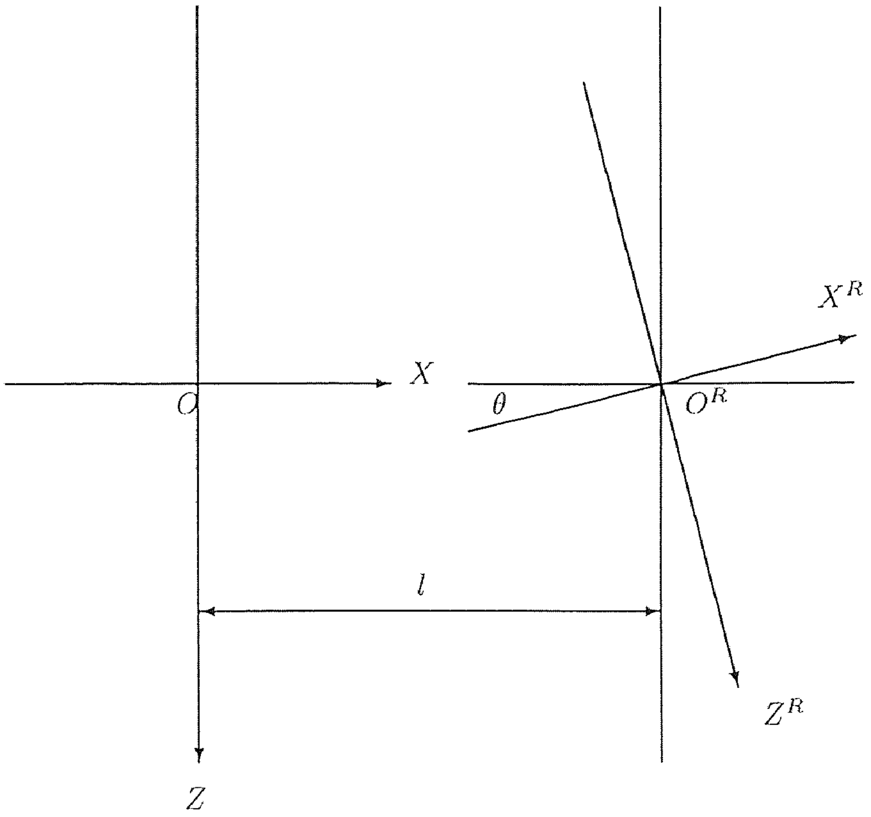


Figure 4.1 Imaging geometry

where the coordinates (x^R, y^R) and (x^L, y^L) are representing the two pixels on the right and the left images, respectively, such that they are corresponding to the same world point in 3-D space.

Equation (4.16) is the equation for a spatial sequence of images which is discussed in detail in [19] [20]. As pointed out in [19] [20], when the optical axes of both the left and right sensors are coplanar as shown in Figure 4.1, and the scene is far from the sensors that Z can be considered equal to Z^R, the next equation holds

$$v^s = 0 \tag{4.19}$$

We then have

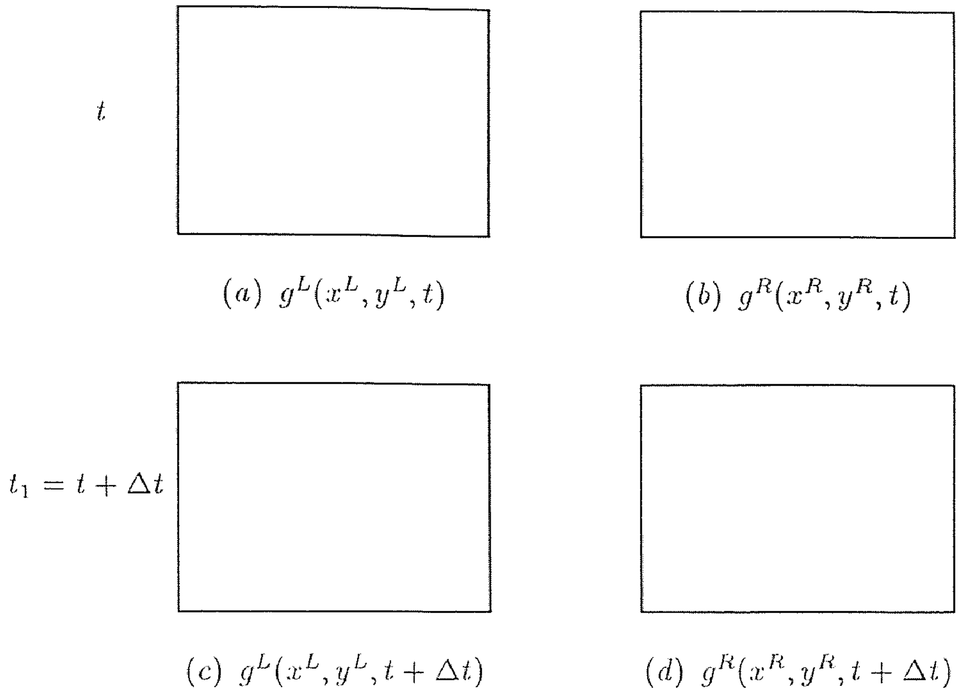
$$u^s = -\frac{g_s}{g_x} \tag{4.20}$$

Since g_s and g_x can be obtained from the given pair of stereo images, u^s can therefore be solved from the above equation.

Case 3: Assume imaging geometry is the same as discussed in the above case. Consider a rectangular four-frame model that is shown in Figure 4.2. There images (a), (b), (c) and (d) are chosen from a stereo image sequence. Images (a) and (c) can be viewed as two images from a monocular image sequence taken with a left sensor, while images (b) and (d) can be thought as two images from a monocular image sequence generated with a right sensor. Images (a) and (b) are taken at t_1 moment, while images (c) and (d) are at $t_1 + \Delta t$ moment.

Recognizing that imaging geometry from images (b) to (d) belongs to Case 1, we can apply Equations(4.14-4.15) to images (b) and (d), resulting in the following two quantities: u^R and v^R where the superscript R indicates that the quantities associated with are with respect to the right image sequence.

Similarly, applying Equations (4.14-4.15) to images (a) and (c), we obtain u^L and v^L which are, respectively, the counterparts of u^R, v^R defined for the left image sequence.



$(a)\ g^L(x^L, y^L, t)$ $(b)\ g^R(x^R, y^R, t)$

$(c)\ g^L(x^L, y^L, t + \Delta t)$ $(d)\ g^R(x^R, y^R, t + \Delta t)$

Figure 4.2 Four frame model

Recognizing that imaging geometry from images (a) to (b) is Case 2 and applying Equations (4.19-4.20) to images (a) and (b), we get u^s and v^s.

Based on these six field quantities of the unified optical flow field, i.e., u^R, v^R, u^L, v^L, u^s and v^s, another set equations can be derived with which 3-D motion can be recovered. A detailed discussion in this regard is contained in the next section.

4.3 A New Approach to Motion Analysis Using UOFF

In order to reconstruct 3-D motion field we need to determine the following six quantities: X, Y, Z, $\dot{X}$, $\dot{Y}$, $\dot{Z}$. From the preceding section, we already have u^L, v^L, u^s, v^s available for each pixel on image (a) in Figure 4.2, and u^R, v^R available for each pixel on image (b) in Figure 4.2. Now we shall discuss the relation between these two sets of quantities.

To reconstruct 3-D motion field we can start from either image in the pair of stereo images. For instance, we can start from a pixel in the left image, say the pixel located on the north-west corner of the left image. So, besides the six known field quantities, u^L, v^L, u^R, v^R u^s and v^s, we also have x^L, y^L available. On the other hand, in order to use $u^R(x^R, y^R, t)$ and $v^R(x^R, y^R, t)$ in reconstruction of 3-D motion, we have to know x^R and y^R which correspond to x^L and y^L, respectively, i.e., both image points (x^L, y^L) and (x^R, y^R) are related to the same world point in 3-D space. In other words, eight quantities: u^L, v^L, u^R, v^R u^s, v^s, x^L and y^L are available while eight quantities: X, Y, Z, $\dot{X}$, $\dot{Y}$, $\dot{Z}$, x^R a nd y^R are required to be solved.

Now consider a pair of (u^L, v^L) and a pair of (u^R, v^R) on image planes such that they are associated with the same world point in 3-D space. That is, u^L, v^L are the velocities of a pixel on the image (a); u^R, v^R are the velocity of a pixel on image (b); such that these two pixels are related to the same world point in 3-D space. Hence, there exists an inherent relation between them. This relation can be shown below.

$$\frac{1}{f}u^R(x^R, y^R, t) = \frac{1}{f}u^L(x^L, y^L, t)\cos\theta + \frac{l\,\dot{Z}}{Z^2}\cos\theta \qquad (4.21)$$

$$\frac{1}{f}v^R(x^R, y^R, t) = \frac{1}{f}v^L(x^L, y^L, t) \qquad (4.22)$$

See Appendix A.1 for the derivation.

Equations (4.23-4.24) relating the spatial variation rates, u^s and v^s, to the imaging setting parameters, l, θ and f, the characteristic length, χ, and 3-D space coordinate, X, Y and Z, are derived in Appendix A.2. There the approximation, i.e., equating derivative and average variation rate, has been made when Equation (A.3) is established. The similar approximation is made in [9].

$$\frac{u^s}{f} = -\left(\frac{l}{Z} + \frac{2(1-\cos\theta)}{\sin\theta}\right)/\sqrt{l^2 + \chi^2\theta^2} \qquad (4.23)$$

$$\frac{v^s}{f} = 0 \qquad (4.24)$$

As mentioned at the beginning of this section, the six quantities in the unified optical flow field have been calculated at this point. Hence, u^s is available. Through the use of Equation (4.23) the coordinate along the depth, Z, can be found for the world point corresponding to the north-west corner pixel in the left image. Via the use of the formulae of perspective projection, i.e., Equations (4.31) and (4.31) the values of the other two coordinates, X and Y, of this world point in 3-D space can be recovered.

Again the approximation, replacing derivative by average variation rate, is made in deriving the following two equations.

$$u^s = \frac{x^R - x^L}{\sqrt{l^2 + \chi^2\theta^2}} \qquad (4.25)$$

$$v^s = \frac{y^R - y^L}{\sqrt{l^2 + \chi^2\theta^2}} \qquad (4.26)$$

These two equations become a bridge in determining the corresponding x^R and y^R from x^L and y^L.

With x^R and y^R determined, $u^R(x^R, y^R, t)$ is then available. Therefore, $\dot{Z}$ can be solved from Equation (4.21). Furthermore, the following two equations derived in Appendix A.3

$$u^L = (\frac{\dot{X}}{Z} - \frac{\dot{Z}}{Z}\frac{X}{Z})f \tag{4.27}$$

$$v^L = (\frac{\dot{Y}}{Z} - \frac{\dot{Z}}{Z}\frac{Y}{Z})f \tag{4.28}$$

can then be used to solve for $\dot{X}$ and $\dot{Y}$.

Up to this point, the eight equations, i.e., Equations (4.30-4.31, 4.21, 4.23, 4.25-4.28), relating the eight known and the eight unknown quantities defined at the beginning of this section have been established. It is also seen that all of the six quantities, X, Y, Z, $\dot{X}$, $\dot{Y}$, and $\dot{Z}$, describing the 3-D motion of the world point associated with the north-west corner in the left image have been calculated. The same computation procedure can be conducted with respect to every pixel in the left image in a certain order. The order via which the computation is carried out could be row-by-row, column-by-column, or diagonal-by-diagonal.

Thus we see that as long as the smoothness constraint is satisfied the proposed technique can recover the whole 3-D motion field directly, i.e., the motion of world points in 3-D space that are projected onto image (a). It is obvious that in the above-described 3-D motion reconstruction the role played by image (a) and that by image (b) are interchangeable, i.e, we also can begin the reconstruction from pixels in image (b). It is also seen that although we start the development of the set of equations for motion analysis from the corresponding pairs, (u^L, v^L) and (u^R, v^R), there is no need for feature correspondence in the reconstruction of 3-D motion field.

In summary, the set of equations used for reconstructing the six 3-D motion parameters is listed below.

$$Z = -\frac{fl\sin\theta\cos\theta}{u^s\sqrt{l^2 + \chi^2\theta^2}\sin\theta + 2f(1 - \cos\theta)} \tag{4.29}$$

$$X = \frac{x^L Z}{f} \tag{4.30}$$

$$Y = \frac{y^L Z}{f} \tag{4.31}$$

$$x^R = u^s \sqrt{l^2 + \chi^2 \theta^2} + x^L \tag{4.32}$$

$$y^R = y^L \tag{4.33}$$

$$\dot{Z} = \frac{(u^R - u^L \cos\theta) Z^2}{l f \cos\theta} \tag{4.34}$$

$$\dot{X} = \frac{u^L Z}{f} + \frac{\dot{Z} X}{Z} \tag{4.35}$$

$$\dot{Y} = \frac{v^L Z}{f} + \frac{\dot{Z} Y}{Z} \tag{4.36}$$

It can be seen that in Equations (4.29-4.36) all quantities on the right-hand sides are available because they are either the imaging setting parameters, the unified optical flow field quantities, or the 3-D coordinates and their derivatives which have been solved by the preceding equations. Quantities x^R and y^R on the left-hand sides of Equations (4.32-4.33), respectively, are used to determine which value of $u^R(x^R, y^R, t)$ should be utilized in Equation (4.34) to calculate $\dot{Z}$. All quantities on the left-hand sides of Equations (4.29-4.31, 4.34-4.36), being the motion parameters: position and velocity in 3-D space, can therefore be solved straightforward. The derivation of UOFF is based on intensity constant equation , i.e., the gradient-based approach. But, when u^L, v^L, u^R, v^R, u^s,and v^s are determined in other techniques, we still can use UOFF to recover the velocity and position of objects.

4.4 Nine-frame Model

In the last chapter, the correlation-feedback approach is developed. It is proved that the approach is more efficient for computing optical flow field. Using the approach, one needs three images to compute an optical flow field. The three images are taken by three cameras that are shown in Figure 4.3.

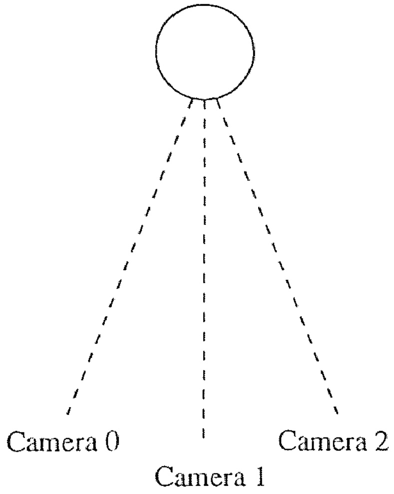


Figure 4.3 Camera Model

The angle between two camera is θ. The selection of θ is very important. When intensity variation is large, θ should be relatively small. When intensity variation is small, θ should be larger. Otherwise, optical flow computation would be affected. The reason has been discussed in Section 2.1.1. Usually, we select $\theta = 2.5^o$. As time increases, a stereo image sequence is obtained. Figure 4.4 shows that nine-frame model when time increments from t_0 to t_2.

From images I_{20}, I_{21}, and I_{22}, $u^s(x^M, y^M)$ and $v^s(x^M, y^M)$ can be computed, where x^M and y^M are coordinates of the image of middle camera. From images I_{01}, I_{11}, and I_{21}, $u^L(x^M, y^M)$ and $v^L(x^M, y^M)$ can be obtained. Similarly, $u^R(x^R, y^R)$ and $v^R(x^R, y^R)$ can be obtained from images I_{02}, I_{12}, and I_{22}. Actually, the nine-frame model only uses seven images. Furthermore, one can compute the 3-D motion and position from the six optical flow fields using Equations (4.29-4.36).

4.5 Experiment

The approach has been tested in many computer simulation experiments, including a rotating sphere and cone, with fairly good results [49]. Furthermore, many real

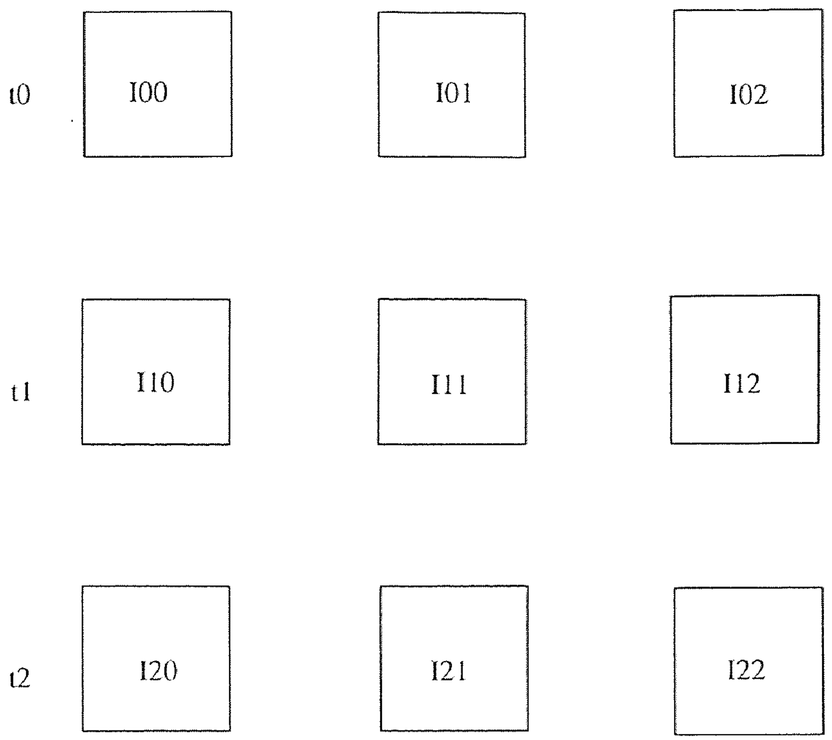


Figure 4.4 Nine-frame model

Table 4.1 3-D motion and position error

Error Type	Errors
RMS Z_{error}	3.92%
RMS X_{error}	4.09%
RMS Y_{error}	3.98%
Average Z_{error}	14.6%
Average X_{error}	3.17%
Average Y_{error}	0.82%

images were taken and experimented. In this section, we can not report every experiment, only one typical experiment is indicated.

Camera model is similar to the model shown in the last section. But, only one camera is used. The camera rotates at 2.5° along clockwise and counter-clockwise directions so that the three cameras model is simulated. The setting of the experiment is depicted in Figure 4.5.

The camera is rotated with respect to a center. The distance between the center and the camera is $1025mm$. The focal length of the camera is $12.5mm$. The box2 and box3 in Figure 4.5 move together in horizontal direction. The motion velocity is $5mm$/frame. Other dimensions are shown in Figure 4.5. Images are taken with a CCD SONY camera and DATACUBE system, and stored in a SUN SPARC station. The nine images are of 512×512 and are cut into 256×256. Via subsampling, the images are further compressed to 64×64. The compression is the same as that in last chapter. Three of the compressed images are shown in Figures 4.6, 4.7, and 4.8 that correspond I_{20}, I_{21}, and I_{22} in Figure 4.4.

After the tenth iteration of optical flow computation using the correlation-feedback algorithm, the six optical flow fields are used to compute 3-D position and motion. The errors of results are shown in Table 4.1.

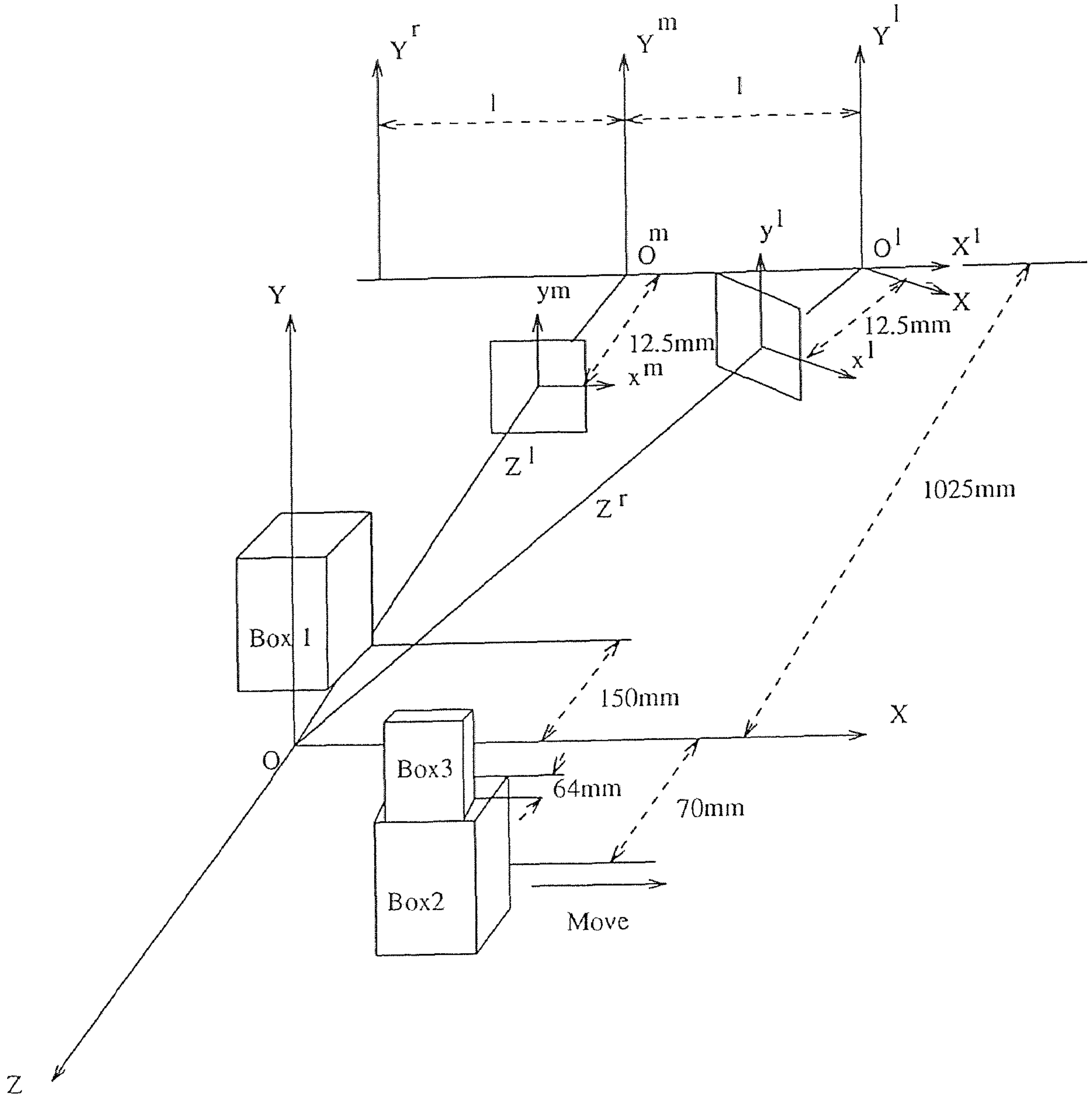


Figure 4.5 Experiment Setting.

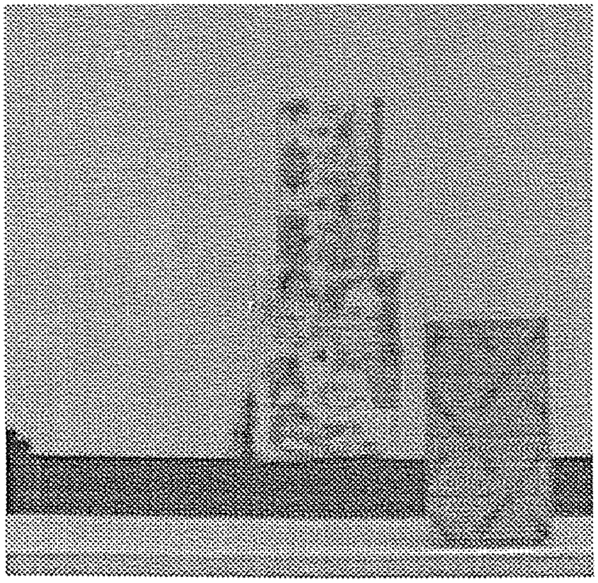


Figure 4.6 The left image

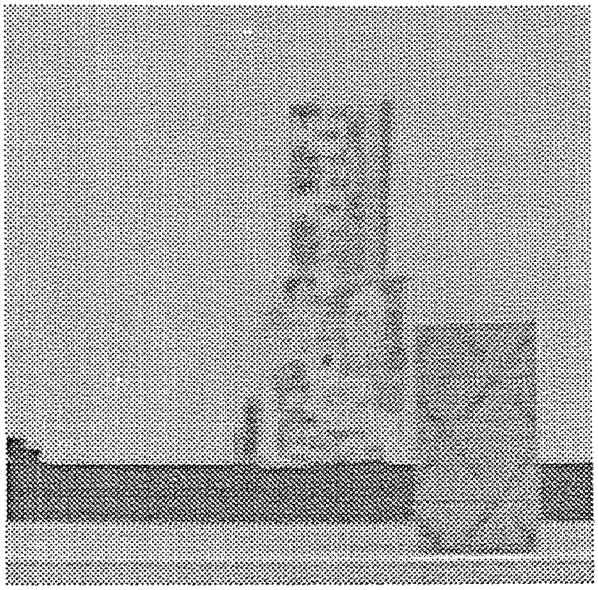


Figure 4.7 The middle image

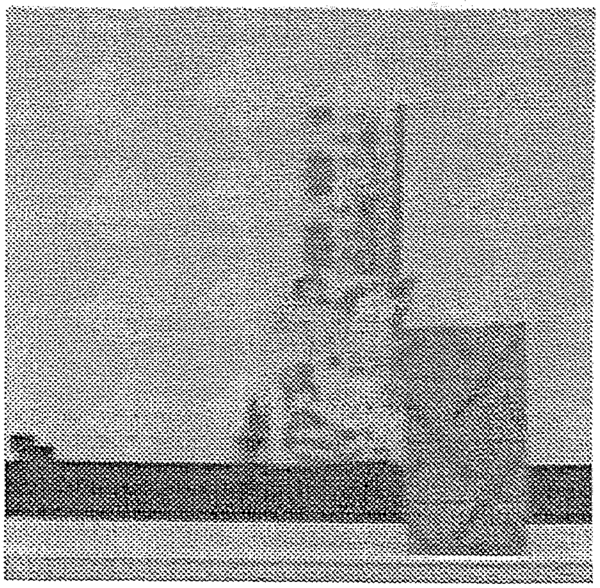


Figure 4.8 The right image

In Table 4.1,

$$RMS\ Z_{error} = \frac{\sqrt{\sum_{i=1}^{44}\sum_{j=1}^{44}((Z^a(i,j) - Z(i,j))^2}}{\sqrt{\sum_{i=1}^{44}\sum_{j=1}^{44}(Z^a(i,j))^2}} \tag{4.37}$$

where $Z^a(i,j)$ is true depth map.

$RMS\ X_{error}$ and $RMS\ Y_{error}$ can be obtained similarly.

$$Average\ \dot{Z}_{error} = \frac{1}{44}\sum_{i=1}^{44}\sum_{j=1}^{44}|(\dot{Z}(i,j) - \dot{Z}^a(i,j)| \tag{4.38}$$

where $\dot{Z}^a(i,j)$ is true velocity map in Z direction.

$Average\dot{X}_{error}$ and $Average\dot{Y}_{error}$ can be obtained similarly.

4.6 Conclusion

The results show that UOFF is feasible. Although the depth of objects is equal, the RMS error of depth map is 3.92%. But, the average error of $\dot{Z}$ is 14.6%. This is because that $\dot{Z}$ is approximately a function of the difference between u^R and u^L as noted in equation (4.34). The difference of optical flow would cause larger errors. The average error of $\dot{Z}$ will be reduced by further effort that will be discussed in next two chapters.

The UOFF approach has a few special features.

1. The approach allows objects to move and to change shape, i.e., it does not have any limitation to objects. Therefore, it can be used in more comprehensive systems.

2. The approach can compute 3-D position at each pixel so that the surface of the objects or structure of the objects can be known. From velocity of 3-D objects, one can know more information such as collision-time.

3. Consider the special case when $\theta = 0$, i.e., the case of two parallel cameras (see Figure 4.1). At this case, the spatial velocity $u \cdot unit$ is essentially the disparity, where *unit* is a constant and is a unit. The *unit* expresses how many millimeters it is when pixel number is one. The *unit* depends on CCD camera and sampling frequency. By using camera calibration, the image velocity (u) is obtained by detecting the corresponding points that in the left and right images have the same intensity and the depth D to a point with the image velocity u is measured. Thus

   $$unit = \frac{fl}{D * u} \tag{4.39}$$

   where unit of optical flow velocity u is pixel, f is the focal length, and l is the distance between two cameras (or the distance over which a camera moves).

   Once the *unit* is obtained and the optical flow field $u^s(x, y)$ is computed, the depth $Z(x, y)$ to a point with image velocity $u^s(x, y)$ is then determined from the static stereo equation: $Z = -fl/(u^s \cdot unit)$. Obviously, this special case of our approach has appeared in many literatures [4] [17] and, hence, is not new. As noted in Section 2.2.2, this depth map computation is based on only translational motion instead of both rotation and translation. We can prove that this depth map computation used in most of existing works has poor robustness as compared with Equation (4.29). From Figure 4.1, one knows that Equation (4.29) is based on both translational and rotational motion of a camera. In Equation (4.29), i.e.,

$$Z = -\frac{fl\sin\theta\cos\theta}{u^s\sqrt{l^2+\chi^2\theta^2}\sin\theta + 2f(1-\cos\theta)}, \tag{4.40}$$

the $\sqrt{l^2+\chi^2\theta^2} = unit$ using our experiment setting. Its derivative is

$$\frac{dZ}{du^s} = \frac{fl(unit)(\sin\theta)^2\cos\theta}{(u^s(unit)\sin\theta + 2f(1-\cos\theta))^2}. \tag{4.41}$$

When

$$Z = -\frac{fl}{u^s(unit)} \tag{4.42}$$

$$\frac{dZ}{du^s} = \frac{fl}{(unit)(u^s)^2} \tag{4.43}$$

The sensitivity of depth is $\frac{dZ}{du^s}\delta_u$, where δ_u is a perturbation of u^s. When Equation (4.41) is divided by Equation (4.43), one has

$$\begin{aligned}\lambda &= \left(\frac{fl(unit)(\sin\theta)^2\cos\theta}{(u^s(unit)\sin\theta + 2f(1-\cos\theta))^2}\right)/\left(\frac{fl}{unit(u^s)^2}\right) \\ &= \cos\theta/\left(1+\frac{2f(1-\cos\theta)}{(unit)u^s\sin\theta}\right)^2. \end{aligned} \tag{4.44}$$

Usually, we use $\theta = 2.5^o$, $f = 12.5mm$ and $unit = 0.054mm/pixel$. Assuming $u^s = 1$, it is obtained that $\lambda = 0.0074$. It is evident that under the same perturbation of optical flow, the error using Equation (4.40) is as 0.74% small as that error using Equation (4.42).

From Section 2.2.1, depth can be computed by following equation:

$$Z = z_0[1 + \frac{z_0(u^2+v^2)^{1/2}}{V_H f}]^{-1}. \tag{4.45}$$

The equation has been discussed in Section 2.2.1. It considers two motions, translation and rotation. According to the imaging geometry shown in Figure 4.1 and the discussion in Section 2.2.1, one has $v = 0$, $u = u^s(unit)$, $V_H = l$, and $z_0 = \frac{l\cos\theta}{\sin\theta}$. Therefore,

$$\begin{aligned} Z &= (\frac{l\cos\theta}{\sin\theta})/(1+\frac{u^s(unit)\cos\theta}{f\sin\theta} \\ &= \frac{lf\cos\theta}{f\sin\theta+u^s(unit)\cos\theta} \end{aligned} \tag{4.46}$$

$$\frac{dZ}{du^s} = -\frac{lf(unit)(\cos\theta)^2}{(f\sin\theta+u^s(unit)\cos\theta)^2} \tag{4.47}$$

When Equation (4.41) is divided by equation (4.47), one has

$$\begin{aligned} \lambda &= (\frac{fl(unit)(\sin\theta)^2\cos\theta}{(u^s(unit)\sin\theta+2f(1-\cos\theta))^2})/(\frac{lf(unit)(\cos\theta)^2}{(f\sin\theta+u^s(unit)\cos\theta)^2}) \\ &= \frac{(\sin\theta)^2(f\sin\theta+u^s(unit)\cos\theta)^2}{(\cos\theta)^2(u^s(unit)\sin\theta+2f(1-\cos\theta))^2} \end{aligned} \tag{4.48}$$

When $\theta = 2.5^\circ$, $f = 12.5mm$, $unit = 0.054$ and $u^s = 1$,

$$\lambda = 1.00. \tag{4.49}$$

Apparently, the more motion make depth map computation more robust. This is why the depth map computation of UOFF is more robust than most existing works.

UOFF has so wide uses that further improvement will become important. In next two chapters, we will discuss how to improve the estimation of 3-D position and motion.

CHAPTER 5

DISCONTINUITY IMPROVEMENT

In the last chapter, UOFF is discussed. UOFF is totally different from existing approaches. It includes six 3-D fields that are $Z(x,y)$, $X(x,y)$, $Y(x,y)$, $\dot{Z}(x,y)$, $\dot{X}(x,y)$, and $\dot{Y}(x,y)$. The six 3-D fields describe not only 3-D motion but also 3-D structure information of the objects. The recovered objects can be rigid as well as nonrigid. Since this reconstruction method has wide applications, further improvement of its accuracy is very important. It is well known that the accuracy of motion estimation is significantly affected by the motion discontinuity. The 3-D motion discontinuity mainly depends on the moving boundaries of optical flow fields.

As discussed in Chapter 3, the feedback technique popularly utilized in the field of automatic control can be applied to determine optical flow, resulting in the correlation-feedback algorithm. It performs better than the gradient-based and correlation-based algorithms in terms of accuracy and conserving moving boundaries. However, the propagation stage is also carried out there. Hence, one concludes that all of the three algorithms will blur moving boundaries in optical flow field.

To further improve optical flow accuracy, it is necessary to address this issue. In this chapter, we propose a Kalman filtering for improving accuracy in determining optical flow along moving boundaries.

Firstly, a quantitative analysis on the error decreasing rate in determining optical flow using the correlation-based technique, i.e., the deceasing rate of estimated optical flow deviation from its true value is given. It concludes that this error decreasing rate is varied for different regions in an image plane: it is larger for the regions where intensity varies more drastically, it is smaller for those where intensity varies more smoothly. This indicates that the iterations needed in optical flow determination should not be uniform for different image regions. That is, for the moving boundaries, where intensity usually changes bigger, less iterations are needed than for other regions. This is reasonable. In fact, the confidence measure [40] [44] is usually high along moving boundaries since richer information exists there. Therefore, an optical flow algorithm needs to have less iterations along moving boundaries than in other areas so that the better estimations of optical flow along boundaries can be propagated into other areas instead of being blurred by those in other areas.

Secondly, we propose a Kalman filter to realize the task of applying different number of necessary iterations in determining optical flow to deblur boundary and enhance accuracy. Loosely speaking, the idea is whenever the deviation of flow at a pixel is less than certain criterion, i.e., good accuracy has been achieved, the Kalman filter will not further update optical flow at this pixel, thus conserving accuracy along moving boundaries. Assuming that estimated optical flow field is contaminated by a Gaussian white noise, we give appropriate considerations to the system and measurement noise covariance matrices, Q^u and R^u, respectively. In this way, the Kalman filter is used to eliminate noise, raise accuracy and refine accuracy along discontinuities.

Finally, an experiment is presented to demonstrate the efficiency of our Kalman filter. Two objects are considered. One is stationary, while another is in translation. Unified optical flow filed (UOFF) quantities are determined by using the proposed technique. The 3-D position and speeds are then estimated by using UOFF approach. Both results obtained with and without the Kalman filter are given. A more than 10% improvement is achieved. It is expected that the more moving boundaries in the scene, the more effectively the scheme works.

5.1 A Quantitative Error Analysis

In Chapter 3, the feedback technique in determining optical flow is developed. There applying feedback technique to the correlation-based approach to optical flow computation has resulted in a correlation-feedback algorithm. It has been shown that the correlation-feedback algorithm performs better than the gradient-based and correlation-based algorithms in general. It can estimate sub-pixel image vectors to raise accuracy of estimated optical flow. It can conserve the discontinuities in optical flow quite well. But, since the neighborhood propagation is carried out in the algorithm, it is inevitable that the boundary information is somehow lost as the number of iterations increases. Therefore it is evident that all of three types of optical flow algorithms: the gradient-based, correlation-based, and correlation-feedback algorithms blur the motion boundaries although the correlation-feedback algorithm conserves the motion boundaries better than others. When such motion discontinuities appear in most of optical flow field, the error will be serious.

In order to improve the accuracy of optical flow around motion discontinuities, we should, first of all, investigate why boundary information is blurred. In this section, we conduct a quantitative analysis of error in determining optical flow along the moving boundaries for, say, the correlation-feedback algorithm.

For the sake of simplicity, only is one dimensional case considered here.

Assume the first given digital image can be described by

$$I_1(x_1) = I(x - u^a) \tag{5.1}$$

where I is a virtual, continuous image field, u^a is a true image velocity component along x direction. The second given digital image can be characterized by

$$I_2(x_2) = I(x). \tag{5.2}$$

According to Section 3.1.3, the error or the dissimilarity measure in the correlation can be calculated as follows.

$$\begin{aligned} E(u) &= \int [I_2(x_2+\lambda) - I_1(x_1+\lambda+u)]^2 w(\lambda) d\lambda \\ &= \int [I(x+\lambda) - I(x+\lambda-u^a+u)]^2 w(\lambda) d\lambda. \end{aligned} \tag{5.3}$$

where u is the estimated optical flow. The Taylor series expansion leads to

$$I(x-u^a+u) = I(x) + I_x(x)(u-u^a) + \theta(x). \tag{5.4}$$

where I_x represents the derivative of I with respect to x, $\theta(x)$ the second and higher order terms. Ignoring $\theta(x)$ and assuming that $(u^a - u)$ is a spatial constant in the vicinity of x, the error $E(u)$ can then be written as

$$\begin{aligned} E(u) &= \int [I(x+\lambda) - I(x+\lambda) - (u-u^a)I_x(x+\lambda)]^2 w(\lambda) d\lambda \\ &= (u^a-u)^2 \bar{I}_x^2. \end{aligned}$$

where

$$\bar{I}_x^2 = \int [I_x(x+\lambda)]^2 w(\lambda) d\lambda. \tag{5.5}$$

Thus, according to Section 3.1.3, the image vector can be expressed as following,

$$\begin{aligned} u^{(k+1)} &= \frac{\sum_u e^{-E(u)} u}{\sum_u e^{-E(u)}} \\ &= \frac{\sum_u e^{-\bar{I}_x^2 (u-u^a)^2} u}{\sum_u e^{-\bar{I}_x^2 (u-u^a)^2}}, \end{aligned} \tag{5.6}$$

where $e^{-E(u)}$ is the so-called response distribution and

$$u \in \{u_0^{(k)}, u_1^{(k)}, u_2^{(k)}, u_3^{(k)}, u_4^{(k)}\}, \tag{5.7}$$

with k being an iteration number, $u_0^{(k)}, u_1^{(k)}, u_2^{(k)}, u_3^{(k)}$, and $u_4^{(k)}$ being the five independent random variables obeying the same Gaussian distribution, i.e., $N(u^a, \delta_{u,k})$. Here, deviation of $u^{(k)}$ is $\delta_{u,k}$,

$$\delta_{u,k+1}^2 = E\{(u^{(k+1)} - u^a)^2\}$$

$$\begin{aligned} &= E\{(\frac{\sum_{u^{(k)}} e^{-\bar{I}_x^2(u^{(k)}-u^a)^2} u^{(k)}}{\sum_{u^{(k)}} e^{-\bar{I}_x^2(u^{(k)}-u^a)^2}} - u^a)^2\} \\ &= E\{\frac{(\sum_{u^{(k)}} e^{-\bar{I}_x^2(u^{(k)}-u^a)^2}(u^{(k)} - u^a))^2}{(\sum_{u^{(k)}} e^{-\bar{I}_x^2(u^{(k)}-u^a)^2})^2}\} \qquad (5.8) \end{aligned}$$

According to the mean-value theorem [48], if $f(x)$ and $g(x)$ are continuous on [a,b], and $g(x) \neq 0$ on [a, b], there exists a value c of in (a, b) such that

$$\int_a^b f(x)g(x)dx = f(c)\int_a^b g(x)dx. \qquad (5.9)$$

Therefore, one has

$$\delta_{u,k+1}^2 = \frac{E\{(\sum_{u^{(k)}} e^{-\bar{I}_x^2(u^{(k)}-u^a)^2}(u^{(k)} - u^a))^2\}}{(\sum_{c_i} e^{-\bar{I}_x^2(c_i-u^a)^2})^2}, \qquad (5.10)$$

where c_i is a mean-value that u_i^k assumes, $i = 0, 1, 2, 3, 4$.

Since $u_0^{(k)}$,$u_1^{(k)}$,$u_2^{(k)}$,$u_3^{(k)}$,and $u_4^{(k)}$ are independent of each other, and the covariances between $e^{-\bar{I}_x^2(u_i^{(k)}-u^a)^2}(u_i^k - u^a)^2$ are zeros, where $i = 0, 1, 2, 3, 4$. (A detailed derivation of this is shown in Appendix B), Equation (5.10) can be written as

$$\delta_{u,k+1}^2 = \frac{\sum_{u^{(k)}} E\{e^{-2\bar{I}_x^2(u^{(k)}-u^a)^2}(u^{(k)} - u^a)^2\}}{(\sum_{c_i} e^{-\bar{I}_x^2(c_i-u^a)^2})^2}. \qquad (5.11)$$

Since $u_0^{(k)}$,$u_1^{(k)}$,$u_2^{(k)}$,$u_3^{(k)}$,and $u_4^{(k)}$ have the same Gaussian distribution whose mean is u^a and variance is $\delta_{u,k}^2$,

$$\begin{aligned} &\sum_{u^{(k)}} E\{e^{-2\bar{I}_x^2(u^{(k)}-u^a)^2}(u^{(k)} - u^a)^2\} \\ &= \frac{5}{\delta_{u,k}\sqrt{2\pi}}\int_{-\infty}^{\infty}(u^{(k)} - u^a)^2 e^{-2\bar{I}_x^2(u^{(k)}-u^a)^2} e^{\frac{-(u^{(k)}-u^a)^2}{2\delta_{u,k}^2}} du^{(k)} \\ &= \frac{5}{\delta_{u,k}\sqrt{2\pi}} \frac{\sqrt{\frac{2\pi}{4\bar{I}_x^2+\delta_{u,k}^{-2}}}}{\sqrt{\frac{2\pi}{4\bar{I}_x^2+\delta_{u,k}^{-2}}}} \int_{-\infty}^{\infty}(u^{(k)} - u^a)^2 e^{\frac{\frac{-(u^{(k)}-u^a)^2}{2}}{4\bar{I}_x^2+\delta_{u,k}^{-2}}} du^{(k)} \\ &= \frac{5}{\delta_{u,k}\sqrt{2\pi}}\sqrt{\frac{2\pi}{4\bar{I}_x^2 + \delta_{u,k}^{-2}}}\frac{1}{4\bar{I}_x^2 + \delta_{u,k}^{-2}} \\ &= \frac{5}{\delta_{u,k}(\delta_{u,k}^{-2} + 4\bar{I}_x^2)^{\frac{3}{2}}} \qquad (5.12) \end{aligned}$$

Hence,

$$\delta_{u,k+1}^2 = \frac{1}{\sum_{c_i} e^{-\bar{I}_x^2(c_i-u^a)^2}} \frac{5}{\delta_{u,k}(\delta_{u,k}^{-2}+4\bar{I}_x^2)^{\frac{3}{2}}} \tag{5.13}$$

Assuming

$$\frac{1}{\sum_{c_i} e^{-\bar{I}_x^2(c_i-u^a)^2}} = c, \tag{5.14}$$

where, obviously, $c\epsilon(\frac{1}{5},\infty)$, and one has

$$\begin{aligned} \delta_{u,k+1}^2 &= \frac{5c}{\delta_{u,k}(\delta_{u,k}^{-2}+4\bar{I}_x^2)^{\frac{3}{2}}} \\ &= \frac{5c\delta_{u,k}^2}{(4\bar{I}_x^2\delta_{u,k}^2+1)^{\frac{3}{2}}} \end{aligned} \tag{5.15}$$

or,

$$\frac{\delta_{u,k+1}^2}{\delta_{u,k}^2} = \frac{5c}{(4\bar{I}_x^2\delta_{u,k}^2+1)^{\frac{3}{2}}} \tag{5.16}$$

Equation (5.16) indicates that the larger the $\bar{I}_x^2$, the faster the deviation decreases as iteration number k increases. Furthermore, as the iteration number increases, even for the case where the $\bar{I}_x^2$ is large, the decreasing rate of the deviation gradually slows down. In Figure 5.1, a curve for the convergence rate, when $\bar{I}_x = 40$,

On the other hand, when $\bar{I}_x^2$ is small, much more iterations are needed to make the deviation less than the value expected. Figure 5.1 also shows the convergence rate when $\bar{I}_x = 1$.

We therefore conclude that the deviation decreasing rate or the convergence speed of the optical flow algorithm is non-homogeneous with respect to different pixels. When the variation of intensity in an image region is greater, the convergence in the region is faster. It is known that most of motion boundaries are of great intensity variation. Since the propagation step is to be carried out in the iteration procedure, as we know, the error increases along moving boundaries as iteration increases. In other words, there are two conflicting actions on the boundaries. That is, the conservation step tries to reduce the error, while the smoothness process increases the error after a few iterations. When the error reducing rate is less than

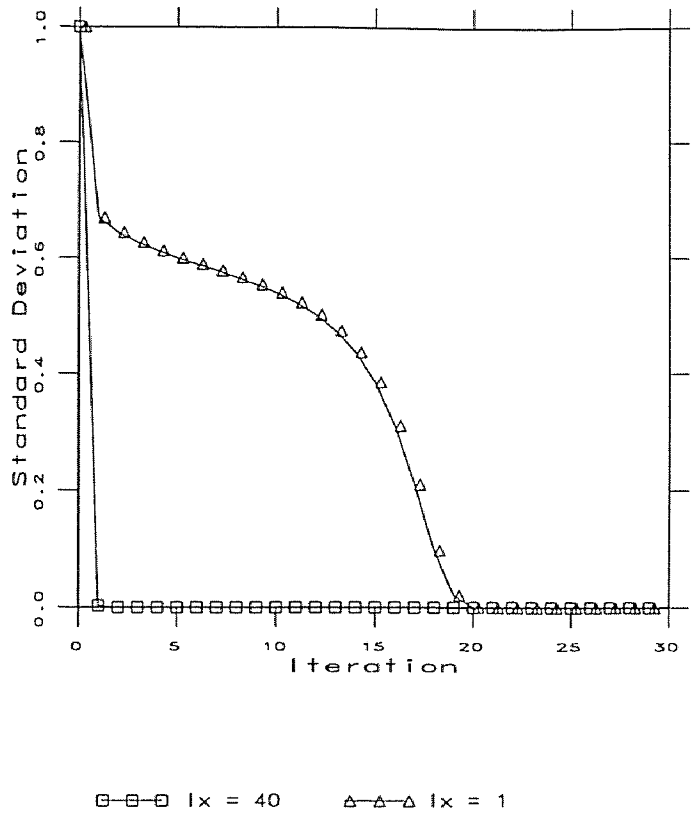


Figure 5.1 Convergence rates

Table 5.1 Kalman filter

Models	System model	$U_k = \Phi_{k-1} U_{k-1} + \eta_k.\ \eta_k \sim N(0, Q_k)$
	Measurement model	$D_k = HU_k + \xi_k.\ \xi_k \sim N(0, R_k)$
	Prior model	$E[U_0] = \hat{U}_0.\ cov[U_0] = P_o$
	(other assumption)	$E[\eta_k \xi_k^T] = 0$
Prediction phase	State estimate extrapolation	$\hat{U}_k^- = \Phi_{k-1}\ \hat{U}_{k-1}^+$
	State covariance extrapolation	$P_k^- = \Phi_{k-1} P_{k-1}^+ \Phi_{k-1}^T + Q_{k-1}$
Update phase	State estimate update	$\hat{U}_k^+ = \hat{U}_k^- + K_k(D_k - H_k \hat{U}_k^-)$
	State covariance update	$P_k^+ = (I - K_k H_k) P_k^-$
	Kalman gain matrix	$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$

the error increasing rate, the iteration procedure should be terminated. Therefore, the iteration number should be different for the regions that are of different intensity variations. A Kalman filter is proposed next to implement this task.

5.2 Kalman Filter

5.2.1 Kalman Filtering

The Kalman filter is best by almost any reasonable criterion in the Gaussian case [51]. If we consider the system that estimates the optical flow field is contaminated by the Gaussian white noise, we can use Kalman filter to eliminate noise, raise accuracy and refine discontinuity.

Kalman filtering is based on a linear measurement model and is operated in two phases: prediction phase and update phase, as shown in Table 5.1.

From the update phase, we can see that when the current measurement error $||R_k||$ increases, the gain K_k decreases. Since $\hat{U}_k^+ = \hat{U}_k^- + K_k(D_k - H_k \hat{U}_k^-)$, $\hat{U}_k^-$ is considered to be the dominant of $\hat{U}_k^+$. On the other hand, when $||R_k||$ decreases, K_k increases. The effect of $D_k - H_k \hat{U}_k^-$ is weighted more heavily. Therefore, the Kalman filtering optimally make the use of previous knowledge and current information.

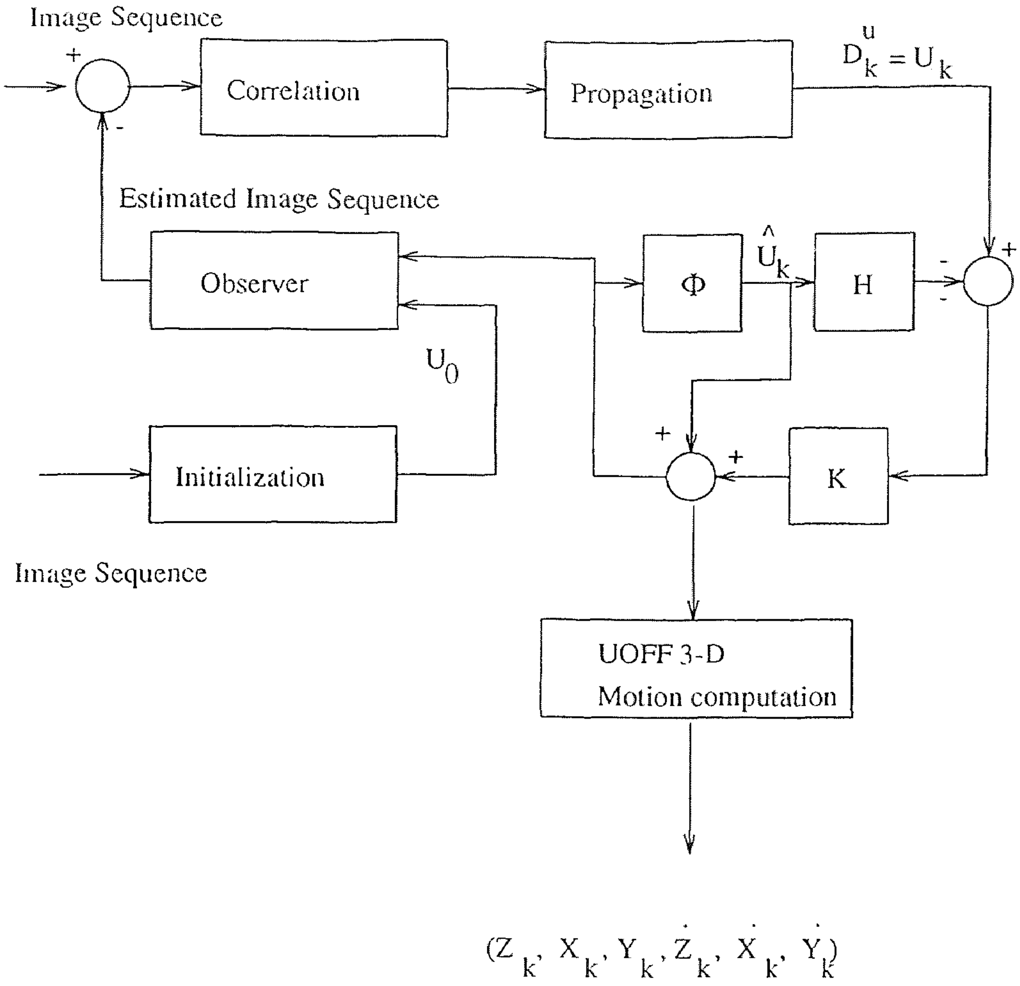


Figure 5.2 Framework

Here, we use this feature to raise accuracy and refine discontinuities of optical flow field.

5.2.2 Proposed Framework

Figure 5.2 shows a proposed framework. Figure 5.3 shows image sequence arrangement.

Though here Kalman filtering is used to raise accuracy of the optical flow field, our goal is to improve the three dimensional motion and position fields. Therefore, there is a block that performs UOFF 3-D motion computation in the Figure 5.2. Four

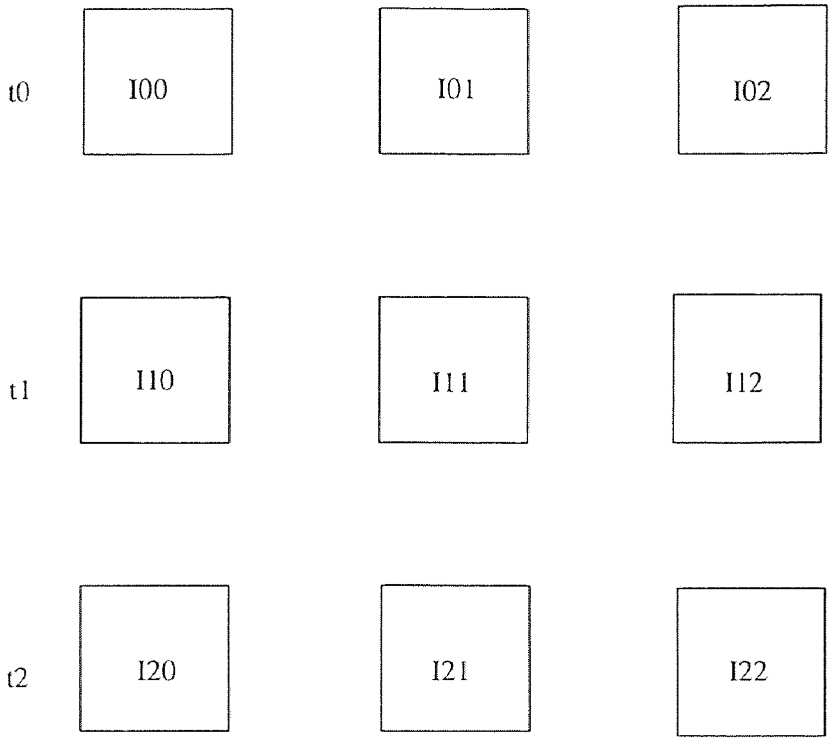


Figure 5.3 Stereo image sequence arrangement

UOFF optical flow fields $u^s(x,y)$, $u^L(x,y)$, $u^R(x,y)$, and $v^L(x,y)$ are considered as state variables because only these four variable fields are used to compute the three dimensional position and speed fields according to Equation (4.29-4.36).

In the linear measurement model of the Kalman filter,

$$
\begin{aligned}
U_k &= \Phi U_{k-1} + \eta_k^u, \quad \eta_k^u \sim N_k(0, Q_k^u) \\
D_k^u &= H U_k + \xi_k^u, \quad \xi_k^u \sim N_k(0, R_k^u) \\
& E\{\eta_k^u (\xi_k^u)^T\} = 0
\end{aligned}
\tag{5.17}
$$

where $U_k(x,y) = (u_k^s(x,y)\ u_k^L(x,y)\ u_k^R(x,y)\ v_k^L(x,y))^T$, Φ is chosen to be an identity matrix because the state variables at current moment should be equal to the state variables at the last moment, and H is also an identity matrix since the measurements equal the state variables in the Kalman filter.

The initial state is

$$
\hat{U}_0^+(x,y) = \begin{pmatrix} u_0^s(x,y) \\ u_0^L(x,y) \\ u_0^R(x,y) \\ v_0^L(x,y) \end{pmatrix}
\tag{5.18}
$$

where u_0^s, u_0^L, u_0^R, and v_0^L are obtained from the correlation-feedback stage at the first iteration.

$P_0^+ = I$. I stands for the identity matrix.

Since H and Φ are identity matrices, $D_k^u = U_{k-1} + \eta_k^u + \xi_k^u$. Theoretically, η_k^u is not necessary since ξ_k^u already includes its effect. But, we still let η_k^u exist because Q_k^u can be used to adjust system dynamic and steady state behaviors. The more parameters, the more flexible system adjustment. Thus, the selection of R_k^u will have more flexibility.

In our experiment, the covariance matrix Q_k^u for η_k^u is chosen as

$Q_k^u(x,y)$

$$= \begin{pmatrix} \sum\sum \frac{(\hat{u}1^{+}_{k-1}(i,j))^2}{9} & 0 & 0 & 0 \\ 0 & \sum\sum \frac{(\hat{u}2^{+}_{k-1}(i,j))^2}{9} & 0 & 0 \\ 0 & 0 & \sum\sum \frac{(\hat{u}3^{+}_{k-1}(i,j))^2}{9} & 0 \\ 0 & 0 & 0 & \sum\sum \frac{(\hat{u}4^{+}_{k-1}(i,j))^2}{9} \end{pmatrix} \tag{5.19}$$

where $i \in (x-1, x+1)$, $j \in (y-1, y+1)$, and

$$\hat{U}^{+}_{k-1}(x,y) = \begin{pmatrix} \hat{u}1^{+}_{k-1}(x,y) \\ \hat{u}2^{+}_{k-1}(x,y) \\ \hat{u}3^{+}_{k-1}(x,y) \\ \hat{u}4^{+}_{k-1}(x,y) \end{pmatrix} = \begin{pmatrix} \hat{u}^{s}_{k-1}(x,y) \\ \hat{u}^{L}_{k-1}(x,y) \\ \hat{u}^{R}_{k-1}(x,y) \\ \hat{v}^{L}_{k-1}(x,y) \end{pmatrix}. \tag{5.20}$$

According to the quantitative analysis conducted in Section 5.1, especially Equation (5.16), $R^{u}_{k}(x,y)$ is defined as following:

$$\begin{aligned} & R^{u}_{k}(x,y)^{u} \\ = & \begin{pmatrix} r11_k(x,y) & 0 & 0 & 0 \\ 0 & r22_k(x,y) & 0 & 0 \\ 0 & 0 & r33_K(x,y) & 0 \\ 0 & 0 & 0 & r44_k(x,y) \end{pmatrix} \\ = & \begin{pmatrix} \frac{5c^{s}\delta^2_{s,k-1}}{(4\bar{I}11^2_x\delta^2_{s,k-1}+1)^{\frac{3}{2}}} & 0 & 0 & 0 \\ 0 & \frac{5c^{L}\delta^2_{l,k-1}}{(4\bar{I}11^2_x\delta^2_{l,k-1}+1)^{\frac{3}{2}}} & 0 & 0 \\ 0 & 0 & \frac{5c^{R}\delta^2_{r,k-1}}{(4\bar{I}12^2_x\delta^2_{r,k-1}+1)^{\frac{3}{2}}} & 0 \\ 0 & 0 & 0 & \frac{5c^{V}\delta^2_{v,k-1}}{(4\bar{I}11^2_y\delta^2_{v,k-1}+1)^{\frac{3}{2}}} \end{pmatrix}, \end{aligned} \tag{5.21}$$

where $I11$ and $I12$ are two images shown in Figure 5.3, $\delta_{s,k-1}$, $\delta_{l,k-1}$, $\delta_{r,k-1}$, and $\delta_{v,k-1}$ are the standard deviations of u^{s}_{k-1}, u^{L}_{k-1}, u^{R}_{k-1}, and v^{L}_{k-1} respectively.

$$\bar{I}11_x = \frac{1}{6}\sum_{i=-1}^{1}(I11(x+1,y+i) - I11(x,y+i))^2 + (I11(x,y+i) - I11(x-1,y+i))^2, \tag{5.22}$$

$$\bar{I}12_x = \frac{1}{6}\sum_{i=-1}^{1}(I12(x+1,y+i) - I12(x,y+i))^2 + (I12(x,y+i) - I12(x-1,y+i))^2, \tag{5.23}$$

and

$$\bar{I1}_y = \frac{1}{6}\sum_{i=-1}^{1}(I1(x+i,y+1) - I1(x+i,y))^2 + (I1(x+i,y) - I1(x+i,y-1))^2. \tag{5.24}$$

In our experiments, c^s,c^L,c^R,and c^V are set to be equal to 5.

In the prediction phase of the Kalman filter,

$$\hat{U}_k^-(x,y) = \hat{U}_{k-1}^+(x,y) \tag{5.25}$$

$$P_k^-(x,y) = P_{k-1}^+(x,y) + Q_{k-1}^u(x,y) \tag{5.26}$$

In the update phase,

$$\hat{U}_k^+(x,y) = \hat{U}_k^-(x,y) + K_k(x,y)(D_k(x,y) - \hat{U}_k^-(x,y)), \tag{5.27}$$

$$K_k(x,y) = P_k^-(x,y)[P_k^-(x,y) + R_k^u(x,y)]^{-1}, \tag{5.28}$$

$$P_k^+(x,y) = (I - K_k(x,y))P_k^-(x,y). \tag{5.29}$$

Since R_k^u, Q_k^u, and P_0^+ are diagonal matrices, K_k, P_k^- and P_k^+ are also diagonal.

When the standard deviation of u_k^s

$$\delta_{s,k}(x,y) > T_s,$$

where T_s is a predefined threshold of $\delta_{s,k}(x,y)$, the iteration procedure continues. If $\delta_{s,k}(x,y) \leq T_s$, the iteration procedure should terminate at the pixel, otherwise the smoothness operation will blur the image vector at the pixel. Similarly, T_l, T_r, and T_v are defined as the thresholds for $\delta_{l,k}$, $\delta_{r,k}$, and $\delta_{v,k}$, respectively.

Hence, whenever a predefined threshold has been reached, it means at this iteration, $\hat{u}_k^+$ is a best estimate. At the pixel, the iteration should terminate. In fact, Kalman filter has the ability to automatically terminate the iteration as long as that $rii_k(x,y) = \infty$ is assumed when $\delta_{i,k}(x,y) < T_i$. From Equation (5.28), it is known that $kii_{k+1} = 0$. Then, that $\hat{u}1_{k+1}(x,y) = \hat{u}1_k(x,y)$ is obtained from Equation (5.27). Therefore, the iteration no long continues at the pixel (x,y). However, iteration still continues at the other pixels whose variances do not reach the minimum.

Therefore, finally

$$R_k^u(x,y) = \begin{pmatrix} r11_k(x,y) & 0 & 0 & 0 \\ 0 & r22_k(x,y) & 0 & 0 \\ 0 & 0 & r33_K(x,y) & 0 \\ 0 & 0 & 0 & r44_k(x,y) \end{pmatrix} \tag{5.30}$$

where

$$r11_k(x,y) = \begin{cases} \delta_{s,k}(x,y) & \text{if } \delta_{s,k} > T_s \\ \infty & \text{otherwise} \end{cases} \tag{5.31}$$

$$r22_k(x,y) = \begin{cases} \delta_{l,k}(x,y) & \text{if } \delta_{l,k} > T_l \\ \infty & \text{otherwise} \end{cases} \tag{5.32}$$

$$r33_k(x,y) = \begin{cases} \delta_{r,k}(x,y) & \text{if } \delta_{r,k} > T_r \\ \infty & \text{otherwise} \end{cases} \tag{5.33}$$

and

$$r44_k(x,y) = \begin{cases} \delta_{v,k}(x,y) & \text{if } \delta_{v,k} > T_v \\ \infty & \text{otherwise} \end{cases} \tag{5.34}$$

5.3 An Experiment

The experiment presented here is intended to examine the effectiveness of the actual boundary conservation achieved by our proposed Kalman filter. It compares the performances of the algorithm using the Kalman filter and without using the Kalman filter.

In the experiment, the camera setting, images obtained, i.e., Figures 4.5-4.8, and image compression are exactly the same as the experiment presented in the last chapter.

In order to compare the experiment results with true three dimensional parameters, Equations (4.29-4.36) are applied. Thus, the optical flow fields that are computed with Kalman filter and without Kalman filter are used to calculate three dimensional depth and velocity fields. Then the results are compared with each other. When we say "without Kalman filter", it means that outputs of the

Table 5.2 Result comparison

Error Type	No Kalman Filter	With Kalman Filter
RMS Z_{error}	3.92%	3.56%
RMS X_{error}	4.09%	3.50%
RMS Y_{error}	3.98%	3.58%
Average $\dot{Z}_{error}$	14.6%	15.15%
Average $\dot{X}_{error}$	3.17%	2.95%
Average $\dot{Y}_{error}$	0.82%	0.87%

correlation-feedback approach are used directly to compute three dimensional information using Equations (4.29-4.36). Table 5.2 shows the results.

In Table 5.2, the $RMS\ Z_{error}$ is defined as

$$RMS\ Z_{error} = \frac{\sqrt{\sum_{i=1}^{44}\sum_{j=1}^{44}(Z^a(i,j) - Z(i,j))^2}}{\sqrt{\sum_{i=1}^{64}\sum_{j=1}^{64}(Z^a(i,j))^2}} \tag{5.35}$$

where $Z^a(i,j)$ is true depth map. It is noted that only central portion of 44 × 44 within the images of 64 × 64 are considered in the calculation. $RMS\ X_{error}$ and $RMS\ Y_{error}$ can be obtained similarly.

$$Average\ \dot{Z}_{error} = \frac{1}{44}\sum_{i=1}^{44}\sum_{j=1}^{44}|(\dot{Z}(i,j) - \dot{Z}^a(i,j)| \tag{5.36}$$

where $\dot{Z}^a(i,j)$ is true velocity map in Z direction. The 44×44 has the same meaning as mentioned above. $Average\ \dot{X}_{error}$ and $Average\ \dot{Y}_{error}$ can be obtained similarly.

From Table 5.2, it is evident that precision of $Z(x,y)$, $X(x,y)$, $Y(x,y)$, and $\dot{X}(x,y)$ are improved, but that of $\dot{Z}(x,y)$ and $\dot{Y}(x,y)$ have not improved (instead, the error even increased a little bit). The improvements of $Z(x,y)$, $X(x,y)$, and $Y(x,y)$ are larger than that of $\dot{X}(x,y)$.

From equations(4.29 -4.36), Z is a function of u^s; $\dot{X}$ is a function of u^L; X and Y are linear functions of Z. Therefore, the errors of Z, X, and Y are determined by that of u^s. The error of $\dot{X}$ relied on u^L. As we know, u^L is an image velocity field that is caused by the motion of the objects instead of camera. However, from Figures 4.5-4.8, it is known that the motion of the objects is only the motion of the Box2 and Box3 in the case. The spatial optical flow along x direction, u^s, is caused by the motion of the camera with respect to a static scene. In the experiment, the motion of the camera causes the motion of the images of the Box1, Box2, and Box3. Therefore, the u^s flow field has more moving boundaries than u^L field does. Hence, the u^s field take more advantage of the proposed moving boundary conservation algorithm. This explains why the improvement on the accuracy of the recovery of Z, X and Y is larger than that on $\dot{X}$.

It is noted that the true values of $\dot{Z}(x,y)$ and $\dot{Y}(x,y)$ are zeros with respect to most of values of (x,y). There are very few discontinuities in the fields. Therefore, the proposed boundary conservation technique does not bring out advantages as the results in Table 5.2 show.

In fact, the effectiveness of boundary conservation can be controlled by the threshold values: T^s, T^l, T^r, and T^v. According to our above discussion in this experiment, we choose these threshold values as follows.

$$T^s = 1.56^{-5}, \quad (5.37)$$

$$T^l = 4.63^{-6}, \quad (5.38)$$

$$T^r = 4.63^{-6}, \quad (5.39)$$

$$T^v = 1.95^{-6}. \quad (5.40)$$

Actually, the trade-off between boundary conservation and interior smoothness is not easy. If there is not any prior knowledge, the selection of threshold will be difficult.

In order to show the effectiveness of the discontinuity conservation, Figure 5.4 are compared with Figure 5.5. Figure 5.4 shows depth map $Z(x,y)$ that is obtained with Kalman filter. Figure 5.5 is also depth map but obtained without the Kalman filter.

Clearly, the depth map with the Kalman filter conserve the discontinuities in this experiment much better than that without the Kalman filter.

5.4 Conclusion

As pointed out in [46], most of the optical flow computation algorithms can be divided into two steps: conservation step and neighborhood propagation step. The neighborhood propagation operation or smoothness operation is inevitable because of the presence of various noises and aperture problem. Most of the existing neighborhood propagation operations are the relaxation algorithms based on some kinds of optimal functions. However, the operations inevitably blur the boundary information. It is well known [40] that the confidence measure is usually high along moving boundaries since richer information exists there. In this chapter, our analysis indicates that the optical flow algorithm needs to have less iterations along moving boundaries than in other areas so that the better estimations of optical flow along boundaries can be propagated into other areas instead of being blurred by those in other areas.

In order to make the iteration procedure stop at a proper time, we define the elements in the main diagonal of the covariance matrix of the measurement noise, R^u, i.e., the variance of measurement noises as

$$rii_k(x,y) = \begin{cases} \delta_{i,k}(x,y) & \text{if } \delta_{i,k} > T_i \\ \infty & \text{otherwise} \end{cases} \quad . \tag{5.41}$$

where index i means the ith measurement noise, T_i the predefined threshold value for the ith measurement noise. When the error (standard deviation $\delta_{i,k}(x,y)$) is small enough, $rii_k(x,y) = \infty$. According to Equations (5.27, 5.28), the Kalman filter can automatically terminate the iteration in the pixel (x,y) so that boundary information is protected.

In this method, the derivation of the error decreasing rate is very important. Different optical flow computation algorithms have different error decreasing rates because their refinement procedures have different effects. For instance, when the

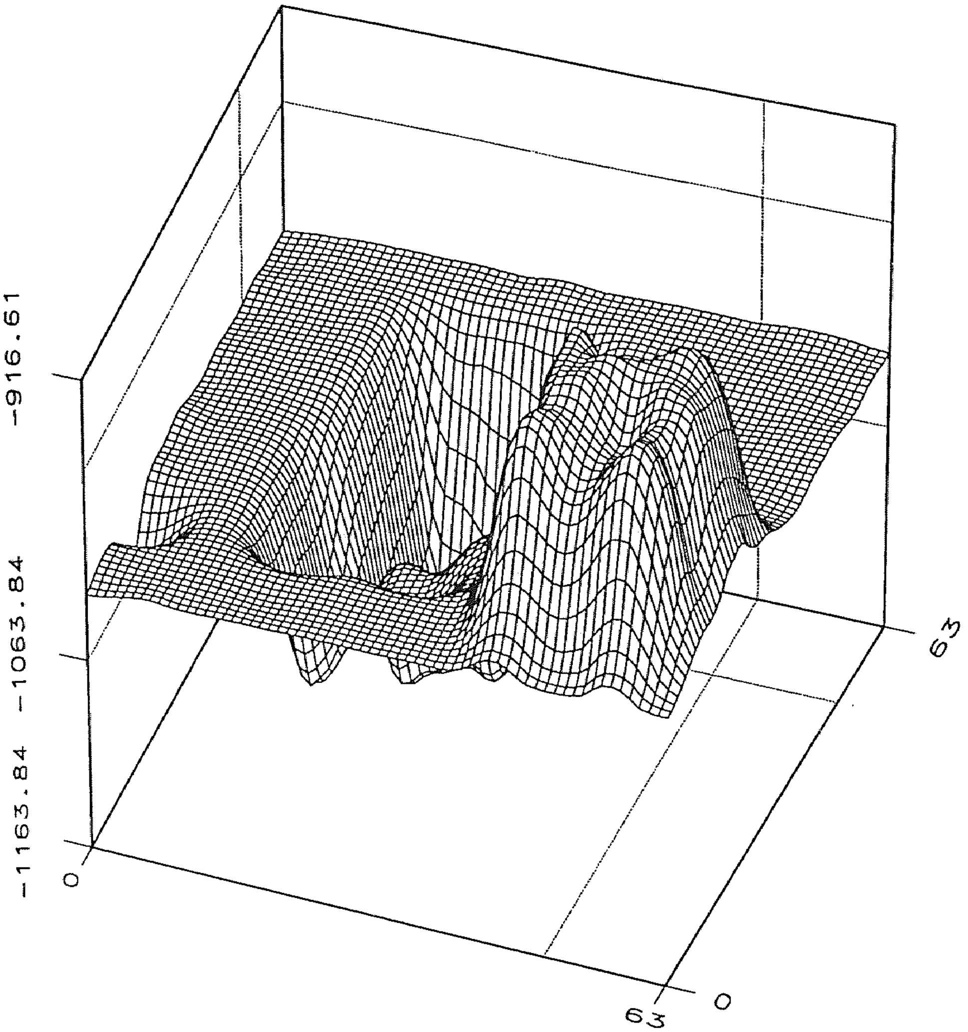


Figure 5.4 Depth map with Kalman filter.

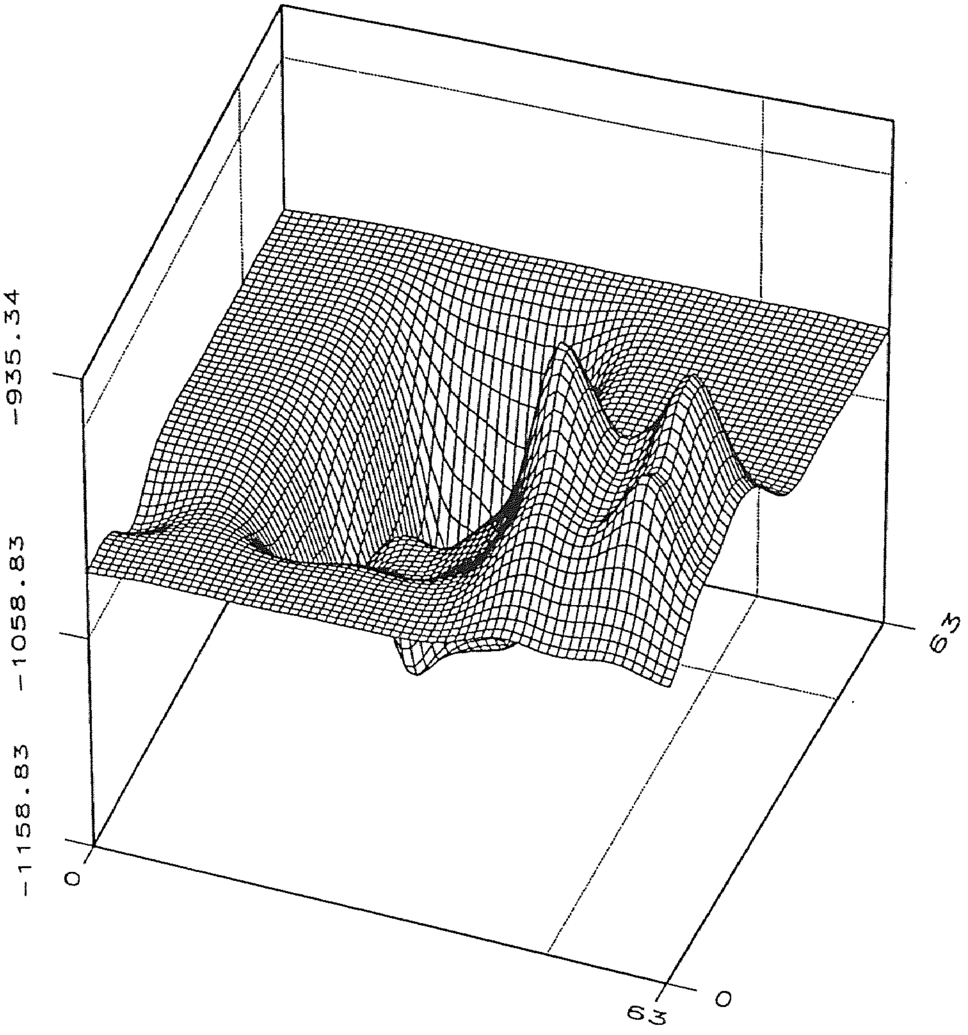


Figure 5.5 Depth map without Kalman filter.

Horn and Shunch's algorithm [9] is applied to the same images as shown in Figure 4.6-4.8, the curve of $RMSY_{error}$ versus iteration number k is shown at Figure 5.6. With the usage of the proposed Kalman filter and the following error decreasing rate,

$$\frac{\delta^2_{u,k+1}}{\delta^2_{u,k}} = \frac{1}{1.1(4\bar{I}^2_x\delta^2_{u,k}+1)} \tag{5.42}$$

results shown in Figure 5.6 indicate that the algorithm can improve boundary information that Horn and Shunch's algorithm can not extract well.

Existing computer vision algorithms using Kalman filter are all based on an incremental fashion. In those algorithms [17] [45], an image sequence is considered. In every iteration, different images are used. In this chapter, the Kalman filter is not used in the incremental fashion, i.e., the algorithm works on the same group of images with different iteration. In other words, information given by the a fixed image group is sufficiently used. On the other hand, image interpolation is not needed in our Kalman filter since the fixed image group is repeatedly used.

For implementation of Kalman filtering, the knowledge about noise models is important. However, it is impossible to exactly know the system noise η^u_k and measurement noise ξ^u_k in optical flow computation. In this chapter, some consideration has been given to these noises. We assume that $\xi^u_k = N(0, R^u_k)$, R^u_k satisfies Equation (5.30). Other effect of noise is covered by η^u_k that is $\eta^u_k = N(0, Q^u_k)$. Q^u_k is defined in Equation (5.19). The R^u_k comes from an analysis about the error decreasing rate.

As we know, $\dot{Z}(x,y)$ and $\dot{Y}(x,y)$ have not been improved. Our purpose in this chapter is to obtain a good method that can conserve discontinuity. Since $\dot{Z}$ and $\dot{Y}$ have very few discontinuity, improvement of them is not purpose of the chapter. In the next chapter, improvement of all 3-D parameter fields will be considered.

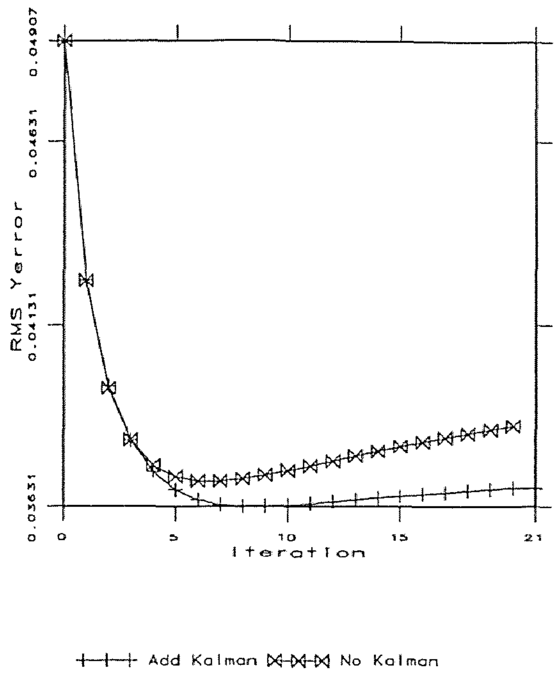


Figure 5.6 Convergence rate comparison.

CHAPTER 6

MOVING OBJECTS ESTIMATION USING KALMAN FILTER

In dealing with stereo imagery, the current techniques [23] [36] [37] [52] [53] classified as the multiple optical flow field approach [15] are, first, to find the optical flow field for each view and, then, to recover the motion and structure of the scene via the use of the correspondence between points in the pair of stereo images. Therefore, it is a combination of the optical flow approach and the point correspondence-approach. However, the feature extraction and correspondence are known difficult and only partial solutions suitable for simplistic situation have been developed [15]. A new theoretical framework: unified temporal-spatial optical flow field (UOFF) and its usage in motion analysis from stereo image sequences have been proposed in Chapter 4. The new concept of imaging space has been developed to include both temporal and spatial sequences of images. The optical flow determined for temporal sequences of images, say, by Horn and Schunck's algorithm [9], is extended to spatial sequences of images resulting in UOFF. Based on a four-frame rectangular model and the associated six UOFF field quantities, a set of equations is derived from which both position and velocity for each pixel can be determined. It does not require feature extraction and correspondence establishment. Using both temporal and spatial sequences of images, its capability of recovering motion exceeds significantly that of the existing motion stereo technique [10] [17] [18] [54] [55] and direct method [4] [6] [7] [8] [56] which can only recover the relative motion between a moving camera and a stationary environment. It can analyze multiple independently moving objects without the necessity of partition in the optical flow field as suggested in [57]. Besides depth map, it can estimate motion fields as well.

Obviously, UOFF approach to motion analysis faces all the problems faced by optical flow approach. One of the problems is the presence of various noises. In order to improve the results of motion analysis using UOFF approach, a correlation-feedback algorithm for determining optical flow [58] and a Kalman filter for further improving the accuracy in optical flow determination along moving boundaries, which will be referred to as the first Kalman filter in this chapter, have been developed in Chapter 3 and Chapter 5. To continue to improve the accuracy in motion analysis, stereo image sequences (sequences of the four-frame model) should be used instead of a single four frame model. It is noted that, to make use of previous knowledge and current states, Kalman filters have been utilized as an on-line estimator of depth map [17] and optical flow field [45], respectively, that are refined over time. The Kalman filters are robust with respect to both system and sensor noise.

In this chapter, a Kalman filter-based algorithm for 3-D estimation using UOFF has been proposed. When the Kalman filter is used, the object motion model needs to be known. However, in reality it is almost impossible to know object motion model exactly in advance. This modeling problem is discussed in this chapter. Significantly more consideration has been given to determine the covariance matrices, Q_k^D and R_k^D, for system noise and sensor noise, respectively than the previous work, say, reported in [17] and [45].

Most of previous works did not consider how to handle the so-called newly visible image areas, i.e., the disocclusion issue [17]. In fact, when multiple object movement is considered, the disocclusion is likely to occur. When the Kalman filter is used to fuse previous knowledge and current states, the disocclusion must be dealt with carefully since the previous knowledge may not reliable when disocclusion takes place. This is handled in our algorithm by using a threshold method. That is, whenever one of the components of state vector has changed more than 30% from its predicted value at the immediately previous iteration, we consider that disocclusion has taken place. (Obviously, occlusion would not satisfy this threshold and, hence, would be treated differently from disocclusion). We then set the covariance matrix of measurement noise, R_k, equal to zero. In other words, the estimated values are only related to the current measurements. This is reasonable since the previous knowledge is not reliable when disocclusion takes place.

Two experiments are presented to demonstrate the effectiveness of our algorithm. In both experiments, the Kalman filter is used. Together with the correlation-feedback technique in determining optical flow, good results have been achieved.

6.1 Modeling

A framework of Kalman filter is shown in Table 5.1 which consists of a system model, a measurement model, and prior models and is operated alternatively in a prediction phase and update phase to obtain an estimate that is unbiased and has a minimum mean-square error.

A block diagram of the proposed Kalman filter is shown in Figure 6.1. The subscript k stands for the kth iteration. There D_k is a state vector at the kth iteration: $(Z_k(x,y), \dot{Z}_k(x,y), \dot{X}_k(x,y), \dot{Y}_k(x,y))^T$, i.e., a vector having depth Z, speeds along Z, X, Y directions as its four components. D_k^u is optical flow vector estimated at the kth iteration by using the correlation-feedback algorithm and the first Kalman filter. O_k is a measurement vector formed by the four components: depth and speeds in 3-D world space. Φ is a transition matrix. K_k is a Kalman gain matrix. The measurement matrix H is here taken as an identity matrix I since the state vector and the measurement vector have the same four components.

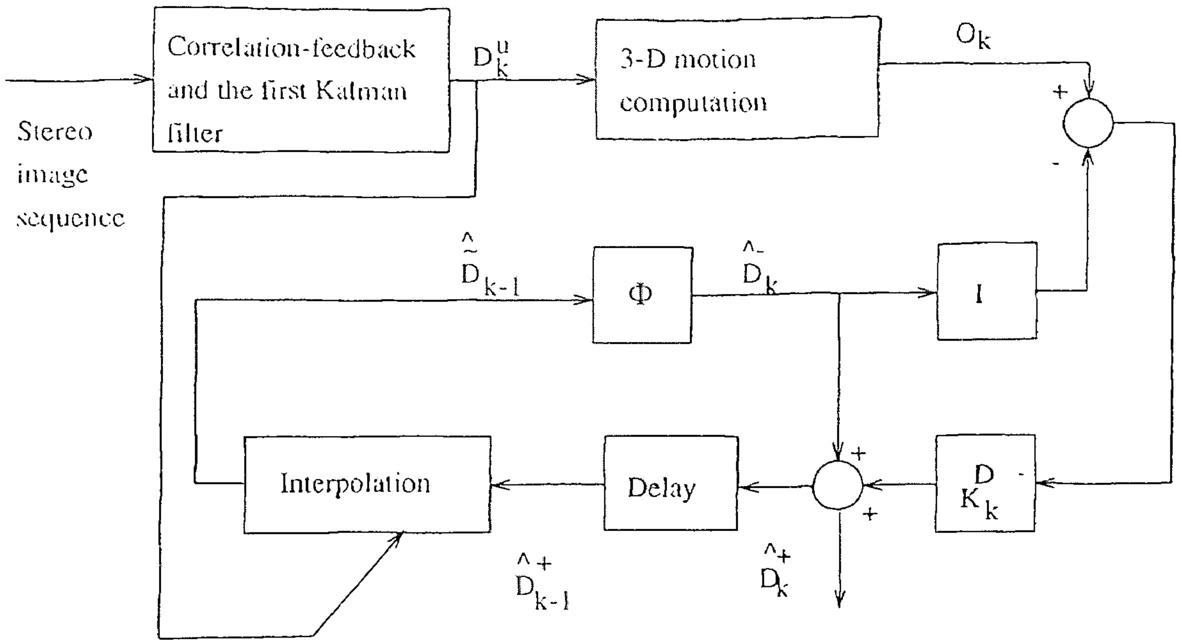


Figure 6.1 Framework.

When an object moves with a constant velocity, the system model can be represented as following:

$$
\begin{aligned}
D_k(x,y) &= \begin{bmatrix} Z_k(x,y) \\ \dot{Z}_k(x,y) \\ \dot{X}_k(x,y) \\ \dot{Y}_k(x,y) \end{bmatrix} \\
&= \begin{bmatrix} Z_{k-1}(x,y) + \dot{Z}_{k-1}(x,y) \\ \dot{Z}_{k-1}(x,y) \\ \dot{X}_{k-1}(x,y) \\ \dot{Y}_{k-1}(x,y) \end{bmatrix} \\
&= \Phi(D_{k-1}(x,y)), \qquad (6.1)
\end{aligned}
$$

where

$$
\Phi = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \qquad (6.2)
$$

Apparently, such a Φ means that the acceleration of any 3-D point is zero.

Taking a close look at the problem under consideration, one will find that the above equation is not absolutely correct for digital images. In digital images, (x, y) is a pixel. $D_k(x, y)$ is computed for the pixel (x, y) of the image at moment k. $D_{k-1}(x, y)$ is computed for the pixel (x, y) of the image at moment $k-1$. However, when the 3-D object(s) is moving, the pixel (x, y) of the digital images at different moments may be perspectively projected by different 3-D points. In order to continuously track a 3-D point, interpolation is needed. Therefore, Equation (6.1) should be rewritten as following:

$$\begin{aligned} D_k(x,y) &= \Phi(interpolation(D_{k-1}(x,y))) \\ &= \Phi(\tilde{D}_{k-1}(x,y)) \\ &= \Phi(\bar{D}_{k-1}(x,y) + \varphi_k(x,y)) \\ &= \Phi\bar{D}_{k-1}(x,y) + \eta_k^D(x,y) \end{aligned} \tag{6.3}$$

where $\tilde{D}_{k-1}(x,y) = interpolation(D_{k-1}(x,y))$, $\bar{D}_{k-1}$ is the certainty part of $\tilde{D}_{k-1}(x,y)$, $\varphi_k(x,y)$ is the uncertainty part of $\tilde{D}_{k-1}(x,y)$, and $\eta_k^D(x,y)$ is a product of Φ and $\varphi_k(x,y)$. $\eta_k^D(x,y)$ is mainly caused by the interpolation. We assume that $\eta_k^D(x,y) = N(0, Q_k^D(x,y))$. The more detailed discussion about $interpolation()$, $\eta_k(x,y)$ and $Q_k^D(x,y)$ will be made in Section 6.4.

Since $X_k(x,y)$ and $Y_k(x,y)$ are linearly dependent of $Z_k(x,y)$ according to the well-known perspective transform, they are not chosen as state variables. Therefore, the system model, measurement model and prior models are represented as following:

$$D_k(x,y) = \Phi\bar{D}_{k-1}(x,y) + \eta_k^D(x,y), \quad \eta_k^D(x,y) = N(0, Q_k^D(x,y)) \tag{6.4}$$

$$O_k(x,y) = H_k D_k(x,y) + \xi_k^D(x,y), \quad \xi_k^D(x,y) \sim N(0, R_k^D(x,y)) \tag{6.5}$$

$$H_k = I, \quad E\{(\xi_k^D(x,y))^T(\eta_k^D(x,y))\} = 0. \tag{6.6}$$

$O_k(x,y)$ is measured by using cameras, A/D converter, the correlation-feedback algorithm for optical flow estimation, the first Kalman filter, and a set of formulae for determining 3-D motion using UOFF in Chapter 4. $\xi_k^D(x,y)$ is the measurement noise that may be produced in the above mentioned measurement process.

Deriving the system model is important. If object motion is known, the transition matrix Φ should be modified and specified according to the known motion. If the motion is unknown, Equation (6.2) may produce error. In Section 6.6, we will discuss the problem in more detail.

6.2 Measurement Noise Covariance Matrix R^D (I)

System model has been set up, if the measurement and system noise covariance matrices R^D and Q^D are determined, Kalman filter can then be utilized. In this section, the determination of R^D is discussed. As mentioned in the last section, the measurement noise $\xi_k^D(x,y)$ is caused by cameras, A/D converter, optical flow determination and 3-D motion estimation. $\xi_k^D(x,y)$ is assumed to be a Gaussian white noise. Therefore, R_k^D can be computed from measurement noises of u^s, u^L, u^R, and v^L, denoted by ξ^u in Chapter 5. It is noted that the u^s, u^L, u^R, and v^L are the UOFF quantities, which are the state variables of the first Kalman filter. In the following discussion, a superscript u is sometimes used to denote quantities defined and used for the first Kalman filter in Chapter 5. Hence, as defined in the last chapter, D^u, U and $\xi^u = N(0, R^u)$ are, respectively, a measurement vector field, a state variable vector field, and the measurement noise in the first Kalman filter. It is also noted that here we do not consider the part of measurement noise caused by applying the set of equations (Equations (4.29-4.36)) in estimating 3-D motion using the UOFF approach, i.e., Equations (4.29-4.36).

From Equation (5.17), one has

$$D^u = U + \xi^u,$$

$$D^u - U = \xi^u,$$

and

$$\begin{bmatrix} d_1^u - u^s \\ d_2^u - u^L \\ d_3^u - u^R \\ d_4^u - v^L \end{bmatrix} = \begin{bmatrix} \xi_1^u \\ \xi_2^u \\ \xi_3^u \\ \xi_4^u \end{bmatrix} \tag{6.7}$$

According to Kalman filter theory, the estimates of D^u and U in the first Kalman filter have the following relationship:

$$\begin{bmatrix} \hat{d}_1^u \\ \hat{d}_2^u \\ \hat{d}_3^u \\ \hat{d}_4^u \end{bmatrix} = \begin{bmatrix} \hat{u}^s \\ \hat{u}^L \\ \hat{u}^R \\ \hat{v}^L \end{bmatrix}. \tag{6.8}$$

From Equations (6.5) and (6.6) of the last section, it is known that

$$O_k(x,y) = D_k(x,y) + \xi_k^D(x,y). \tag{6.9}$$

i.e.,

$$\begin{bmatrix} o1_k(x,y) \\ o2_k(x,y) \\ o3_k(x,y) \\ o4_k(x,y) \end{bmatrix} = \begin{bmatrix} Z_k(x,y) \\ \dot{Z}_k(x,y) \\ \dot{X}_k(x,y) \\ \dot{Y}_k(x,y) \end{bmatrix} + \begin{bmatrix} \xi_1^D \\ \xi_2^D \\ \xi_3^D \\ \xi_4^D \end{bmatrix}$$

From Equations (4.29-4.36) in the discussion of UOFF approach, refer to Chapter 4, it is easy to see that $o1(x,y)$, $o2(x,y)$, $o3(x,y)$, and $o4(x,y)$ are the functions of d_1^u, d_2^u, d_3^u, and d_4^u.

We can assume $D_k(x,y) = O_k \mid_{D_k^u = U_k}$ and use the Taylor series expansion to obtain

$$o1 = Z + \frac{d(o1)}{d(d_1^u)} \mid_{d_1^u = u^s} (d_1^u - u^s) + \theta_Z, \tag{6.10}$$

$$\begin{aligned} o2 &= \dot{Z} + \frac{d(o2)}{d(d_1^u)} \mid_{d_1^u = u^s} (d_1^u - u^s) \\ &+ \frac{d(o2)}{d(d_2^u)} \mid_{d_2^u = u^L} (d_2^u - u^L) \\ &+ \frac{d(o2)}{d(d_3^u)} \mid_{d_3^u = u^R} (d_3^u - u^R) \\ &+ \theta_{\dot{Z}}, \end{aligned} \tag{6.11}$$

$$\begin{aligned} o3 &= \dot{X} + \frac{d(o3)}{d(d_1^u)} \Big|_{d_1^u = u^s} (d_1^u - u^s) \\ &+ \frac{d(o3)}{d(d_2^u)} \Big|_{d_2^u = u^L} (d_2^u - u^L) \\ &+ \frac{d(o3)}{d(d_3^u)} \Big|_{d_3^u = u^R} (d_3^u - u^R) \\ &+ \theta_{\dot{X}}, \end{aligned} \tag{6.12}$$

and

$$\begin{aligned} o4 &= \dot{Y} + \frac{d(o4)}{d(d_1^u)} \Big|_{d_1^u = u^s} (d_1^u - u^s) \\ &+ \frac{d(o4)}{d(d_2^u)} \Big|_{d_2^u = u^L} (d_2^u - u^L) \\ &+ \frac{d(o4)}{d(d_3^u)} \Big|_{d_3^u = u^R} (d_3^u - u^R) \\ &+ \frac{d(o4)}{d(d_4^u)} \Big|_{d_4^u = v^L} (d_4^u - v^L) \\ &+ \theta_{\dot{Y}}, \end{aligned} \tag{6.13}$$

where θ_Z, $\theta_{\dot{Z}}$, $\theta_{\dot{X}}$, and $\theta_{\dot{Y}}$ represent the second order and above terms. They can be ignored. One can have

$$\begin{bmatrix} o1 - Z \\ o2 - \dot{Z} \\ o3 - \dot{X} \\ o4 - \dot{Y} \end{bmatrix} = \begin{bmatrix} \frac{d(o1)}{d(d_1^u)} \big|_{d_1^u = u^s} & 0 & 0 & 0 \\ \frac{d(o2)}{d(d_1^u)} \big|_{d_1^u = u^s} & \frac{d(o2)}{d(d_2^u)} \big|_{d_2^u = u^L} & \frac{d(o2}{d(d_3^u)} \big|_{d_3^u = u^R} & 0 \\ \frac{d(o3)}{d(d_1^u)} \big|_{d_1^u = u^s} & \frac{d(o3)}{d(d_2^u)} \big|_{d_2^u = u^L} & \frac{d(o3)}{d(d_3^u)} \big|_{d_3^u = u^R} & 0 \\ \frac{d(o4)}{d(d_1^u)} \big|_{d_1^u = u^s} & \frac{d(o4)}{d(d_2^u)} \big|_{d_2^u = u^L} & \frac{d(o4)}{d(d_3^u)} \big|_{d_3^u = u^R} & \frac{d(o4)}{d(d_4^u)} \big|_{d_4^u = v^L} \end{bmatrix} \begin{bmatrix} d_1^u - u^s \\ d_2^u - u^L \\ d_3^u - u^R \\ d_4^u - u^v \end{bmatrix}. \tag{6.14}$$

That is,

$$O_k - D_k = G_k \xi_k^u. \tag{6.15}$$

Comparing Equation (6.9) with Equation (6.15), one has

$$\xi_k^D = G_k \xi_k^u. \tag{6.16}$$

Here, $\xi_k^D = N(0, R_k^D)$ and $\xi^u = N(0, R^u)$, therefore

$$R^D = covar\{G_k \xi_k^u\} = G_k covar\{\xi_k^u\} G_k^T = G_k R^u G_k^T. \tag{6.17}$$

where R^u is the covariance matrix for the measurement noise in optical flow determination used in the first Kalman filter. Thus, R_k^D can obtained. But, since th newly visible area exposure problem needs to be handled, this matrix must be modified accordingly. A discussion in this regard is conducted next.

6.3 Measurement Noise Covariance Matrix R^D (II): Newly Visible Image Areas

As pointed out in Section 6.1, when 3-D objects move, newly visible areas become exposed on images. Thus, detecting and properly handling the newly visible area, i.e., disocclusion, is important. These two questions will be addressed in this section.

Obviously, all previous knowledge becomes unreliable in these newly visible areas. Only current measurement is reliable. This situation amounts to that the Kalman filter-based system works at initial conditions in these newly visible areas. This can be justified as follows.

According to Kalman filter theory, when the measurement noise covariance matrix is equal to zero, i.e., $R_k^D = 0$, the Kalman gain matrix K_k^D will be

$$\begin{aligned} K_k^D &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k^D)^{-1} \\ &= P_k^- H_k^T (H_k P_k^- H_k^T)^{-1} \\ &= H_k^{-1} H_k P_k^- H_k^T (H_k P_k^- H_k^T)^{-1} \\ &= H_k^{-1}, \end{aligned} \tag{6.18}$$

and the updated state covariance matrix P_k^+ will be

$$P_k^+ = (I - K_k^D H_k) P_k^- = 0, \tag{6.19}$$

and the updated state variable vector will be

$$\begin{aligned} \hat{D}_k^+ &= \hat{D}_k^- + K_k^D (O_k^- - H_k \hat{D}_k^-) \\ &= \hat{D}_k^- + H_k^{-1} (O_k^- - H_k \hat{D}_k^-) \\ &= H_k^{-1} O_k. \end{aligned} \tag{6.20}$$

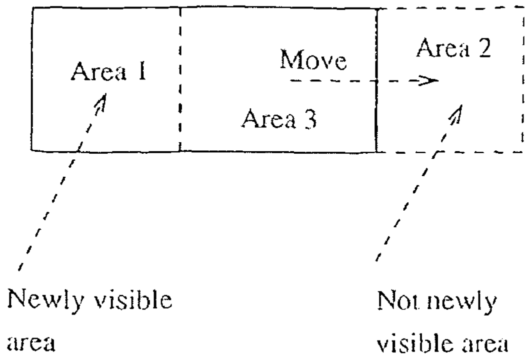


Figure 6.2 Disocclusion.

The above result means that estimated state variable vector are associated only with the current measurement vector and the estimated values are unbiased. Or, equivalently, in the newly exposed areas, the Kalman filter amounts to work on the current information only, as if, in this sense, work at initial state. Its initial state covariance matrix P_k^+ is zero. Hence, the conclusion is that as long as the measurement covariance matrix R_k^D is set to 0 in the newly visible areas, Kalman filter can properly handle the newly visible areas automatically.

Then, how to detect the newly visible image areas?

In Figure 6.2, the entire block surrounded by solid lines moves into the place surrounded by dash lines, it is evident that Area 1 is a newly visible area but Area 2 is not. This is because Area 3 fills Area 2, Area 1 moves into Area 3, but, there is no any area that can shift into Area 1. Area 1 is a newly visible area.

If the block surrounded by the solid lines is used to represent the image of a 3-D object at moment $k-1$, the block surrounded by the dash lines is used to represent the image of this object at moment k, and the image moving velocity is known as $(u^L(x_k, y_k), v^L(x_k, y_k))$, then $D_k(x_k, y_k)$ is the current state variable vector and $D_{k-1}(x_{k-1}, y_{k-1})$ is the state variable vector at moment $k-1$.

From Section 6.1, one knows

$$D_k(x,y) = \Phi(interpolation(D_{k-1}(x,y))). \tag{6.21}$$

From Equation (6.9), one has

$$D_k(x,y) = O_k(x,y) - \xi_k^D(x,y). \tag{6.22}$$

However, in Area 1, Equation (6.21) no longer holds because the image there at moment k is not related to the 3-D object any more. Hence $D_k(x_k, y_k)$ no longer has any relationship with $D_{k-1}(x_{k-1}, y_{k-1})$ in the area. $D_k(x_k, y_k)$ can only be obtained from current measurement vector $O_k(x_k, y_k)$ in Area 1, refer to Equation (6.22).

In general, if the system based on the Kalman filter shown in Figure 6.1 is used, the estimate $\hat{D}_k(x_k, y_k)$ can be obtained from $\Phi(interpolation(\hat{D}_{k-1}(x,y))$. Thus,

$$E\{O_k(x,y) - \hat{D}_k^-(x,y)\} = E\{D_k(x,y) - \hat{D}_k^-(x,y) + \xi_k^D(x,y)\}.$$

Since ξ_k^D is independent of $D_k(x,y) - \hat{D}_k^-(x,y)$ and $E\{\xi_k^D\} = 0$,

$$E\{O_k(x,y) - \hat{D}_k^-(x,y)\} = E\{D_k(x,y) - \hat{D}_k^-(x,y)\}.$$

Hence except Area 1, $E\{D_k(x,y) - \hat{D}_k^-(x,y)\} \simeq 0$. It is evident that $|E\{O_k(x,y) - \hat{D}_k^-(x,y)\}|$ is much greater in Area 1 than that in any other areas. This observation can be used to detect disocclusion.

In our experiments, $|O_k(x,y) - \hat{D}_k^-(x,y)|$ instead of $|E\{O_k(x,y) - \hat{D}_k^-(x,y)\}|$ is used. This is because the local average computation may fail the detection of newly visible areas in the discontinuity areas, since an average computation can blur a boundary, while the newly visible areas are usually related to discontinuities.

Since $|O_k(x,y) - \hat{D}_k^-(x,y)| = |D_k(x,y) - \hat{D}_k^-(x,y) + \xi_k^D(x,y)|$, the accuracy of the detection of disocclusion is mainly depends on the measurement noise ξ_k^D. Due to the uses of the correlation-feedback algorithm and the first Kalman filter, the amplitude of $\xi_k^D(x,y)$ are expected to be small. Therefore, detecting disocclusions is not difficult. Threshold methods can be used. In our experiments, following thresholds are applied.

$$\frac{|o1_k(x,y) - \hat{Z}_k^-(x,y)|}{|D_{is} - \hat{Z}_k^-(x,y)|} \leq 30\% \tag{6.23}$$

$$\frac{|o2_k(x,y) - \hat{Z}_k^-(x,y)|}{|\hat{Z}_k^-(x,y)|} \leq 30\% \tag{6.24}$$

$$\frac{|o3_k(x,y) - \hat{X}_k^-(x,y)|}{|\hat{X}_k^-(x,y)|} \leq 30\% \tag{6.25}$$

$$\frac{|o4_k(x,y) - \hat{Y}_k^-(x,y)|}{|\hat{Y}_k^-(x,y)|} \leq 30\% \tag{6.26}$$

where D_{is} is the distance between the fixation point and a camera, and $D_{is} = l \cdot cotan(\theta)$, l and θ are shown in Figure 6.3. When any one of the above conditions is not satisfied, one concludes that disocclusions happen at the pixel (x, y). Thus, at the pixel, set $R_k^D(x,y) = 0$.

The accuracy of detecting disocclusion is partially determined by the accuracy of detecting the moving boundaries of optical flow fields. The moving boundaries always exist around newly exposed areas. If the moving boundaries are blurred, the edges around the newly exposed areas can not be detected effectively by using the thresholds. In other words, the accuracy improvement along moving boundaries discussed in the last chapter is quite useful for detecting the newly exposed areas.

6.4 System Noise Covariance Matrix Q

The derivation of measurement noise covariance matrix R_k^D has been conducted in the previous two sections. Here we will determine system noise covariance matrix Q_k^D.

System model of standard Kalman filter is a linear function. But, here as we discussed in Section 6.1, the system model is not linear, i.e.,

$$D_k = \Phi(interpolation(D_{k-1})).$$

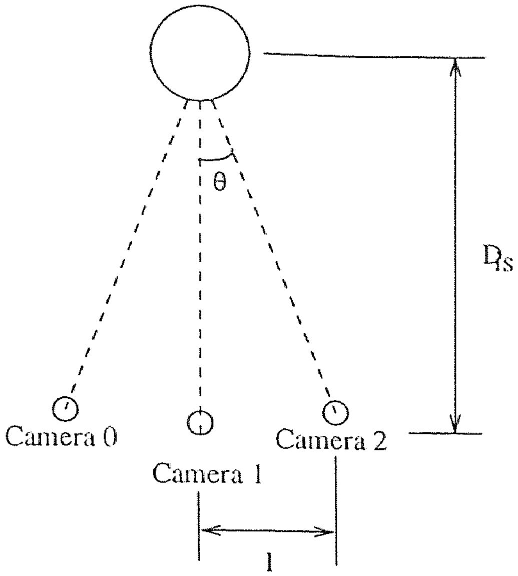


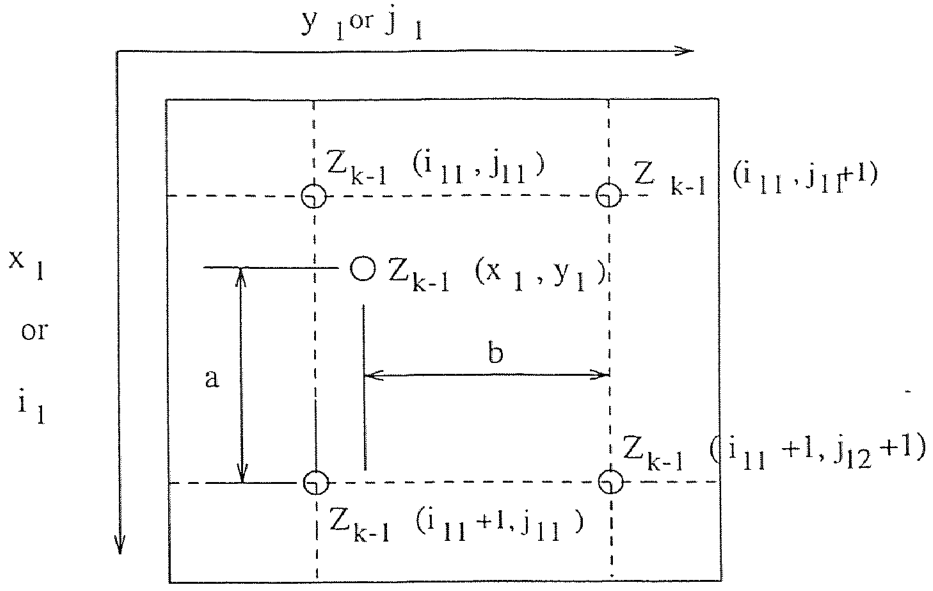
Figure 6.3 Camera model

Therefore, the state covariance extrapolation $P_k^- = \Phi P_{k-1}^+ \Phi^T + Q_{k-1}^D$ shown in Table 5.1 does not hold and needs to derive in this section.

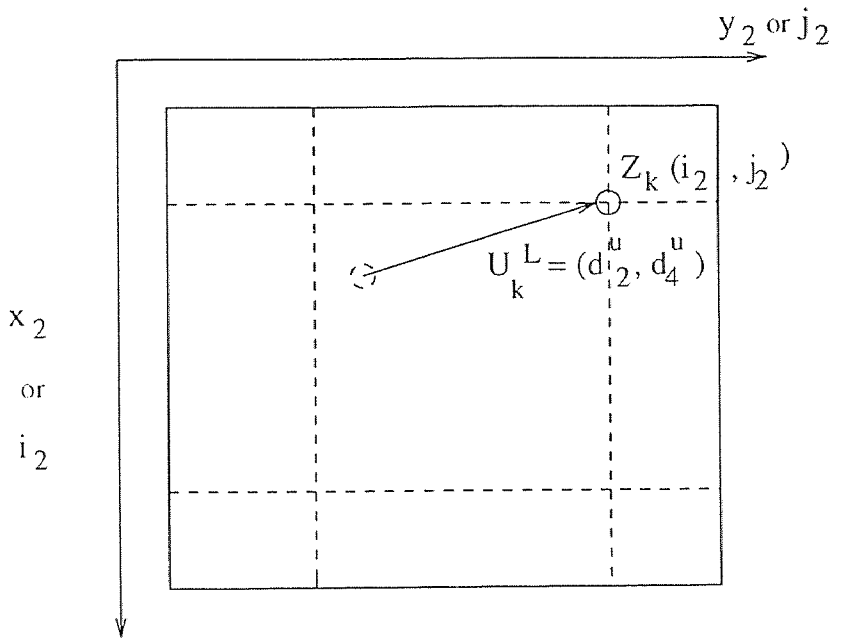
Equation (6.3) in Section 6.1 is rewritten below.

$$\begin{aligned} D_k(x,y) &= \Phi(interpolation(D_{k-1}(x,y))) \\ &= \Phi(\tilde{D}_{k-1}(x,y)) \\ &= \Phi(\bar{D}_{k-1}(x,y) + \varphi_k(x,y)) \\ &= \Phi\bar{D}_{k-1} + \eta_k^D(x,y) \qquad (6.27) \end{aligned}$$

where $\eta_k(x,y) \sim N(0, Q_k^D(x,y))$. The function, *interpolation*(), will affect computation of the state covariance P_k^- and noise covariance Q_k^D in the prediction phase. Furthermore, we assume that $\eta_k(x,y)$ mainly comes from the function, *interpolation*(). The so-called interpolation can be explained by referring to Figure 6.4. What we did here is a little bit different from what discussed in [17]. There the authors suggested that the depth field at the $(k-1)$th moment be first extrapolated (shifting) to form that at the kth moment according to optical flow vector, an



Depth map (Z_{k-1} (i_1 , j_1))



Depth map (Z_k (i_2 , j_2))

Figure 6.4 Interpolation

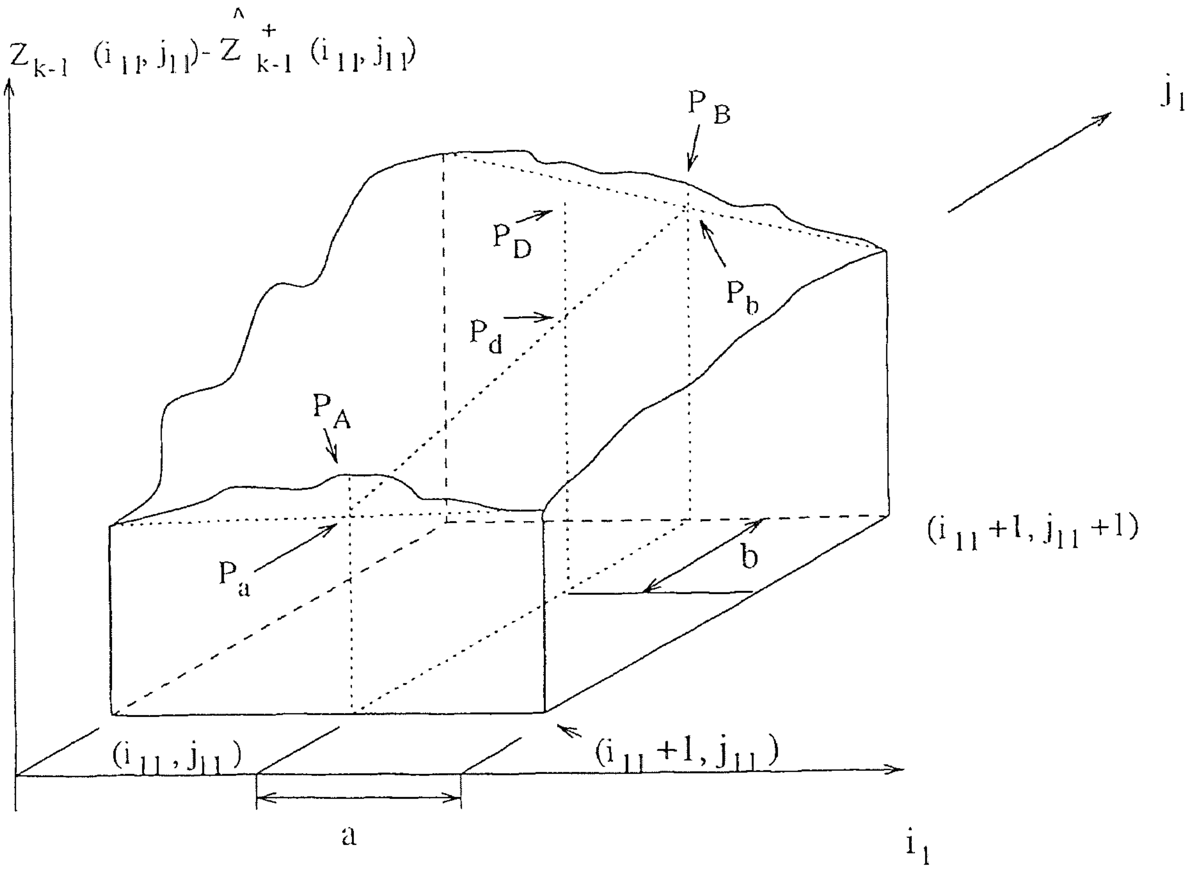


Figure 6.5 The reason why noises are produced by bilinear interpolation.

interpolation (resampling) be then conducted. We found that it is more practical to first interpolate the depth field (in our case, the motion fields as well) at the $(k-1)$th moment, then shift the interpolated depth to form that at the kth moment by using optical flow vector. This is clearly described in Figure 6.4.

A bilinear interpolation is utilized which is illustrated in Figure 6.5. There certain errors are generated. From Figure 6.5, we have

$$\begin{aligned} P_a &= a[Z_{k-1}(i_{11}, j_{11}) - \hat{Z}^{+}_{k-1}(i_{11}, j_{11})] + (1-a)[Z_{k-1}(i_{11}+1, j_{11}) \\ &\quad - \hat{Z}^{+}_{k-1}(i_{11}+1, j_{11})] \end{aligned} \tag{6.28}$$

and

$$P_A = P_a + w_1. \tag{6.29}$$

We assume $w_1 = N(0, \delta_{w_1})$.

Similarly, we have

$$\begin{aligned} P_b &= a[Z_{k-1}(i_{11}, j_{11}+1) - \hat{Z}^+_{k-1}(i_{11}, j_{11}+1)] + (1-a)[Z_{k-1}(i_{11}+1, j_{11}+1) \\ &\quad -\hat{Z}^+_{k-1}(i_{11}+1, j_{11}+1)] \end{aligned} \tag{6.30}$$

and

$$P_B = P_b + w_2, \quad w_2 = N(0, \delta_{w_2}). \tag{6.31}$$

Furthermore,

$$P_d = bP_a + (1-b)P_b, \tag{6.32}$$

and we assume

$$P_D = P_d + w_3, \quad w_3 = N(0, \delta_{w_3}). \tag{6.33}$$

According to the last chapter, we assume that d_2^{ua} and d_4^{ua} are the true values of d_2^u and d_4^u, respectively, and

$$d_2^u = d_2^{ua} + \mu_x,$$

$$d_4^u = d_4^{ua} + \mu_y,$$

where $\mu_x = N(0, \delta_x)$ and $\mu_y = N(0, \delta_y)$.

We define

$$\sigma_k^Z(i_2, j_2) = E\{(\varphi_{k-1}^Z(i_2, j_2))^2\}, \tag{6.34}$$

where the superscript Z denotes the first component in the measurement vector, i.e., the depth. The derivation in Appendix C shows

$$\begin{aligned} &\sigma_k^Z(i_2, j_2) \\ = \ &[\bar{b}((Z_{k-1}(i_{11}, j_{11}) - \hat{Z}^+_{k-1}(i_{11}, j_{11})) - (Z_{k-1}(i_{11}+1, j_{11}) - \hat{Z}^+_{k-1}(i_{11}+1, j_{11}))) \end{aligned}$$

$$\begin{aligned}
&+(1-\bar{b})((Z_{k-1}(i_{11},j_{11}+1)-\hat{Z}^{+}_{k-1}(i_{11},j_{11}+1))\\
&-(Z_{k-1}(i_{11}+1,j_{11}+1)-\hat{Z}^{+}_{k-1}(i_{11}+1,j_{11}+1)))]^2\delta_x^2\\
&+[\bar{a}((Z_{k-1}(i_{11},j_{11})-\hat{Z}^{+}_{k-1}(i_{11},j_{11}))-(Z_{k-1}(i_{11},j_{11}+1)-\hat{Z}^{+}_{k-1}(i_{11},j_{11}+1)))+\\
&(1-\bar{a})((Z_{k-1}(i_{11}+1,j_{11})-\hat{Z}^{+}_{k-1}(i_{11}+1,j_{11}))\\
&-(Z_{k-1}(i_{11}+1,j_{11}+1)-\hat{Z}^{+}_{k-1}(i_{11}+1,j_{11}+1)))]^2\delta_y^2\\
&+\bar{b}^2\delta_{w1}^2+(1-\bar{b})^2\delta_{w2}^2+\delta_{w3}^2. \qquad (6.35)
\end{aligned}$$

It is noted that $\sigma_k^{\dot{Z}}, \sigma_k^{\dot{X}}, \sigma_k^{\dot{Y}}$ can be expressed in the same way.

In our experiments, the linearization noises are represented as

$$\delta_{w1}^2 \approx (\frac{\hat{Z}_{k-1}(i_{11},j_{11})-\hat{Z}_{k-1}(i_{11}+1,j_{11})}{2})^2. \qquad (6.36)$$

$$\delta_{w2}^2 \approx (\frac{\hat{Z}_{k-1}(i_{11},j_{11}+1)-\hat{Z}_{k-1}(i_{11}+1,j_{11}+1)}{2})^2. \qquad (6.37)$$

$$\delta_{w3}^2 \approx ((\frac{\hat{Z}_{k-1}(i_{11},j_{11})-\hat{Z}_{k-1}(i_{11},j_{11}+1)}{4})+(\frac{\hat{Z}_{k-1}(i_{11}+1,j_{11})-\hat{Z}_{k-1}(i_{11}+1,j_{11}+1)}{4}))^2. \qquad (6.38)$$

The δ_x^2 and δ_y^2 are variances of d_2^u and d_4^u that can be obtained from Equation (5.21) in the last chapter, i.e.,

$$\begin{aligned}
\delta_{x,k}^2 &= \delta_{l,k}^2\\
\delta_{y,k}^2 &= \delta_{v,k}^2 \qquad (6.39)
\end{aligned}$$

where k being the iteration index.

The Appendix C further shows that the system noise covariance matrix is

$$Q_k^D(i_2,j_2)=\Phi\begin{pmatrix}\sigma_k^Z(i_2,j_2) & 0 & 0 & 0\\ 0 & \sigma_k^{\dot{Z}}(i_2,j_2) & 0 & 0\\ 0 & 0 & \sigma_k^{\dot{X}}(i_2,j_2) & 0\\ 0 & 0 & 0 & \sigma_k^{\dot{X}}(i_2,j_2)\end{pmatrix}\Phi^T. \qquad (6.40)$$

and the predicted state covariance matrix at the kth iteration, P_k^-, is as follows.

$$P_k^- = \Phi[interpolation^2(P_{k-1}^+)]\Phi^T + Q_k^D \qquad (6.41)$$

It is noted that the difference between *interpolation* and the notation of *interpolation*^{2} is illustrated in Equations (C.30) and (C.31), refer to Appendix C.

For the completeness, we also write the prior model below.

$$E\{D_0\} = \hat{D}_0 \tag{6.42}$$

$$cov\{D_0\} = P_0 = (D_0 - E\{D_0\})(D_0 - E\{D_0\})^T. \tag{6.43}$$

$E\{D_0(x,y)\}$ can be substituted by the average around $D_0(x,y)$, i.e.,

$$\sum_{i=-1}^{1} \sum_{j=-1}^{1} D_0(x+i, y+j)/9.$$

The update phase the Kalman filter-based system is:

$$\hat{D}_k^+(x,y) = \hat{D}_k^- + K_k^D(O_k - I\hat{D}_k^-) \tag{6.44}$$

$$K_k^D(x,y) = P_k^-(P_k^- + R_k^D)^{-1} \tag{6.45}$$

$$P_k^+(x,y) = (I - K_k^D)P_k^- \tag{6.46}$$

Now the proposed Kalman filter has been completely specified.

6.5 Experiments

Two experiments using the proposed Kalman filter-based algorithm are presented here. The first experiment is about three objects. There is a relative motion between them. UOFF approach together with the feedback technique in determining optical flow and the proposed Kalman filter are applied. Fairly good results about depth and motion fields illustrate the feasibility of our approach.

In the second experiment the entire scene is a flat poster experiencing translation. Again both depth and motion field are recovered by our approach. A comparison between the depth map obtained by our approach and that reported in [17] is made. It turns out that our method performs better: more accurate and more robust against noise.

6.5.1 Experiment I

Setting: The experiment setting is shown in Figure 4.5 where the superscript l stands for left, r for right and m for middle. The angles between two neighboring optical axes is 2.5°. The focal length is 12.5 mm. Flat posters are mounted on three boxes facing the camera. The box 1 is fixed, while the box 2 and box 3 move together horizontally. The translation velocity is 5 mm/frame. Other dimensions are shown in Figure 4.5. Images are taken with a CCD SONY camera via a DATACUBE system and stored in a SUN SPARC workstation. The images are 512×512 and cut into 256×256. Through a subsampling, images are further reduced to 64×64. Three of the reduced images at the first moment are shown in Figures 4.6-4.8.

In other words, the first experiment setting is the same as the experiment setting in the last chapter except that the combination of box 2 and box 3 now continues to move at a speed of 5 mm/frame. That is, a stereo image sequence is used. The arrangement of stereo image sequence is shown in Figure 6.6.

Results: The image sequence is processed by using the UOFF approach, the correlation-feedback algorithm and the first Kalman filter, and the Kalman filter proposed in this chapter, refer to Figure 6.1. The Kalman filter iterates 11 times and during each iteration the correlation feedback algorithm iterates 10 times. The results of the experiment are shown in Table 6.1. There the RMS errors and average errors are defined the same as in [15].

Comparing the first column of Table 6.1 with last column of the table, $\dot{Z}(x, y)$ and $\dot{Y}(x, y)$ are indeed improved. In the middle column, the two Kalman filters are used but the technique of detecting newly visible image areas is not performed. It is worth noting that if newly visible areas are considered, the error can be reduced.

Discussion: Though this experiment is not a complicated one, however, the scene contains multiple objects that have a relative motion between box 1, and the combination of box 2 and box 3. That is, the scene is not stationary. If the camera is in

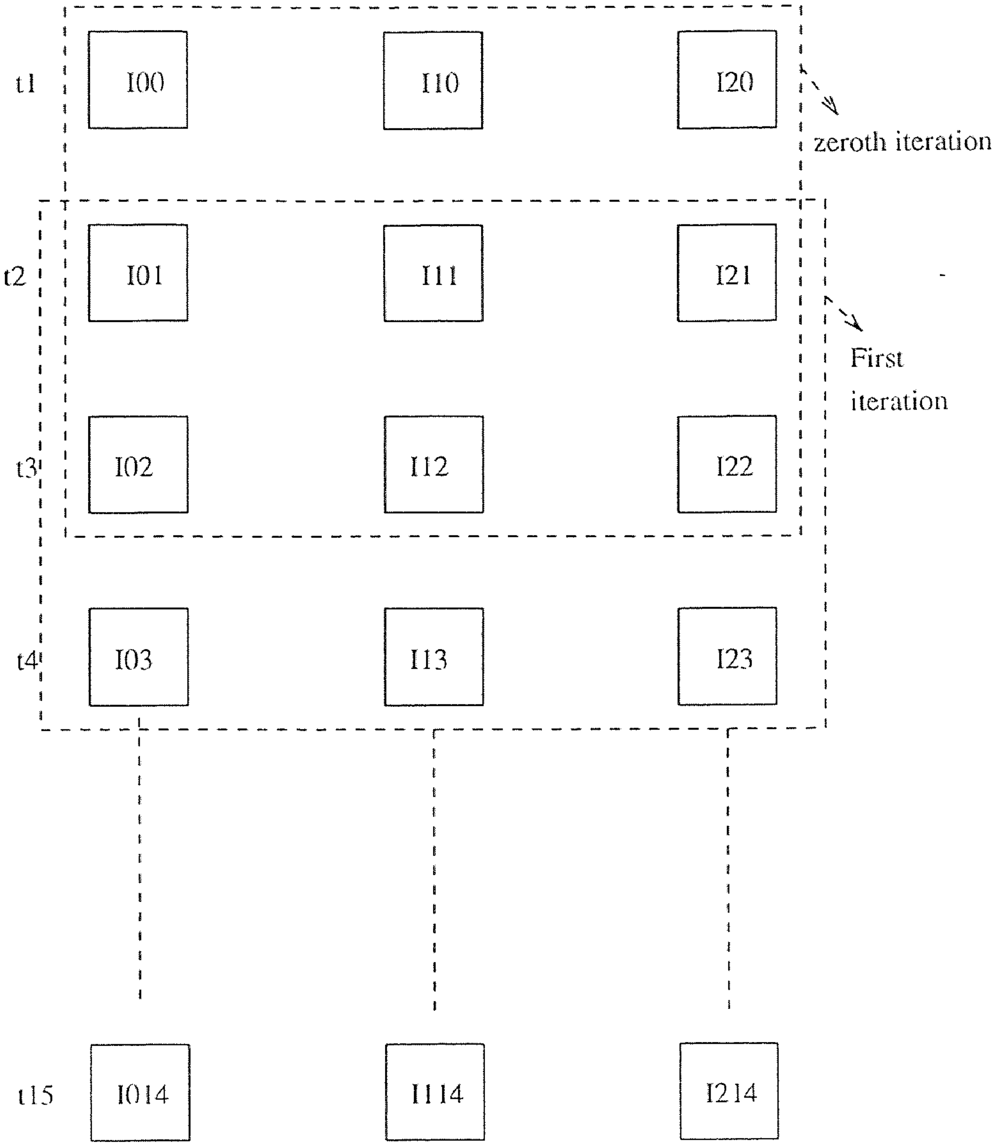


Figure 6.6 Stereo image sequence

Table 6.1 Result comparison

Error Type	First Kalman Filter	No Detecting New Areas	Two Kalman Filters
RMS Z_{error}	3.56%	3.53%	3.47%
RMS X_{error}	3.50%	3.48%	3.37%
RMS Y_{error}	3.58%	3.60%	3.53%
Average Z_{error}	15.15%	11.75%	12.69%
Average X_{error}	2.95%	3.07%	2.89%
Average Y_{error}	0.87%	0.62%	0.58%

another movement, then this case cannot be handled in the conventional framework of motion stereo or direct method as discussed before. With our UOFF approach, both depth map and motion fields may be recovered. The correlation-feedback algorithm for optical flow determination and the Kalman filter for incrementally image sequence processing have assisted us to achieve fairly good accuracy. More complicated experiments containing multiple independently moving objects are planned to be conducted.

6.5.2 Experiment II

Setting: It has been mentioned that the previous works in motion analysis mainly considered only camera motion. Furthermore a planar scene is often used to verify efficiency of an algorithm [17]. In order to compare the accuracy achieved by our approach and that by others, the second experiment has been conducted and is reported here. In the experiment, the entire scene is a flat poster, that horizontally moves with a speed of 4 mm/frame. The distance between the camera and the plane in which the poster is located and moving is 885 mm. The focal length is 12.5 mm. In order to test robustness of our approach under the circumstance that image quality is low, only 7 bits are used for quantization levels in the experiment. Hence, the

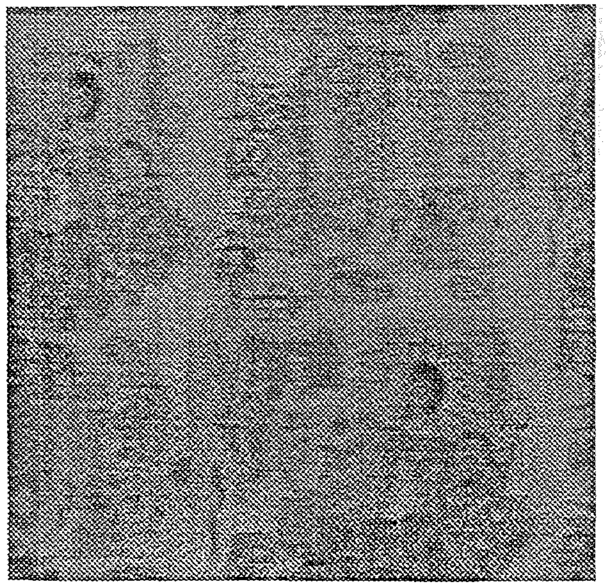


Figure 6.7 First frame of plane motion.

1st and 12th frames taken by the middle camera, shown in Figures 6.7 and 6.8, are relatively dark.

The second Kalman filter iterates 10 times. In each iteration, the correlation-feedback algorithm iterates twice and the first Kalman filter iterates once. In this experiment, the first Kalman filter is useless since there is no any moving boundary. Hence, $T^s = T^l = T^r = T^v = 0$ is applied. In fact, the first Kalman filter can be ignored in such experiment.

In the experiment, the transition matrix Φ is an identity matrix.

Results: By applying our UOFF approach, including the correlation-feedback algorithm and Kalman filter, we obtained good results which are shown in Figures 6.9-6.11. It is noted that the velocities along Y and Z directions, $\dot{Y}$ and $\dot{Z}$, are expected to be zero in this experiment.

Discussion: In [17] Matthies, Kanade and Szeliski reported a quantitative experiment. There a poster of a tiger is fixed. The image sequence was taken

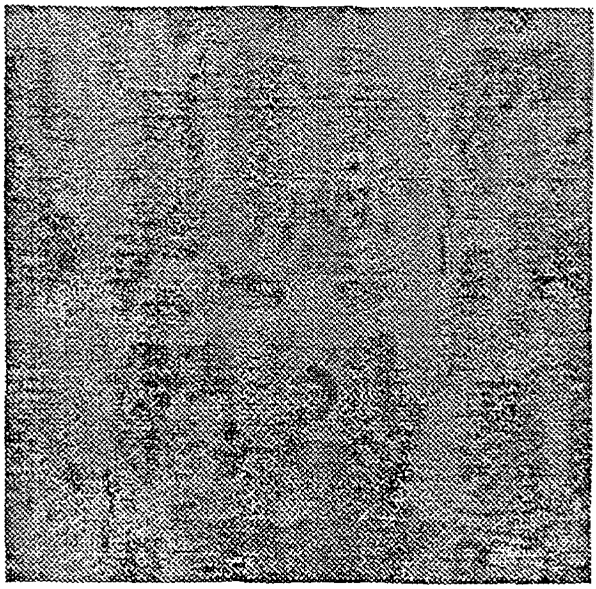


Figure 6.8 Twelfth frame of plane motion.

with vertical camera motion. The authors believed that it is better than horizontal motion. They used a correlation-based algorithm to estimate optical flow for each pixel. The depth map was then recovered from the estimated optical flow and known camera motion with respect to the whole poster: the depth-from-motion, one of the typical motion stereo techniques. A Kalman filter was used. In their experiment, from the 1st to the 10th iterations of the Kalman filter, the RMS relative error in the depth Z decreases from 7% to a little bit more than 2%, refer to the so-called "Actual dense iconic" curve in Figure 10 in [17]. In our approach, from the 1st to 10th iterations of the proposed Kalman filter, the RMS relative error in Z deceases from 3% to 1.71%. Considering 7 bit in quantization, this means our results are more accurate and robust. Furthermore, not only the depth map but also the speed fields: $\dot{X}, \dot{Y}, \dot{Z}$ are recovered.

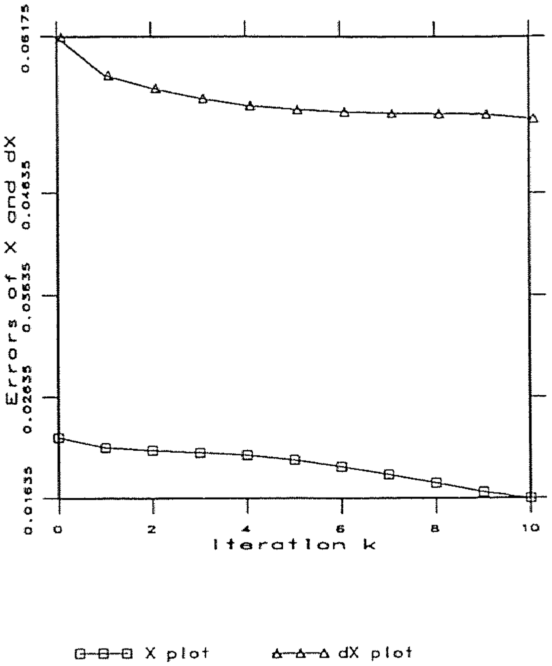


Figure 6.9 RMS errors of X and Average errors of $\dot{X}$.

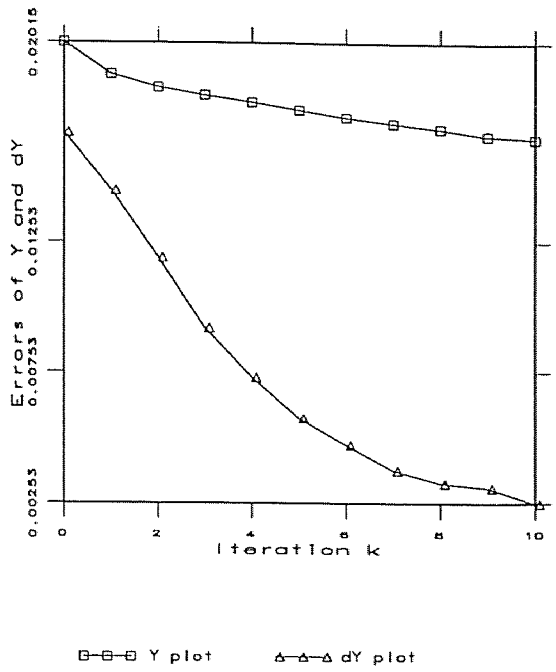


Figure 6.10 RMS errors of Y and Average errors of $\dot{Y}$.

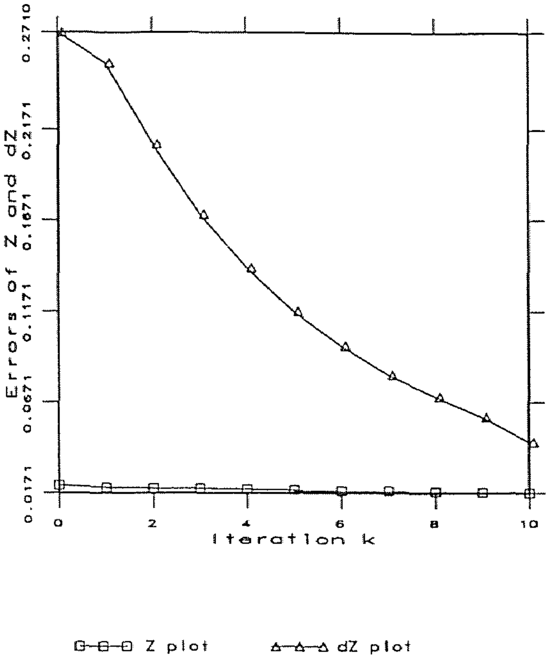


Figure 6.11 RMS errors of Z and Average errors of $\dot{Z}$.

6.6 Discussions and Conclusions

6.6.1 System Model

In the first experiment, according to the discussion of Section 6.2, we should use

$$\Phi = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{6.47}$$

This means

$$Z_k = Z_{k-1} + \dot{Z}_k.$$

However, sometimes, we used

$$\Phi = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{6.48}$$

That is,

$$Z_k = Z_{k-1}.$$

Our experiment has shown that the results obtained by using these two different kinds of Φ do not have significant differences. This is because the surfaces of the scene in the first experiment vary drastically. The errors in depth map itself are larger than the errors caused by ignoring $\dot{Z}$.

In the second experiment, however, transition matrix should be represented as following

$$\Phi = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{6.49}$$

since the flat poster is in the horizontal translating, perpendicular to the Z axis. This is because the surface of the depth map is flat. Consequently, the errors in depth map are less than that in the first experiment. Therefore, the errors caused the error increases significantly if Equation (6.47) is utilized.

If the motion mode of objects is unknown, the selection of Φ is difficult. Generally speaking, the selection of Φ will affect the errors in estimation and even the system stability. But, in our system, the detection of newly visible areas makes some difference which will be discussed below.

So far, we assume that the accelerations in X, Y, and Z directions are zeros. If they are not zeros, we can adjust the thresholds in Equations (6.24-6.26) to handle the situation. When the velocities change too fast, the current states relate to not only the previous states but also the accelerations of X, Y, and Z. For instance, if $\ddot{Z}_k(x,y)$ is great, $\hat{\dot{Z}}_k(x,y)$ should be equal to $\hat{\dot{Z}}_{k-1}(x,y) + \hat{\ddot{Z}}_k(x,y)$. If one still use $\hat{\dot{Z}}_k(x,y) = \hat{\dot{Z}}_{k-1}(x,y)$, more error will be produced. Since $o2_k$ is a measurement value of $\dot{Z}_k$ and $\hat{\dot{Z}}_k^- = interpolation(\hat{\dot{Z}}_{k-1}^+)$ as discussed in Sections 6.1 and 6.4, when $\ddot{Z}_k(x,y)$ is large, $o2_k$ contains the effect of $\ddot{Z}_k(x,y)$, but $\hat{\dot{Z}}_k^-(x,y)$ may not. Therefore, $|o2_k - \hat{\dot{Z}}_k^-|$ increases. From Equation (6.24), if $|o2_k - \hat{\dot{Z}}_k^-|/|\hat{\dot{Z}}_k^-| > 30\%$, $R_k^D = 0$. This amounts to treat a newly visible area. That is, when the errors are greater than the thresholds, the Kalman filter will automatically give up the previous information and work only on the present information. As shown in Equation (6.20), the output of the system, $\hat{D}_k^+$, is just current measurement vector O_k since H is an identity matrix. Thus, the effect of this error is reduced. This scheme can guarantee that errors are always less than thresholds regardless of the selection of Φ. Therefore, one can adjust the thresholds to control the error produced by the velocity change. But if the thresholds are too large, errors will be great. If the thresholds are too small, the areas that are not newly exposed areas will be considered as newly visible areas so that errors will increase in such areas. As a result, Φ and the thresholds must be selected appropriately.

6.6.2 Newly Visible Image Areas

When camera and/or object(s) move, newly visible area exposure is unavoidable. Since in the newly visible area, the previous knowledge is not available, Kalman filter should work only on the present information. Fortunately, Kalman filter has an ability to automatically handle the newly visible area exposure problem. That is, as long as the measurement noise covariance matrix (R_k^D) is set equal to 0, the Kalman filter can ignore the previous information. Therefore, a threshold method is developed in this chapter so that R_k^D is set as 0 in an area where the newly visible area is detected.

6.6.3 Selection of R, Q and P Matrices

In [17] and [45], a simple method is used to lump all of errors together by inflating the current state covariance estimates by a small multiplicative factor in the prediction stage. Thus,

$$P_k^- = (1+\epsilon)P_{k-1}^+. \tag{6.50}$$

Then, the P_k^- is interpolated. As discussed in [17], a more exact approach is to attempt to model individual sources of error and to propagate their effects through the prediction equations. In this chapter, the interpolation process is analyzed so that the system noise covariance matrix Q_k^D and state covariance extrapolation of the Kalman filter P_k can be computed more reasonably than simply using Equation (6.50). The analysis indicates that interpolation of state variables causes errors. In fact, when interpolation is used, the system is not linear, the prediction phase of Kalman filter must be modified. In this chapter, we conclude

$$\begin{aligned}
\hat{D}_k^- &= \Phi_{k-1}[interpolation(\hat{D}_{k-1}^+)] \\
P_k^- &= \Phi_{k-1}[interpolation^2(P_{k-1}^+)]\Phi_{k-1} + Q_{k-1}^D.
\end{aligned} \tag{6.51}$$

Here, only noise caused by the interpolation is considered in the system noise covariance matrix Q_k^D because all other noises are measurement noises that are considered in the R_k^D. R_k^D is the covariance matrix of measurement noises ξ_k^D. ξ_k^D is caused by cameras, A/D converter, optical flow determination and 3-D motion estimation. Therefore, R_k^D can be derived from the measurement noise of optical flow fields. If there is any other system noise, as long as it is additive and white Gaussian noise, to expend the derivation is not difficult.

6.6.4 Experiments

The first experiment shows that $\dot{Z}$ and $\dot{Y}$ are improved more than other 3-D fields. This is because the first Kalman filter has not improved the $\dot{Z}$ and $\dot{Y}$. The other 3-D fields have moving boundaries. They were improved by the first Kalman filter.

In addition, since X, Y, Z, and $\dot{X}$ have newly visible areas, they are improved by detecting newly visible areas. However, $\dot{Z}$ is worse and $\dot{Y}$ is not improved very much when the newly visible area detection is conducted. This is because $\dot{Z}$ and $\dot{Y}$ are constant (zero) fields. When the second Kalman filter works at initial condition in the newly visible areas, noise becomes larger.

In [17], a Kalman filter is used to refine depth map. Its efficiency has been verified. Here, the Kalman filter is used to refine six 3-D fields. Since the correlation-feedback algorithm and UOFF approach to 3-D motion analysis are adopted, its efficiency is obviously improved. The similar poster translation experiment shows that our relative RMS error of depth map is 1.7%, while the RMS error of depth map is larger than 2.0% in [17].

CHAPTER 7

SUMMARY

This chapter contains a summary of our major research contributions, a review of some of unsolved problems and possible avenues for future research.

7.1 Major Contributions

The most significant contribution of our research described in this dissertation is the development of a robust algorithm based on a theoretical computational framework for the determination of dense 3-D position and motion fields from a stereo image sequence. The framework is applicable for camera motion as well as both rigid and nonrigid object(s) motion.

In the dissertation, a novel optical flow field computation approach is developed. The approach is called "correlation-feedback approach." The approach has three features distinct from any other existing approach. They are feedback, rubber window, and special refinement. The feedback technique is used for the first time in optical flow determination . Since the output optical flow field is fed back to input to compensate the output's uncertainties, the accuracy is improved considerablely. The rubber window is applied so that subpixel problem can be handled. Its refinement is based on local best matching between original image and the estimated image. The refinement and feedback compensation are used iteratively so that optical flow field computation is robust against noise and discontinuities are conserved. The approach performs especially well for nonuniform optical flow fields. This is because the refinement is a local best matching scheme. We have verified that the approach is generally convergent. At least, when the intensity is a linear function of the coordinates, the algorithm must be convergent. Our experiments also prove that the approach is one of the best ways to compute optical flow field in view of a recent comprehensive study of all of the existing optical flow techniques [40]. Although the algorithm has used Singh's subpixel estimation scheme and propagation stage [44], we also can use our new ideas such as feedback, rubber window and special refinement to other subpixel estimation and propagation schemes.

The UOFF theoretic framework and the UOFF approach to 3-D position and motion estimation have been developed by Shu and Shi ([41]-[43], [49]). The approach is different from all existing 3-D motion recovery approaches. It is based on a four frame model to compute six dense 3-D position and velocity fields. They are $Z(x,y)$, $X(x,y)$, $Y(x,y)$, $\dot{Z}(x,y)$, $\dot{X}(x,y)$, and $\dot{Y}(x,y)$ where x and y are coordinates of image plane. The approach can compute 3-D motion and surface structure at any pixel as long as optical flow vector at the pixel is known. Therefore, it can estimate any mode of motion such as rigid and/or nonrigid object(s) motion, a relative motion between camera and the 3-D world space, as well as the multiple independent objects motion.

Since the approach is based on the optical flow vector at each pixel, the accuracy of the approach is determined by the accuracy of the optical flow vector. In other words, the approach is sensitive to uncertainty of the optical flow. Especially the determination of the velocity fields such as the $\dot{Z}(x,y)$ field is sensitive to uncertainty of the optical flow. In this research, we verify that the UOFF approach is feasible with large numbers of experiments: using both computer simulation and real image sequences in the laboratory. Our formula of computing depth map is verified to be more robust than most of the previous works. Kalman filter is utilized to improve the UOFF's sensitivity to uncertainties of optical flow. Nine frame mode instead of four frame mode and stereo image sequences instead of still images are used in implementation of the UOFF approach.

In the dissertation, two Kalman filter-based algorithms have been developed. The first Kalman filter-based algorithm is used to improve accuracy along moving boundaries in optical flow field. The Kalman filter is used as a kind of refinement mechanism. It uses a fixed group of images instead of a stereo image sequence. This is different from the previous works. We prove that the convergence of optical flow field computation is nonhomogeneous. On object's boundary, convergence is faster than that in an interior area. Therefore, different pixels need different number of iterations in computation. We use the first Kalman filter to implement this idea. The efficiency is obvious. The estimation of the optical flow fields and the 3-D motion reconstruction are improved effectively. The system and measurement noise covariance matrices Q_k^u and R_k^u are analyzed carefully. By means of computation of R_k^u, discontinuity is preserved.

As we know, a stereo image sequence carries much more information than a few fixed images. Many previous works have used stereo image sequences to improve their motion estimation. But, these works are restricted to only consider the relative motion between camera and 3-D world space, for instance, a moving camera and a static 3-D world space. They usually do not consider that motion changes and there are newly visible image areas. In fact, when either camera or objects are allowed to move, or objects change shape, the newly visible areas may be exposed in the image sequence. In the newly exposed areas, the information and results based on previous image frames can not be fused with current information. Otherwise, errors will be generated. In other words, stereo image sequence must be used carefully. We have developed a threshold scheme to determine the newly exposed areas. In the newly visible image areas, the second Kalman filter works at an initial condition so that only current information is used to compute 3-D motion. In addition, the interpolation process is analyzed in detail so that covariance of the system noise (Q_k^D) and state covariance ($P_{k-1}^+(x,y)$) extrapolation of the Kalman filter can be computed more reasonably than treated in the previous works [17] [45]. Besides, the measurement noise ($\xi_k^D = N(0, R_k^D)$) of the system is considered to be generated from camera, A/D converter, and optical flow determination. The measurement noise covariance R_k^D is then derived.

To our knowledge, the error analysis and derivation of Q, P, and R matrices for these two Kalman filters in this dissertation are more advance than the previous works reported in the literatures.

As shown in our experiments, the two Kalman filters make our 3-D motion and structure reconstruction more efficient and more accurate.

7.2 Major Unsolved Issues

The dissertation research focuses on improvement of accuracy of 3-D motion and structure estimation. In fact, computation efficiency should be also considered. Because accuracy depends on image quality, if high resolution and 16 bits A/D converter are adopted, accuracy will be raised. But computation speed will become a major problem. In our framework, most time in computation is spent in the correlation-feedback stage. The correlation-feedback approach can raise accuracy considerablely. But it also increases computation complexity. Especially, when optical flow field is uniform, i.e., optical flow vectors are constant with respect to coordinates, the convergence of our algorithm is slower than a few other approaches. This is because we have used Singh's subpixel estimation scheme in our algorithm. We have not examined whether such subpixel estimation is of fastest convergence.

We have indicated that the 3-D estimation framework can work in a dynamic world. But we have not arranged experiments to verify this observation because we don't have necessary experiment conditions yet to conduct advanced experiments.

7.3 Directions for Further Research

Firstly, we should pursue faster convergence of the correlation-feedback algorithm in determining optical flow. We can try to derive faster and more applicable subpixel estimation and propagation schemes.

Secondly, we should search for a best way to use the confidence measure to enhance the accuracy of 3-D motion analysis in our UOFF approach. In an optical flow vector field, not all vectors have high accuracy. However, in many cases, not all vectors are needed as well. In fact, from the part of the vector field, 3-D information can still be reconstructed in many cases. Therefore, we can extract those flow vectors with high reliability to estimate 3-D information. The usage of confidence measure technique may enhance not only accuracy but also computation speed.

Thirdly, we should try to apply the UOFF approach to nonrigid object motion analysis. Although the framework can be used in the subject theoretically, we should use experimental work to verify the conclusion. When experimental conditions are improved, real robot navigation in dynamic environment should be pursued. We also can consider to use high resolution and high quantization in digital image acquisition to improve accuracy.

Finally, since the framework can be used for both rigid and nonrigid motion estimation, it may be extended to video signal compression.

APPENDIX A

3-D MOTION PARAMETERS DERIVATION

A.1 A

It is obvious that

$$\frac{1}{f}u^R = \frac{1}{f}\frac{\partial x^R}{\partial t}$$

Using the formulae of perspective projection and coordinate transformation, we have

$$\begin{aligned}\frac{1}{f}u^R &= \frac{(\dot{X}\cos\theta - \dot{Z}\sin\theta)Z - [(X-l)\cos\theta - Z\sin\theta]\dot{Z}}{Z^2} \\ &= \frac{(\dot{X}Z - X\dot{Z})\cos\theta}{Z^2} + \frac{l\dot{Z}\cos\theta}{Z^2}\end{aligned}$$

The use of the perspective projection leads to

$$\frac{1}{f}u^R = \frac{1}{f}u^L\cos\theta + \frac{l\dot{Z}}{Z^2}\cos\theta \tag{A.1}$$

It follows from the perspective projection that

$$\frac{1}{f}v^R = \frac{1}{f}v^L \tag{A.2}$$

A.2 B

Following the definition of u^s, one has

$$\frac{u^s}{f} = \frac{1}{f}\lim_{\delta s \to 0}\frac{\delta x}{\delta s}$$

where $\delta s = \sqrt{\tilde{x}^2 + \tilde{z}^2 + \chi^2\tilde{\theta}^2}$ for the case when two optical axes OZ and $O^R Z^R$ are coplanar. As pointed out in Section 4.2, $\tilde{z}$ is restricted to be zero in this paper. Hence, $\delta s = \sqrt{\tilde{x}^2 + \chi^2\tilde{\theta}^2}$. Approximating derivative by using average variation rate and combining the formulae of coordinate transformation and perspective projection, and the definition of u^s and v^s, one has

$$\begin{aligned}\frac{u^s}{f} &\approx (\frac{(X-l)\cos\theta - Z\sin\theta}{Z^R} - \frac{X}{Z})/\sqrt{l^2+\chi^2\theta^2} \\ &\approx (\frac{X\cos\theta - l\cos\theta - Z\sin\theta - X}{Z})/\sqrt{l^2+\chi^2\theta^2} \\ &\approx -(\frac{l}{Z} + \frac{2(1-\cos\theta)}{\sin\theta})/\sqrt{l^2+\chi^2\theta^2} \qquad (A.3)\end{aligned}$$

The above equation comes from the far-field assumption. Similarly, we have

$$\begin{aligned}\frac{v^s}{f} &\approx \{(\frac{1}{Z^R} - \frac{1}{Z})Y\}/\sqrt{l^2+\chi^2\theta^2} \\ &\approx 0 \qquad (A.4)\end{aligned}$$

A.3 C

Since $\frac{x^L}{f} = \frac{X}{Z}$ one has

$$\frac{1}{f}u^L = \frac{1}{f}\frac{\partial x^L}{\partial t} = \frac{\dot{X}Z - \dot{Z}X}{Z^2} = \frac{\dot{X}}{Z} - \frac{\dot{Z}}{Z}\frac{X}{Z} \qquad (A.5)$$

Similarly, the equation $\frac{y^L}{f} = \frac{Y}{Z}$ leads to

$$\frac{1}{f}v^L = \frac{1}{f}\frac{\partial y^L}{\partial t} = \frac{\dot{Y}Z - \dot{Z}Y}{Z^2} = \frac{\dot{Y}}{Z} - \frac{\dot{Z}}{Z}\frac{Y}{Z} \qquad (A.6)$$

APPENDIX B

COVARIANCE PROOF

Two random variables $u_0^{(k)}$ and $u_1^{(k)}$ are independent, and $u_0^{(k)} = N(u^a, \delta_0^2)$ and $u_1^{(k)} = N(u^a, \delta_1^2)$.

The covariance of the $e^{-\bar{I}_x^2(u_0^{(k)}-u^a)^2}(u_0^{(k)} - u^a)$ and $e^{-\bar{I}_x^2(u_1^{(k)}-u^a)^2}(u_1^{(k)} - u^a)$ is C_{01}.

$$
\begin{aligned}
C_{01} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(e^{-\bar{I}_x^2(u_0^{(k)}-u^a)^2})(e^{-\bar{I}_x^2(u_1^{(k)}-u^a)^2}) \\
&\quad \frac{(u_1^{(k)} - u^a)(u_0^{(k)} - u^a)}{2\pi\delta_1\delta_0} e^{\frac{-(u_1^{(k)}-u^a)^2}{2\delta_0^2}} e^{\frac{-(u_0^{(k)}-u^a)^2}{2\delta_1^2}} du_1^{(k)} du_0^{(k)} \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{(u_1^{(k)} - u^a)(u_0^{(k)} - u^a)}{2\pi\delta_1\delta_0} \\
&\quad e^{-(\bar{I}_x^2+\frac{1}{2\delta_0^2})(u_0^{(k)}-u^a)^2} e^{-(\bar{I}_x^2+\frac{1}{2\delta_1^2})(u_1^{(k)}-u^a)^2} du_1^{(k)} du_0^{(k)} \\
&= \frac{1}{2\pi\delta_1\delta_0}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(u_1^{(k)} - u^a)(u_0^{(k)} - u^a) \\
&\quad e^{-(\bar{I}_x^2+\frac{1}{2\delta_0^2})(u_0^{(k)}-u^a)^2} e^{-(\bar{I}_x^2+\frac{1}{2\delta_1^2})(u_1^{(k)}-u^a)^2} du_1^{(k)} du_0^{(k)} \qquad \text{(B.1)}
\end{aligned}
$$

Assuming that two random variables v_0 and v_1 are independent of each other, v_0 is $N(u^a, \frac{1}{2\bar{I}_x^2+\delta_0^{-2}})$, v_1 is $N(u^a, \frac{1}{2\bar{I}_x^2+\delta_1^{-2}})$, the Covariance of v_0 and v_1 is $C_{v_0 v_1}$. Therefore,

$$
\begin{aligned}
C_{v_0 v_1} &= \frac{1}{2\pi(2\bar{I}_x^2 + \delta_1^{-2})^{1/2}(2\bar{I}_x^2 + \delta_1^{-2})^{1/2}}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(v_1 - u^a)(v_0 - u^a) \\
&\quad e^{-(\bar{I}_x^2+\frac{1}{2\delta_1^2})(v_1-u^a)^2} e^{-(\bar{I}_x^2+\frac{1}{2\delta_0^2})(v_0-u^a)^2} dv_0 dv_1 \\
&= 0 \qquad \text{(B.2)}
\end{aligned}
$$

Comparing equation (B.1) with equation (B.2), one can conclude that $C_{01} = 0$.

APPENDIX C

DERIVATION OF Q AND P MATRICES

In Section 6.1, it is pointed out that due to image digitization a linear interpolation is needed to obtain state vector $D_k(x,y)$ at moment k from $D_{k-1}(x,y)$ at moment $k-1$.

In order to simplify the derivation, one dimensional case is considered, i.e., we only discuss the derivation for the first component of the state and measurement vector: the depth $Z(x,y)$. The results we derived can be readily available for the other three components. Furthermore, the transition matrix Φ is assumed to be diagonal. That is, ϕ_{11} is assumed to be a constant. Thus, Equation (6.3) leads to

$$\begin{aligned} Z_k(x,y) &= \phi_{11}(interpolation(Z_{k-1}(x,y))) \\ &= \phi_{11}\tilde{Z}_{k-1}(x,y) \\ &= \phi_{11}\bar{Z}_{k-1}(x,y) + \phi_{11}\varphi_k \\ &= \phi_{11}\bar{Z}_{k-1}(x,y) + \eta_k^Z(x,y). \end{aligned} \tag{C.1}$$

Let's examine how noise is caused by the interpolation process.

In Figure 6.4, depth maps of $Z_k(i_2,j_2)$ and $Z_{k-1}(i_1,j_1)$ are shown. Since the 3-D object(s) is moving, its depth map is also moving. The velocity of motion of the depth map is the optical flow vector (d_2^u, d_4^u), i.e., the measured values of (u^L, v^L). In other words, during an iteration, $\tilde{Z}_{k-1}(i_2,j_2)$ should have been obtained by shifting $Z_{k-1}(i_1,j_1)$ at the velocity $(d_2^u(i_2,j_2), d_4^u(i_2,j_2))$, i.e.,

$$\begin{aligned} \tilde{Z}_{k-1}(i_2,j_2) &= Z_{k-1}(i_1 + d_2^u(i_2,j_2), j_1 + d_4^u(i_2,j_2)) \\ &= Z_{k-1}(i_2 - d_2^u(i_2,j_2), j_2 - d_4^u(i_2,j_2)), \end{aligned} \tag{C.2}$$

But, unfortunately, $d_2^u(i_2, j_2)$ and $d_4^u(i_2, j_2)$ generally may not be integer-valued. Therefore, Equation (C.2) may not hold. Refer to Figure 6.4, one has

$$\tilde{Z}_{k-1}(i_2, j_2) = Z_{k-1}(x_1, y_1) \tag{C.3}$$

where x_1 and y_1 are real and

$$\begin{aligned} x_1 &= i_2 - d_2^u(i_2, j_2) \\ y_1 &= j_2 - d_4^u(i_2, j_2), \end{aligned} \tag{C.4}$$

From Figure 6.4, it is clear that

$$\begin{aligned} Z_{k-1}(x_1, y_1) &= interpolation(Z_{k-1}(i_1, j_1)) \\ &= b[aZ_{k-1}(i_{11}, j_{11}) + (1-a)Z_{k-1}(i_{11}+1, j_{11})] + \\ &= (1-b)[aZ_{k-1}(i_{11}, j_{11}+1) + (1-a)Z_{k-1}(i_{11}+1, j_{11}+1)] \end{aligned} \tag{C.5}$$

The equation is a standard bilinear interpolation. Figure 6.4 shows the relationship among $Z_{k-1}(x_1, y_1)$, $Z_{k-1}(i_{11}, j_{11})$, $Z_{k-1}(i_{11}+1, j_{11})$, $Z_{k-1}(i_{11}, j_{11}+1)$ and $Z_{k-1}(i_{11}+1, j_{11}+1)$ and the meanings of a and b. $Z_{k-1}(i_{11}, j_{11})$ is located at the pixel (i_{11}, j_{11}) which is such a pixel among the four that is closest to (x_1, y_1) and at the north-west corner of (x_1, y_1). This can be represented as following equations

$$\begin{aligned} i_{11} &= int(x_1) = i_2 - int(d_2^u(i_2, j_2)) - 1 \\ j_{11} &= int(y_1) = j_2 - int(d_4^u(i_2, j_2)) - 1 \end{aligned} \tag{C.6}$$

where $int()$ is an operation that changes real number into integer that is always less than the real number. In addition,

$$1 - a = x_1 - i_{11},$$

$$1 - b = y_1 - j_{11}.$$

Assuming that d_2^{ua} and d_4^{ua} are the true values of d_2^u and d_4^u respectively, and

$$d_2^u = d_2^{ua} + \mu_x,$$

$$d_4^u = d_4^{ua} + \mu_y,$$

where $\mu_x = N(0, \delta_x)$ and $\mu_y = N(0, \delta_y)$, one has

$$\begin{aligned}
1 - a = x_1 - i_{11} &= 1 - d_2^{ua}(i_2, j_2) + int(d_2^{ua}(i_2, j_2)) + \mu_x \triangleq 1 - \bar{a} + \mu_x \\
a &= \bar{a} - \mu_x \\
1 - b = y_1 - j_{11} &= 1 - d_4^{ua}(i_2, j_2) + int(d_2^{ua}(i_2, j_2)) + \mu_y \triangleq 1 - \bar{b} + \mu_y \\
b &= \bar{b} - \mu_y
\end{aligned} \tag{C.7}$$

Though *interpolation* is a bilinear interpolation $Z_{k-1}(x_1, y_1) - \hat{Z}_{k-1}^+(x_1, y_1)$ may not be linearly related with $Z_{k-1}(i_{11}, j_{11}) - \hat{Z}_{k-1}^+(i_{11}, j_{11})$, $Z_{k-1}(i_{11}+1, j_{11}) - \hat{Z}_{k-1}^+(i_{11}+1, j_{11})$, $Z_{k-1}(i_{11}, j_{11}+1) - \hat{Z}_{k-1}^+(i_{11}, j_{11}+1)$, and $Z_{k-1}(i_{11}+1, j_{11}+1) - \hat{Z}_{k-1}^+(i_{11}+1, j_{11}+1)]$. Therefore errors w_1, w_2, and w_3 are produced. Figure 6.5 clearly shows how these errors are produced.

From the Figure 6.5, one can know

$$\begin{aligned}
P_a &= a[Z_{k-1}(i_{11}, j_{11}) - \hat{Z}_{k-1}^+(i_{11}, j_{11})] + (1 - a)[Z_{k-1}(i_{11}+1, j_{11}) \\
&\quad - \hat{Z}_{k-1}^+(i_{11}+1, j_{11})]
\end{aligned} \tag{C.8}$$

Obviously $P_A \neq P_a$. We assume

$$P_A = P_a + w_1, \quad w_1 = N(0, \delta_{w_1}). \tag{C.9}$$

Similarly, we know

$$\begin{aligned}
P_b &= a[Z_{k-1}(i_{11}, j_{11}+1) - \hat{Z}_{k-1}^+(i_{11}, j_{11}+1)] + (1 - a)[Z_{k-1}(i_{11}+1, j_{11}+1) \\
&\quad - \hat{Z}_{k-1}^+(i_{11}+1, j_{11}+1)]
\end{aligned} \tag{C.10}$$

and $P_B \neq P_b$

$$P_B = P_b + w_2, \quad w_2 = N(0, \delta_{w_2}). \tag{C.11}$$

Furthermore, from Figure6.5

$$P_d = bP_a + (1 - b)P_b, \tag{C.12}$$

and $P_d \neq P_D$. We assume

$$P_D = P_d + w_3, \quad w_3 = N(0, \delta_{w_3}). \tag{C.13}$$

Substituting Equations (C.9 - C.12) into Equation (C.13), one has

$$P_D = b(P_a + w_1) + (1 - b)(P_b + w_2) + w_3. \tag{C.14}$$

From Figure 6.5, one has

$$P_D = Z_{k-1}(x_1, y_1) - \hat{Z}^+_{k-1}(x_1, y_1),$$

where $\hat{Z}^+_{k-1}(x_1, y_1)$ is an estimate of $Z_{k-1}(x_1, y_1)$. Hence the above discussion leads to

$$\begin{aligned}
& Z_{k-1}(x_1, y_1) - \hat{Z}^+_{k-1}(x_1, y_1) \\
= \; & b\{a[Z_{k-1}(i_{11}, j_{11}) - \hat{Z}^+_{k-1}(i_{11}, j_{11})] + (1 - a)[Z_{k-1}(i_{11} + 1, j_{11}) \\
& -\hat{Z}^+_{k-1}(i_{11} + 1, j_{11})] + w_1\} + (1 - b)\{a[Z_{k-1}(i_{11}, j_{11} + 1) \\
& -\hat{Z}^+_{k-1}(i_{11}, j_{11} + 1)] + (1 - a)[Z_{k-1}(i_{11} + 1, j_{11} + 1) \\
& -\hat{Z}^+_{k-1}(i_{11} + 1, j_{11} + 1)] + w_2\} + w_3
\end{aligned} \tag{C.15}$$

where the noises w_1, w_2, and w_3 are produced by the linearization.

Equation (C.15) can be rearranged as follows.

$$\begin{aligned}
& Z_{k-1}(x_1, y_1) - \hat{Z}^+_{k-1}(x_1, y_1) \\
= \; & b[a(Z_{k-1}(i_{11}, j_{11}) - \hat{Z}^+_{k-1}(i_{11}, j_{11}) + (1 - a)(Z_{k-1}(i_{11} + 1, j_{11}) \\
& -\hat{Z}^+_{k-1}(i_{11} + 1, j_{11})] + (1 - b)[a(Z_{k-1}(i_{11}, j_{11} + 1) \\
& -\hat{Z}^+_{k-1}(i_{11}, j_{11} + 1) + (1 - a)(Z^a_{k-1}(i_{11} + 1, j_{11} + 1) \\
& -\hat{Z}^+_{k-1}(i_{11} + 1, j_{11} + 1)] + w_3 + bw_1 + (1 - b)w_2
\end{aligned} \tag{C.16}$$

Substituting Equation (C.7) into Equation (C.16), one has

$$Z_{k-1}(x_1, y_1) - \hat{Z}^+_{k-1}(x_1, y_1)$$

$$
\begin{aligned}
= \quad & (\bar{b} - \mu_y)[(\bar{a} - \mu_x)(Z_{k-1}(i_{11}, j_{11}) - \hat{Z}^{+}_{k-1}(i_{11}, j_{11})) \\
& +(1 - \bar{a} + \mu_x)(Z_{k-1}(i_{11} + 1, j_{11}) - \hat{Z}^{+}_{k-1}(i_{11} + 1, j_{11}))] \\
& +(1 - b + \mu_y)[(\bar{a} - \mu_x)(Z_{k-1}(i_{11}, j_{11} + 1) - \hat{Z}^{+}_{k-1}(i_{11}, j_{11} + 1)) \\
& +(1 - \bar{a} + \mu_x)(Z_{k-1}(i_{11} + 1, j_{11} + 1) - \hat{Z}^{+}_{k-1}(i_{11} + 1, j_{11} + 1))] + w_3 \\
& +(b - \mu_y)w_1 + (1 - \bar{b} + \mu_y)w_2. \qquad \text{(C.17)}
\end{aligned}
$$

From Equation(C.3), one knows

$$\tilde{Z}_{k-1}(i_2, j_2) - \hat{\tilde{Z}}_{k-1}(i_2, j_2) = Z_{k-1}(x_1, y_1) - \hat{Z}^{+}_{k-1}(x_1, y_1).$$

From Equation (6.3), one has

$$
\begin{aligned}
& \tilde{Z}_{k-1}(i_2, j_2) - \hat{\tilde{Z}}_{k-1}(i_2, j_2) \\
= \quad & Z_{k-1}(x_1, y_1) - \hat{Z}^{+}_{k-1}(x_1, y_1) \\
= \quad & \bar{Z}_{k-1}(i_2, j_2) - \hat{\bar{Z}}_{k-1}(i_2, j_2) + \varphi_k \qquad \text{(C.18)}
\end{aligned}
$$

φ_k is a noise produced by interpolation. $\bar{Z}_{k-1}(i_2, j_2) - \hat{\bar{Z}}_{k-1}(i_2, j_2)$ is not related to the noise. In order to obtain $\bar{Z}_{k-1}(i_2, j_2) - \hat{\bar{Z}}_{k-1}(i_2, j_2)$ and φ_k, we should rearrange the Equation (C.17) as follows.

$$
\begin{aligned}
& Z_{k-1}(x_1, y_1) - \hat{Z}^{+}_{k-1}(x_1, y_1) \\
= \quad & \{\bar{Z}_{k-1}(i_2, j_2) - \hat{\bar{Z}}_{k-1}(i_2, j_2)\} + \{\varphi_k\} \\
\triangleq \quad & \{\bar{b}[\bar{a}(Z_{k-1}(i_{11}, j_{11}) - \hat{Z}^{+}_{k-1}(i_{11}, j_{11})) \\
& +(1 - \bar{a})(Z_{k-1}(i_{11} + 1, j_{11}) - \hat{Z}^{+}_{k-1}(i_{11} + 1, j_{11}))] \\
& +(1 - \bar{b})[\bar{a}(Z_{k-1}(i_{11}, j_{11} + 1) - \hat{Z}^{+}_{k-1}(i_{11}, j_{11} + 1)) \\
& +(1 - \bar{a})(Z_{k-1}(i_{11}, j_{11} + 1) - \hat{Z}^{+}_{k-1}(i_{11}, j_{11} + 1))]\} + \\
& \{-\bar{b}[(Z_{k-1}(i_{11}, j_{11}) - \hat{Z}^{+}_{k-1}(i_{11}, j_{11})) - (Z_{k-1}(i_{11} + 1, j_{11}) - \hat{Z}^{+}_{k-1}(i_{11} + 1, j_{11}))]\mu_x \\
& +[(Z_{k-1}(i_{11}, j_{11}) - \hat{Z}^{+}_{k-1}(i_{11}, j_{11})) - (Z_{k-1}(i_{11} + 1, j_{11}) - \hat{Z}^{+}_{k-1}(i_{11} + 1, j_{11}))]\mu_x\mu_y \\
& -(1 - \bar{b})[(Z_{k-1}(i_{11}, j_{11} + 1) - \hat{Z}^{+}_{k-1}(i_{11}, j_{11} + 1))
\end{aligned}
$$

$$
\begin{aligned}
&-(Z_{k-1}(i_{11}+1,j_{11}+1)-\hat{Z}_{k-1}^{+}(i_{11}+1,j_{11}+1))]\mu_x \\
&-[(Z_{k-1}(i_{11},j_{11}+1)-\hat{Z}_{k-1}^{+}(i_{11},j_{11}+1)) \\
&-(Z_{k-1}(i_{11}+1,j_{11}+1)-\hat{Z}_{k-1}^{+}(i_{11}+1,j_{11}+1))]\mu_x\mu_y \\
&-\bar{a}[(Z_{k-1}(i_{11},j_{11})-\hat{Z}_{k-1}^{+}(i_{11},j_{11}))-(Z_{k-1}(i_{11},j_{11}+1)-\hat{Z}_{k-1}^{+}(i_{11},j_{11}+1))]\mu_y \\
&-(1-\bar{a})[(Z_{k-1}(i_{11}+1,j_{11})-\hat{Z}_{k-1}^{+}(i_{11}+1,j_{11})) \\
&-(Z_{k-1}(i_{11}+1,j_{11}+1)-\hat{Z}_{k-1}^{+}(i_{11}+1,j_{11}+1))]\mu_y \\
&+(\bar{b}-\mu_y)w_1+(1-\bar{b}+\mu_y)w_2+w_3\}
\end{aligned}
\tag{C.19}
$$

where $\bar{Z}_{k-1}(i_2,j_2)-\hat{\bar{Z}}_{k-1}^{+}(i_2,j_2)$ and $\varphi_{k-1}^{Z}(i_2,j_2)$ are, respectively, the first and the second terms encompassed by the brackets in the right-hand side of the last equal sign.

According to Equation (C.1),

$$
\begin{aligned}
Z_k(i_2,j_2) &= \phi_{11}\tilde{Z}_{k-1}(i_2,j_2) \\
\hat{Z}_k^{-}(i_2,j_2) &= \phi_{11}\hat{\tilde{Z}}_{k-1}(i_2,j_2)
\end{aligned}
\tag{C.20}
$$

Thus,

$$
Z_k(i_2,j_2)-\hat{Z}_k^{-}(i_2,j_2)=\phi_{11}(\tilde{Z}_{k-1}(i_2,j_2)-\hat{\tilde{Z}}_{k-1}(i_2,j_2))
\tag{C.21}
$$

Assuming that all of the random noises are independent of each other, from Equation (C.21) and (C.18), one has

$$
\begin{aligned}
&p_{11k}^{-}(i_2,j_2) \\
={}& E\{(Z_k(i_2,j_2-\hat{Z}_k^{-}(i_2,j_2))(Z_k(i_2,j_2)-\hat{Z}_k^{-}(i_2,j_2))\} \\
={}& E\{[\phi_{11}(\tilde{Z}_{k-1}(i_2,j_2)-\hat{\tilde{Z}}_{k-1}^{+}(i_2,j_2))]^2\} \\
={}& E\{(\phi_{11}\bar{Z}_{k-1}(i_2,j_2)-\phi_{11}\hat{\bar{Z}}_{k-1}^{+}(i_2,j_2)+\phi_{11}\varphi_{k-1}^{Z}(i_2,j_2))^2\} \\
={}& E\{(\bar{Z}_{k-1}(i_2,j_2)-\hat{\bar{Z}}_{k-1}^{+}(i_2,j_2))^2\}(\phi_{11})^2+E\{(\varphi_{k-1}^{Z}(i_2,j_2))^2\}(\phi_{11})^2
\end{aligned}
\tag{C.22}
$$

where $p_{11k}^{-}(i_2,j_2)$ is the first element of the state covariance $P_k^{-}(i_2,j_2)$.

If we define

$$\rho_k^Z(i_2,j_2) = E\{(\bar{Z}_{k-1}(i_2,j_2) - \hat{\bar{Z}}^{+}_{k-1}(i_2,j_2))^2\} \tag{C.23}$$

From the expression of $\bar{Z}_{k-1}(i_2,j_2) - \hat{\bar{Z}}^{+}_{k-1}(i_2,j_2)$ defined in Equation (C.19),

$$\begin{aligned} & \rho_k^Z(i_2,j_2) \\ = \; & E\{(\bar{b}[\bar{a}(Z_{k-1}(i_{11},j_{11}) - \hat{Z}^{+}_{k-1}(i_{11},j_{11})) \\ & +(1-\bar{a})(Z_{k-1}(i_{11}+1,j_{11}) - \hat{Z}^{+}_{k-1}(i_{11}+1,j_{11}))] \\ & +(1-\bar{b})[\bar{a}(Z_{k-1}(i_{11},j_{11}+1) - \hat{Z}^{+}_{k-1}(i_{11},j_{11}+1)) \\ & +(1-\bar{a})(Z_{k-1}(i_{11}+1,j_{11}+1) - \hat{Z}^{+}_{k-1}(i_{11}+1,j_{11}+1))])^2\} \end{aligned} \tag{C.24}$$

If we define

$$C = \begin{bmatrix} (\bar{b}\times\bar{a}) \\ (\bar{b}\times(1-\bar{a})) \\ ((1-\bar{b})\times\bar{a}) \\ ((1-\bar{b}\times(1-\bar{a})) \end{bmatrix} \tag{C.25}$$

and

$$B_{k-1} = \begin{bmatrix} Z_{k-1}(i_{11},j_{11}) - \hat{Z}^{+}_{k-1}(i_{11},j_{11}) \\ Z_{k-1}(i_{11}+1,j_{11}) - \hat{Z}^{+}_{k-1}(i_{11}+1,j_{11}) \\ Z_{k-1}(i_{11},j_{11}+1) - \hat{Z}^{+}_{k-1}(i_{11},j_{11}+1) \\ Z_{k-1}(i_{11}+1,j_{11}+1) - \hat{Z}^{+}_{k-1}(i_{11}+1,j_{11}+1) \end{bmatrix}, \tag{C.26}$$

then

$$\begin{aligned} \rho_k^Z(i_2,j_2) = E\{C^T B_{k-1} B^T_{k-1} C\} & \\ & = C^T E\{B_{k-1}B^T_{k-1}\}C. \end{aligned} \tag{C.27}$$

Apparently, the four diagonal elements of $E\{B_{k-1}B^T_{k-1}\}$ are $p^{+}_{11(k-1)}(i_{11},j_{11})$, $p^{+}_{11(k-1)}(i_{11}+1,j_{11})$, $p^{+}_{11(k-1)}(i_{11},j_{11}+1)$, and $p^{+}_{11(k-1)}(i_{11}+1,j_{11}+1)$ that are four elements of the state covariance matrix field $(P^{+}_{k-1}(i_1,j_1))$ at the last moment $(k-1)$. If we assume that $Z_{k-1}(i_{11},j_{11}) - \hat{Z}^{+}_{k-1}(i_{11},j_{11})$, $Z_{k-1}(i_{11}+1,j_{11}) - \hat{Z}^{+}_{k-1}(i_{11}+1,j_{11})$, $Z_{k-1}(i_{11},j_{11}+1) - \hat{Z}^{+}_{k-1}(i_{11},j_{11}+1)$, and $Z_{k-1}(i_{11}+1,j_{11}+1) - \hat{Z}^{+}_{k-1}(i_{11}+1,j_{11}+1)$ are independent of each other as previous works [17] and [45],

$$E\{B_{k-1}B^T_{k-1}\}$$

$$= \begin{pmatrix} p_{11}^{+}(i_{11}, j_{11}) & 0 & 0 & 0 \\ 0 & p_{11}^{+}(i_{11}+1, j_{11}) & 0 & 0 \\ 0 & 0 & p_{11}^{+}(i_{11}, j_{11}+1) & 0 \\ 0 & 0 & 0 & p_{11}^{+}(i_{11}+1, j_{11}+1) \end{pmatrix} \tag{C.28}$$

Thus,

$$\begin{aligned} & C^T E\{B_{k-1}B_{k-1}^T\}C \\ = \; & \bar{b}^2\bar{a}^2 p_{11(k-1)}^{+}(i_{11}, j_{11}) + \bar{b}^2(1-\bar{a})^2 p_{11(k-1)}^{+}(i_{11}+1, j_{11}) \\ & +(1-\bar{b})^2\bar{a}^2 p_{11(k-1)}^{+}(i_{11}, j_{11}+1) \\ & +(1-\bar{b})^2(1-\bar{a})^2 p_{11(k-1)}^{+}(i_{11}+1, j_{11}+1) \end{aligned} \tag{C.29}$$

In order to cite the formula later with ease, we define

$$\begin{aligned} & interpolation^2(p_{11(k-1)}^{+}) \\ = \; & \bar{b}^2\bar{a}^2 p_{11(k-1)}^{+}(i_{11}, j_{11}) + \bar{b}^2(1-\bar{a})^2 p_{11(k-1)}^{+}(i_{11}+1, j_{11}) \\ & +(1-\bar{b})^2\bar{a}^2 p_{11(k-1)}^{+}(i_{11}, j_{11}+1) \\ & +(1-\bar{b})^2(1-\bar{a})^2 p_{11(k-1)}^{+}(i_{11}+1, j_{11}+1) \end{aligned} \tag{C.30}$$

It is worth noting that

$$\begin{aligned} & interpolation(p_{11(k-1)}^{+}) \\ = \; & \bar{b}\bar{a} p_{11(k-1)}^{+}(i_{11}, j_{11}) + \bar{b}(1-\bar{a}) p_{11(k-1)}^{+}(i_{11}+1, j_{11}) \\ & +(1-\bar{b})\bar{a} p_{11(k-1)}^{+}(i_{11}, j_{11}+1) \\ & (1-\bar{b})(1-\bar{a}) p_{11(k-1)}^{+}(i_{11}+1, j_{11}+1) \end{aligned} \tag{C.31}$$

If we define

$$\sigma_k^Z(i_2, j_2) = E\{(\varphi_{k-1}^Z(i_2, j_2))^2\}, \tag{C.32}$$

substituting the expression of $\varphi_{k-1}^Z(i_2, j_2)$ defined in Equation (C.19) into Equation (C.32) and assuming that all random noises in $\varphi_{k-1}^Z(i_2, j_2)$ are independent of each other, then one has

$$
\begin{aligned}
& \sigma_k^Z(i_2, j_2) \\
= \; & [\bar{b}((Z_{k-1}(i_{11}, j_{11}) - \hat{Z}_{k-1}^{+}(i_{11}, j_{11})) - (Z_{k-1}(i_{11}+1, j_{11}) - \hat{Z}_{k-1}^{+}(i_{11}+1, j_{11}))) \\
& +(1-\bar{b})((Z_{k-1}(i_{11}, j_{11}+1) - \hat{Z}_{k-1}^{+}(i_{11}, j_{11}+1)) \\
& -(Z_{k-1}(i_{11}+1, j_{11}+1) - \hat{Z}_{k-1}^{+}(i_{11}+1, j_{11}+1)))]^2 \delta_x^2 \\
& +[\bar{a}((Z_{k-1}(i_{11}, j_{11}) - \hat{Z}_{k-1}^{+}(i_{11}, j_{11})) - (Z_{k-1}(i_{11}, j_{11}+1) - \hat{Z}_{k-1}^{+}(i_{11}, j_{11}+1))) + \\
& (1-\bar{a})((Z_{k-1}(i_{11}+1, j_{11}) - \hat{Z}_{k-1}^{+}(i_{11}+1, j_{11})) \\
& -(Z_{k-1}(i_{11}+1, j_{11}+1) - \hat{Z}_{k-1}^{+}(i_{11}+1, j_{11}+1)))]^2 \delta_y^2 \\
& +\bar{b}^2 \delta_{w1}^2 + (1-\bar{b})^2 \delta_{w2}^2 + \delta_{w3}^2.
\end{aligned}
\tag{C.33}
$$

Thus, according to Equations (C.22, C.23, C.30 and C.33), one has

$$
\begin{aligned}
& p_{11k}^{-}(i_2, j_2) \\
= \; & \phi_{11}[\rho_k^Z(i_2, j_2) + \sigma_k^Z(i_2, j_2)]\phi_{11} \\
= \; & \phi_{11}[interpolation^2(p_{11(k-1)}^{+}(i_1, j_1)) + \sigma_k^Z(i_2, j_2)]\phi_{11}.
\end{aligned}
\tag{C.34}
$$

Similarly, one can obtain

$$
\begin{aligned}
& p_{22k}^{-}(i_2, j_2) \\
= \; & \phi_{22}[interpolation^2(p_{22(k-1)}^{+}(i_1, j_1)) + \sigma_k^{\dot{Z}}(i_2, j_2)]\phi_{22},
\end{aligned}
\tag{C.35}
$$

$$
\begin{aligned}
& p_{33k}^{-}(i_2, j_2) \\
= \; & \phi_{33}[interpolation^2(p_{33(k-1)}^{+}(i_1, j_1)) + \sigma_k^{\dot{X}}(i_2, j_2)]\phi_{33},
\end{aligned}
\tag{C.36}
$$

and

$$
\begin{aligned}
& p_{44k}^{-}(i_2, j_2) \\
= \; & \phi_{44}[interpolation^2(p_{44(k-1)}^{+}(i_1, j_1)) + \sigma_k^{\dot{Y}}(i_2, j_2)]\phi_{44}.
\end{aligned}
\tag{C.37}
$$

When Equations (C.34 - C.37) are represented in terms of the matrix equation,

$$P_k^- = \Phi[interpolation^2(P_{k-1}^+)]\Phi^T + Q_k^D \tag{C.38}$$

where

$$Q_k^D(i_2, j_2) = \Phi \begin{pmatrix} \sigma_k^Z(i_2, j_2) & 0 & 0 & 0 \\ 0 & \sigma_k^{\dot{Z}}(i_2, j_2) & 0 & 0 \\ 0 & 0 & \sigma_k^{\dot{X}}(i_2, j_2) & 0 \\ 0 & 0 & 0 & \sigma_k^{\dot{X}}(i_2, j_2) \end{pmatrix} \Phi^T. \tag{C.39}$$

REFERENCES

1. W.B.Thompson, "Introduction to the special issue on visual motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 5, pp. 449-450, May 1989.
2. Harry Wechsler, Computational Vision, *Academic Press. Inc.*, pp.273-292, 1990.
3. Enrico De Micheli, Vincent Torre, and Sergio Uras, "The accuracy of the computation of optical flow and of the recovery of motion parameters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 5, pp. 434-447, 1993.
4. B. K. P. Horn and E. J. Weldon Jr., "Direct methods for recovering motion," *International Journal of Computer Vision*, 2, pp. 51-76, 1988.
5. J. Heel and S. Negahdaripour, "Time-sequential structure and motion estimation without optical flow," *Proceedings of SPIE Vol 1260 Sensing and Reconstruction of Three-Dimensional Objects and Scenes*, pp. 50-61, Feb. 1990.
6. Shahriar Negahdaripour and Berthold K. P. Horn, "Direct passive navigation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 1, pp. 168-176, Jan. 1987.
7. John Aloimonos and Jen-yves Herve, "Correspondenceless stereo and motion: planar surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 504-510, May 1990.
8. B. Hayashi and S. Negahdaripour, "Direct motion stereo," *Proceedings SPIE Vol 1260 Sensing and Reconstruction of Three-Dimensional Objects and Scenes*, pp. 78-85, Feb. 1990.
9. B.K.P. Horn and B.G. Schunck, "Determining optical flow," *Artificial Intelligence 17* pp. 185-203, 1981.
10. J. Weng, T. S. Huang and N. Ahuja, "Motion and Structure from two perspective views: algorithm, error analysis and error estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 5, pp. 451-476, May 1989.
11. Pierre Moulin and Alexander Loui, "Hierarchical-finite-element parameterization and estimation of optical flow field," *Proceedings of IEEE Signal Processing Society 8th Workshop on Image and Multidimensional Signal Processing*, pp. 130-131, Cannes, France, Sept. 8-10, 1993.

12. T. S. Huang, S. C. Reddy, and K. Aizawa, "Human facial motion modeling, analysis, and synthesis for video compression," *SPIE Visual Comm. and Image Proc' 91*, vol. 1605, pp. 234-241, Nov. 1991.

13. H. Li, P. Roivainen, and Forcheimer, "3-D motion estimation in model-based facial image coding," *IEEE Trans. Patt. Anal. Mach. Intel.*, vol. 15, pp. 545-555, June, 1993.

14. T. S. Huang, "Modeling, analysis, and visualization of nonrigid object motion," *Proceedings of 10th International Conference on Pattern Recognition*, vol. 1, pp. 361-364, Atlantic City, New Jersey, June, 1990.

15. J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images – a review," *Proceedings of the IEEE* , vol. 76, no. 8, pp. 917-935, Aug. 1988.

16. P. Anandan, "Computing dense displacement fields with confidence measures in scenes containing occlusion," *DARPA Workshop on Image Understanding*, pp. 236-246, 1984.

17. L. Matthies, R. Szeliski and T. Kanade, "Kalman filter-based algorithms for estimating depth from image sequences," *International Journal of Computer Vision*, vol. 3, pp. 209-236, 1989.

18. L. Matthies, "Dynamic stereo vision," Ph.D. Dissertation, Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1989.

19. W. Eukelman, "Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences," Workshop on motion: Representation and analysis, S.C., pp.81-87, 1986.

20. F. Glazer, "Hierarchical motion detection," Ph.D. dissertation, COINS Department, University of Massachusetts, Amherst, Ma., Jan. 1987.

21. E. C. Hildreth, "The measurement of visual motion," Ph.D. dissertation, Dept. of Electrical Engineering and Computer Science, MIT, Cambridge, Ma., 1983.

22. H. H. Nagel, "Constraints for the Estimation of Displacement Vector Fields from Image Sequences," *IJCCAI-83*, Karlsruhe, W. Germany, pp945-951, 1983.

23. H. H. Nagel, "Dynamic stereo vision in a robot feedback loop based on the evaluation of multiple interconnected displacement vector fields," *Proceedings of the International Symp. on Robotics Research*, pp.200-206, 1985.

24. P. J. Burt, "The pyramid as a structure for efficient computation," in A. Rosenfeld, editor, Multi Resolution Image Processing and Analysis, pp. 6-37, Springer Verlag, 1984.

25. P. J. Burt, C. Yen and X. Xu, "Multi-resolution flow-through motion analysis," *IEEE CVPR Conference Proceedings*, pp.246-252, 1983.

26. F. Glazer, G. Reynolds and P. Anandan, "scene matching by hierarchical correlation", *IEEE CVPR Conference*, pp. 432-441, June 1983.

27. L. H. Quam, "Hierarchical Warp Stereo," *Proceedings of DARPA IU Workshop*, Louisiana, pp. 149-156. IU Workshop 84, Oct. 1984.

28. R. Y. Wong and E. L. Hall,"sequential hierarchical scene matching," *IEEE transactions on Computers*, Vol.27, No. 4, pp. 359-366, 1978.

29. S. T. Barnard and W. B. Thompson, "Disparity analysis of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, Number 4, pp.333-340, July 1980.

30. W. E. L. Grimson, "Computational experiments with a feature based stereo algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, No.1, pp. 17-34, 1985.

31. D. Marr and T. Poggio, "A computational theory of human stereo vision," *Proc. Royal Society of London*, Ser. B204, pp. 301-308, 1979.

32. E. H. Adelson and J. R. Bergen, "Spatiotemproal energy models for the perception of motion," *Journal of Optical Society of America A*, Vol.2, No. 2, pp. 284-299, 1985.

33. J. P. H. Van Saanten and G. Sperling, "Elaborated Reichardt detectors," *Journal of Optical Society of America A* , Vol. 2, No. 2, pp. 300-321, 1985.

34. Joseph K. Kearney, B. William Thompson and Daniel L. Boley, "Optical flow estimation: Error analysis of Gradient-Based Methods with local optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 2, March 1987.

35. D. H. Ballard, "Eye fixation and early vision: kinetic depth," *ICVV*, pp.524-531, Tampa, Florida, 1988.

36. A. M. Waxman and J. H. Duncan, "Binocular image flow: Steps toward stereo-motion fusion," *IEEE Trans. Pattern Anal. machine Intell.*, vol. PAMI-8, no. 6, pp. 715-729, Nov. 1986.

37. A. M. Waxman and S. Sinha, "Dynamic stereo: Passive ranging to moving objects from relative image flows," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 4, pp. 406-412, July 1986.

38. Rosenfeld Azriel and C. Kuk Avinash, Digital Picture Processing, *Academic Press*, Edition, Vol. 2, pp. 33-36, 1982.

39. P. Anandan, "Measuring visual motion from image sequences" *Ph.D. thesis*, COINS Department, University of Massachusetts, Amherst, 1987.

40. J. L. Barron, D. J. Fleet and S. S. Beauchemin, "Performance of optical flow techniques," *Technical report No. 299*, Dept. of Computer Science, Uni. of Western Ontario, London, Ontario, N6A 5B7 and Dept. of Computing Science, Queens Uni., Kingston Ontario, K7L 3N6, July 1992.

41. C. Q. Shu and Y. Q. Shi, "A new approach to motion analysis from a sequence of stereo images," *Technical Report No. 18*, Electronic Imaging Laboratory, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, 1990.

42. C. Q. Shu and Y. Q. Shi, "Computation of motion from stereo image sequence using the unified optical flow field," *SPIE's 1990 International Symposium on Optical and Optoelectronic Applied Science and Engineering*, San Diego, CA, July 1990.

43. C. Q. Shu and Y. Q. Shi, "On unified optical flow field," *Pattern Recognition*, vol. 24, no. 6, pp. 579-586, June, 1991.

44. A. Singh. "An estimation-theoretic framework for image-flow computation," *Proceedings of the 3rd International Conference on Computer Vision*, Osaka. Japan, Dec. 4-6 1990.

45. Ajit Singh "Incremental Estimation of Image-Flow Using a Kalman Filter," *Proceeding of IEEE Workshop on Visual Motion* pp. 36-43, Princeton, Oct. 7-9, 1991.

46. Ajit Singh, Optic Flow Computation: A Unified Perspective, *IEEE Computer Society Press*, 1992.

47. D. E. Dudgeon and R. M. Mersereau, Multidimensional Digital Signal Processing, *Prentice Hall , Inc.*, Englewood Cliffs, New Jersey, 1984.

48. Granino A. Korn and Theresa M. Korn, Mathematical Handbook for Scientists and Engineers, pp. 118, *McGraw-Hill Book Company*, 1968.

49. Y. Q. Shi, C. Q. Shu, J. N. Pan, "Unified optical flow field approach to motion analysis," (Pattern Recognition, Accepted, March, 1994).

50. Y. Q. Shi, C. Q. Shu and J. N. Pan, "Unified optical flow field approach to motion analysis from a sequence of stereo images," *Proceedings of the 8 th IEEE Workshop on Image and Multidimensional Signal Processing*, pp. 230-231, Cannes, France, Sept. 8-10, 1993.

51. R. G. Brown and P. Y. C. Hwang, Introduction to Random Signals and Applied Kalman Filtering, *John Wiley & Sons, Inc.*, Second edition.

52. A. Mitiche, "On combining stereopsis and kineopsis for space perception," *Proceedings of the First Conf. on Artificial Intell igence*, (Denver, CO), pp. 156-160, Dec. 1984.

53. W. Richards, "Structure from stereo and motion," *J. Opt. Soc. Amer.*, vol. 2, pp. 343-349, Feb. 1985.

54. H. C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," *Proc. Roy. Soc.London Ser.* B 208, 1980, 358-397.

55. A. R. Bruss and B. K. P. Horn, "Passive navigation," *Computer Vision, Graphics, and Image Processing*, **21**, 3-29 pp. 3-20, 1983.

56. J. Heel and S. Negahdaripour, "Time-sequential structure and motion estimation without optical flow," *Proceedings of SPIE Vol 1260 Sensing and Reconstruction of Three-Dimensional Objects and Scenes*, pp. 50-61, Feb. 1990.

57. G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligent*, vol. PAMI-7, no. 4, pp. 384-401, July 1985.

58. J. N. Pan, Y. Q. Shi and C. Q. Shu, "A correlation-feedback approach to optical flow determination," *IEEE 1994 International Symposium on Circuits and Systems*, May 1994.