High Precision Image Matching and Shape Recovery

Yalin Xiong

CMU-RI-TR-95-35

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Robotics

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

September 1, 1995

© 1995 by Yalin Xiong. All rights reserved.

This research was sponsored in part by the USAF and NSF. The views and conclusions containted in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of USAF, NSF, or the U.S. Government.

Carnegie Mellon

Robotics

Thesis

High Precision Image Matching and Shape Recovery

Yalin Xiong

Submitted in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in the field of Robotics

ACCEPTED:

Steven A. Shafer — Steven A. Shafer, Thesis Committee Chair — 3 October 1995 — Date

Matthew T. Mason — Matthew T. Mason, Program Chair — October 4, 1995 — Date

Raj Reddy — Raj Reddy, Dean — 10/14/95 — Date

APPROVED:

Paul Christiano — Paul Christiano, Provost — 10 October 1995 — Date

Contents

1 Introduction **1**

1.1 Image Matching 2

1.1.1 Matching Primitives 6

1.1.2 From Gabor Filter to Moment Filter to Hypergeometric Filter 8

1.2 Depth from Focus 9

1.2.1 Focus Matching 9

1.2.2 Depth from Focus Matching 12

1.3 Depth from Stereo 13

1.3.1 Stereo Matching 13

1.3.2 Depth from Stereo Matching 14

1.4 Structure from Motion 15

1.4.1 Optical Flow 15

1.4.2 Structure from Optical Flow 16

1.5 Structure of the Thesis 18

2 Gabor Filter Approach **20**

2.1 Introduction 20

2.2 Related Research 21

2.3 Variable Window Gabor Filters 22

2.4 Stability of Filtering 25

2.4.1 Window Contamination 25

2.4.2 Noise Contamination 31

2.5 Applications 32

2.5.1 Focus Quality Measurement 32

2.5.2 Optical Flow 34

2.6 Experimental Results 35

2.6.1 Elimination of Unstable Information 35
2.6.2 Fixed Window versus Variable Window 39
2.7 Summary 43

3 Moment Filter Approach **44**
3.1 Introduction 44
3.2 Theory of Moment Filters 46
3.2.1 Definitions 46
3.2.2 Decomposition by Moment Filters 49
3.3 Other Important Properties of Moment Filters 50
3.4 Moment Filters for Focus 51
3.4.1 Problem Definition 51
3.4.2 Finite Window Problem 53
3.4.3 Shift Variance Problem 56
3.4.4 Error Estimation 58
3.5 Moment Filters For Stereo 60
3.5.1 Problem Definition 60
3.5.2 Finite Window Problem 60
3.5.3 Relations to the Phase-Based Method 62
3.5.4 Shift Variance Problem 63
3.6 Moment Filters for the General Image Matching Problem 65
3.7 New Interpretations of Window Schemes 66
3.8 Experiments 67
3.8.1 Sampling of the Frequency Domain 67
3.8.2 Focus 70
3.8.3 Stereo 72
3.9 Summary 84

4 Hypergeometric Filter Approach **88**
4.1 Introduction 88
4.2 Hypergeometric Filters 90
4.2.1 Definitions 90
4.2.2 Recursiveness of Hypergeometric Filters 92
4.2.3 Other Properties of Hypergeometric Filters 93
4.3 Translational Matching 94
4.4 Affine Matching 99

4.5 General Matching Problem 101
4.6 Comparisons of Moment Filters and Hypergeometric Filters 102
4.7 Implementation Issues and Experiments 103
4.7.1 Computation of H Filters 103
4.7.2 Experiments on Focus 104
4.7.3 Experiments on Stereo 108
4.7.4 Experiments on Optical Flow 113
4.7.5 Experiments on Affine Matching 126
4.8 Summary 129

5 Dense Structure from Optical Flow Sequence 132
5.1 Introduction 132
5.2 System Overview 134
5.2.1 Coordinates and Motion 134
5.2.2 Block Diagram of the System 135
5.3 Initial Motion Estimation 136
5.3.1 Dynamic Motion Parameterization 137
5.4 EKF-based Uncertainty Update 138
5.4.1 Decomposition of Independent and Correlated Uncertainty . . 141
5.4.2 Weighted Principal Component Analysis 143
5.5 Interpolation and Forward Transformation 146
5.6 Implementation Issues and Experiments 151
5.6.1 Implementation Issues 151
5.6.2 Experiments 152
5.7 Summary 168

6 Summary, Major Contributions, and Future Directions 170
6.1 Summary 170
6.2 Major Contributions 171
6.3 Future Directions 173

A Stability Criteria for an Arbitrary Signal 175

B Orthogonality and Completeness of Moment Filters 178

C Properties of Moment Filters **182**
C.1 Recursive Properties of Moment Filters 182
C.2 Relations to Instantaneous Frequency and Stability Criterion 183

D Error Estimations in Moment Filter Approach **185**
D.1 Focus 185
D.2 Stereo 186

E Properties of Hypergeometric Filters **188**

F Sherman-Morrison-Woodbury Inversion **191**

G Eigen Analysis of Symmetric Outer Products **193**

H Kalman Filtering for Large Covariance Matrices **195**

List of Figures

1.1 The General Image Matching Problem 2
1.2 Window Effects 4
1.3 Image Deformation Caused by Shift Variance Effects 5
1.4 Window Effects in Focus 10
1.5 Depth from Focus 12
1.6 Depth from Stereo 15
1.7 Structure from Motion 16

2.1 Variable Window Gabor Filters 23
2.2 Oriented 2D Gabor Filter 24
2.3 The Set of 2D Variable Window Gabor Filters 26
2.4 Stability Analysis 28
2.5 Stability Criterion 30
2.6 Matching by Gabor Filters 33
2.7 Elimination of Unstable Amplitude Information 35
2.8 Elimination of Unstable Phase Information 36
2.9 Two Images Taken under Different Lens Apertures 37
2.10 Difference of Focus Quality Using Stability Threshold 38
2.11 Difference of Focus Quality Using Amplitude Threshold 38
2.12 Two Images of a Ball 39
2.13 Optical Flow Using Stability Threshold 40
2.14 Optical Flow Using Amplitude Threshold 40
2.15 Nearly Focused Image Pair 41
2.16 Defocused Image Pair 42
2.17 Focus Quality Difference of the Nearly Focused Images 42
2.18 Focus Quality Difference of the Defocused Images 43

3.1 The Pass Band Moves as i Increases 47

3.2 The Moment Filters in Spatial Domain when $f_0 = 0$ 48
3.3 The Moment Filters in Spatial Domain when $f_0 \neq 0$ 49
3.4 Polynomial Fitting of the Transformation Curve 55
3.5 Modelling Window Effects 65
3.6 Modelling Shift Variance Effects 66
3.7 Frequency Sampling 69
3.8 Synthetic Texture Images for Focus 71
3.9 Computed Blurring Difference of the First Pair 73
3.10 Computed Blurring Difference of the Second Pair 74
3.11 Computed Surface Slope of the Second Pair 75
3.12 Real Image Pair of a Toy House 75
3.13 Blurring Difference of House from SA and MFF1 75
3.14 Blurring Difference and Slope of House from MFF2 76
3.15 Images for Synthetic Experiments 77
3.16 Disparity from the Nonuniformly Shifted Pair 79
3.17 Slope of the Texture Pair Computed by MFS2 80
3.18 Disparity Maps of the Tree Pair from KOA and MFS1 80
3.19 Disparity and Slope of the Tree Pair from MFS2 81
3.20 Stereo Images of a Textured Cube 81
3.21 Disparity Map of the Cube from KOA and MFS1 82
3.22 Disparity and Slope of the Cube from MFS2 83
3.23 Stereo Images of a Toy House 84
3.24 Disparity Map of the House from KOA and MFS1 85
3.25 Disparity and Slope of the House from MFS2 86

4.1 The Passbands of Hypergeometric Filters 91
4.2 H Filters and Their Closest Gabor Filters 92
4.3 A Two-Dimensional Hypergeometric Filter 93
4.4 Uncertainty in the Spatial and Frequency Domains 95
4.5 Product of the Spatial and Frequency Uncertainty 96
4.6 Frequency Sampling in STFT and H Filters 96
4.7 Two-Step Decomposition of an Affine Transform 101
4.8 Frequency Sampling in Our Implementation 104
4.9 Computed σ_0^2 from the First Pair 105
4.10 Computed σ_0^2 from the Second Pair 106

4.11 Slope Computed by HFF2 for the Second Pair 106
4.12 Blurring Difference from HFF1 for the House Pair 107
4.13 Blurring Difference from HFF2 for the House Pair 107
4.14 Slope for the House Pair 108
4.15 Foreshortening Estimated by HFS2 109
4.16 Disparity of the Cube Pair from HFS1 110
4.17 Disparity and Slope of the Cube Pair from HFS2 111
4.18 Disparity of the House Pair from HFS1 111
4.19 Disparity and Slope of the House Pair from HFS2 112
4.20 Yosemite Image Sequence 114
4.21 Computed Optical Flow of the Yosemite Scene 114
4.22 Average Error for the Yosemite Scene 116
4.23 Uncertainty Estimation for the Yosemite Scene 117
4.24 Translating Tree Image Sequence 118
4.25 Computed Optical Flow of the Translating Tree Scene 118
4.26 Average Error for the Translating Tree Scene 119
4.27 Uncertainty Estimation in Translating Tree Scene 120
4.28 The Diverging Tree Image Sequence 120
4.29 Optical Flow of the Taxi Sequence 122
4.30 Optical Flow of the NASA Coke Sequence 123
4.31 Optical Flow of the SRI Tree Sequence 124
4.32 Optical Flow of the Rubic Sequence 125
4.33 Image before and after Affine Transform 126
4.34 True Gradients of Optical Flow 127
4.35 Recovered (u_0, v_0) from Diverging Tree Images 128
4.36 Recovered Gradient Parameters from Diverging Tree Images 129
4.37 Images of a Textured Cube 130
4.38 Recovered (u_0, v_0) from the Cube Images 130
4.39 Recovered Gradient Parameters from the Cube Images 131

5.1 The Camera Coordinate 134
5.2 Block Diagram of the System 136
5.3 An Image Sequence of a Toy House 145
5.4 Six Non-Weighted Principal Components 147
5.5 Six Weighted Principal Components 148

5.6 Vector Field as a Correlated Uncertainty 149
5.7 One Frame in a Straw Hat Sequence 155
5.8 Three Eigenimages for the Straw Hat Sequence 156
5.9 Evolutions of Straw Hat Uncertainties 157
5.10 A Road Sequence 158
5.11 The First Three Eigenimages of the Road Sequence 159
5.12 Evolutions of Road Uncertainties 160
5.13 The Straw Hat Sequence 162
5.14 Sketch of True Shape 163
5.15 Evolution of the Shape of Straw Hat 163
5.16 Measuring Quality of Results 164
5.17 The Chair Sequence 165
5.18 The Cube Sequence 166
5.19 The Basket Sequence 167
5.20 The Sphere and Dog Sequence 169

List of Tables

2.1 Sampling of the Frequency Domain by Gabor Filters 25
2.2 RMS Errors of Different Window Schemes 42

3.1 RMS Errors of Different Algorithms 71
3.2 RMS Errors of Disparity from Synthetic Images 78
3.3 RMS Errors of the Cube 79
3.4 RMS Errors of the Toy House 84

4.1 RMS Errors of Hypergeometric Filter Approach 105
4.2 RMS Errors for Synthetic Images 109
4.3 RMS Errors of the Cube 110
4.4 RMS Errors of the Toy House 112
4.5 Density vs. Average Error for the Yosemite Sequence 115
4.6 Density vs. Average Error of the Translating Tree Scene 119
4.7 Average Error of the Diverging Tree Sequence 121
4.8 Recovered Affine Parameters from a Small Patch 127
4.9 Comparison under Translation and Affine Model 128

Abstract

In this thesis, we examine several depth and shape recovery methods from a methodological point of view. Specifically, we concentrate our efforts on approaches which require only very limited and realistic assumptions. Those approaches are: depth from stereo, depth from focus, and structure from motion. They all start with image matching as the first step, and then map the matching parameters into depth or shape information.

At first, we examine the traditional Gabor filter-based approach and propose to use a general criterion to detect situations where the filter outputs are severely contaminated by window effects. By eliminating the contaminated information, we can improve the precision of image matching. Then we go further to analyze the originations of window effects and propose "moment" filters. Due to the recursiveness of moment filters in the spatial and Fourier domains, we can directly model window effects and foreshortening effects analytically, and achieve much higher precision than with traditional approaches. Finally, we go beyond moment filters to propose "hypergeometric" filters which, in addition to modelling window and foreshortening effects, sample the frequency domain canonically.

We will also look into the classic problem of structure from motion, which recovers 3D structural information from optical flows resulting from unknown camera motions. We will describe a structure-from-motion system based on the Extended Kalman Filter (EKF), which is capable of *incrementally* generating 3D *dense* depth maps from an optical flow sequence. Most traditional feature-based approaches cannot be extended to compute dense structure due to the impractical computational complexity. We demonstrate that, by decomposing uncertainty information into independent and correlated parts, we can decrease the complexity from $O(N^2)$ to $O(N)$, where N is the number of pixels in the images. We also show that this dense structure-from-motion system requires only local optical flows, i.e., image matchings between two adjacent frames, instead of the tracking of features over a long sequence of frames. We will demonstrate such a system on real image sequences.

Acknowledgments

Thanks to my advisor, Steve Shafer. He took me into his exciting research group when my first advisor left Carnegie Mellon University. Under his supervision, I started my first project on computer vision in the advanced perception class. During the past four years, he gave me enough freedom to pursue whatever interested me and guided me when I got lost. I have learned many valuable things from him about research and integrity.

Thanks to other committee members: Martial Hebert, Katsushi Ikeuchi, and Narendra Ahuja, all of whom provided many helpful suggestions to improve my thesis. I would also like to thank many other faculty members in the Robotics Institute, with whom I had many interesting and inspiring discussions. They are Takeo Kanade, Yangsheng Xu, Eric Krotkov, Norman Sadeh, Mark Fox, and Katia Sycara.

The Robotics Institute in CMU is a world-class place. There is a group of young, energetic and dedicated graduate students, many of whom I have had the luck to befriend. Thank you to Mark Maimone, Casper Cheng, JyiShane Liu, Yan-Bin Jia, Dajun Zeng, Kan Deng, Bruce Maxwell, David Simon, Julio Rosenblatt, Lei Tang, Yoichi Sato, Fred Solomon, Marcel Bergerman, Conrad Poleman, Ping Fu and Joyoni Dey. For technical and facility help I thank Reg Willson, John Krumm, Keith Gremban, Jim Moody, Kate Fissell and Bill Ross.

I also received much help from general staff members in Robotics Institute. Thanks to Marce Zaragoza, Patty Mackiewicz, Jim Kocher, Carolyn Kraft and Marie Elm.

Thanks to my wife Belinda, who has accompanied me through difficult time and kept me happy. I am deeply indebted to her for her unconditional sacrifice for our new family and unwavering support during my stay at CMU. Thank you to my parents-in-law for raising such a wonderful daughter, and to sister-in-law for much financial help.

Special thanks to my parents who managed to provide me a good education under an extremely harsh environment. Thank you to my brother who served as my role model. Thank you to my sister who took good care of me when I was a baby and my parents could not afford a babysitter.

Chapter 1

Introduction

Three-dimensional shape recovery based on image matching is one of the most important problems in computer vision because it provides vital information for many higher level tasks such as navigation, manipulation, recognition, etc. High precision is desired to enable a robot to perform those tasks robustly. Unfortunately there are error sources in image matching which are ignored by most algorithms, and computation problems in mapping the results from image matching into 3D shape. In this thesis, we present a systematic approach to the image matching problem in order to achieve high precision, and a new eigen-based uncertainty representation to deal with the computation problem arising from structure from motion.

First, we examine several depth and shape recovery methods from a methodological point of view. Specifically, we concentrate our efforts on approaches which require only very limited and realistic assumptions. Those approaches are: depth from stereo, depth from focus, and structure from motion. They all start with image matching as the first step, and then map the matching parameters into depth or shape information. The thesis demonstrates an evolution of ideas in pursuing high precision in image matching.

For the image matching problems, we identify two sources of error: window effects and shift variance effects. In Chapter 2, we examine the the traditional Gabor filter-based approach and propose a general criterion to detect where the filter outputs are significantly contaminated by window effects. We show that by eliminating that contaminated information, we can improve the precision of image matching. In Chapter 3, we go further to analyze the origination of window effects and propose "moment" filters. Due to the recursiveness of moment filters in the spatial and Fourier domains,

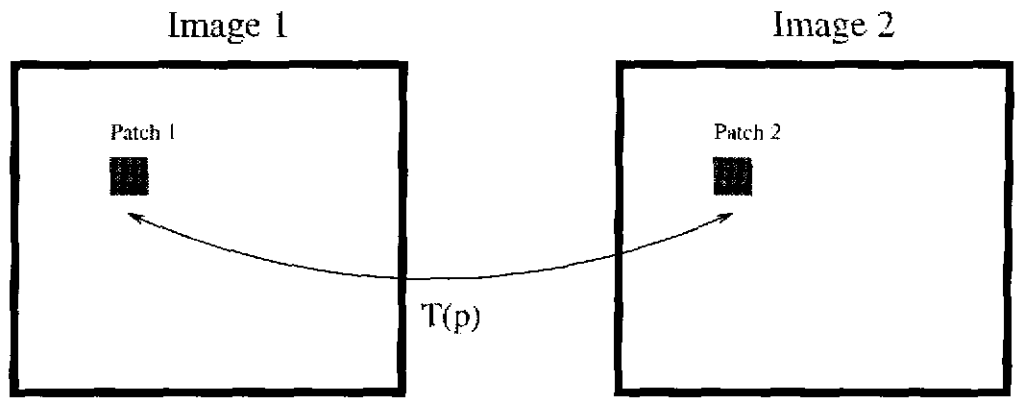


Figure 1.1: The General Image Matching Problem

we can directly take window effects and foreshortening effects into consideration, and achieve much higher precision than with traditional approaches. In Chapter 4, we go beyond moment filters to propose "hypergeometric" filters which, in addition to modelling window and foreshortening effects, sample the frequency domain canonically.

For the shape recovery problem based on image matching, we look into the classical problem of structure from motion, which recovers 3D structural information from optical flows resulting from unknown camera motions. We identify the representation and manipulation of uncertainty information as the major problem in a dense structure from motion system. In Chapter 5, we describe a structure-from-motion system based on the Extended Kalman Filter (EKF), which is capable of *incrementally* generating 3D *dense* depth maps from an optical flow sequence. We demonstrate such a system on real image sequences.

1.1 Image Matching

An abstract image matching problem can be formulated as extracting one or more parameters of a *known* type of transformation between two *finite* image patches as in Figure 1.1. The known transformation could be, for example, spatial translation for stereo or image blurring for focus. And the parameters we are trying to extract are those of the transformation, such as how much one image patch is translated or blurred with respect to the other.

Let us formulate the image matching problem mathematically as follows: $i_0(x)$ and $i_1(x)$ are the two images in the spatial domain, $I_0(f)$ and $I_1(f)$ are the two images in

the Fourier domain, and the transformation between the two images is $T(f;p)$ where p is the matching parameter to be computed. Therefore, the relation between $I_1(f)$ and $I_0(f)$ is

$$I_1(f) = I_0(f)T(f;p). \tag{1.1}$$

Ideally, if we can obtain $I_0(f)$ and $I_1(f)$, the parameter p could be computed by fitting $T(f;p) = I_1(f)/I_0(f)$. Unfortunately $I_0(f)$ and $I_1(f)$ can never be obtained using finite-duration filters or windows; therefore, there is no way to directly use the above equation to compute p without using infinitely large windows. In fact, one of the most important questions we try to answer in this thesis is: *How can we compute p precisely given only finite-duration signals as in Figure 1.1?*

The usual way of using information from a finite-duration signal is by imposing a window or a filter with finite effective width. Suppose $w(x)$ and $W(f)$ is a filter in the spatial and Fourier domains respectively, what we obtain by convolving finite-duration filters with images are $I_0'(f)$ and $I_1'(f)$ defined as

$$I_0'(f) = \int_{-\infty}^{+\infty} I_0(f_0)W(f-f_0)df_0, \tag{1.2}$$

$$\begin{aligned} I_1'(f) &= \int_{-\infty}^{+\infty} I_1(f_0)W(f-f_0)df_0 \\ &= \int_{-\infty}^{+\infty} I_0(f_0)T(f_0;p)W(f-f_0)df_0. \end{aligned} \tag{1.3}$$

Unfortunately, unless $T(f,p)$ is a constant within the passband of $W(f_0-f)$, generally $I_1'(f)/I_0'(f) \neq I_1(f)/I_0(f)$, i.e., the original relation in Eq. 1.1 does not hold true for windowed filter outputs.

One of the major difficulties in obtaining high precision results in image matching is that the known transformation is usually specified without considering the finite size of image patches. Figure 1.2 illustrates the situation in which the two underlying signals are indeed shifted versions of each other, but the two windowed signals are not. This is usually referred to as the "window effect". In the Fourier domain, the window effect is that $I_1'(f)/I_0'(f) \neq I_1(f)/I_0(f)$ unless $T(f;p)$ is a constant within the pass band (Figure 1.2) of the filter as in Eq. 1.2 and Eq. 1.3. Therefore, if the passband of the filter is the interval around f_0 as illustrated in Figure 1.2, ignoring window effects is equivalent to approximating the transformation curve in the passband as a constant.

Another difficulty is that the parameters may not be constant at all within the window width. In other words, the transformation between the two image patches

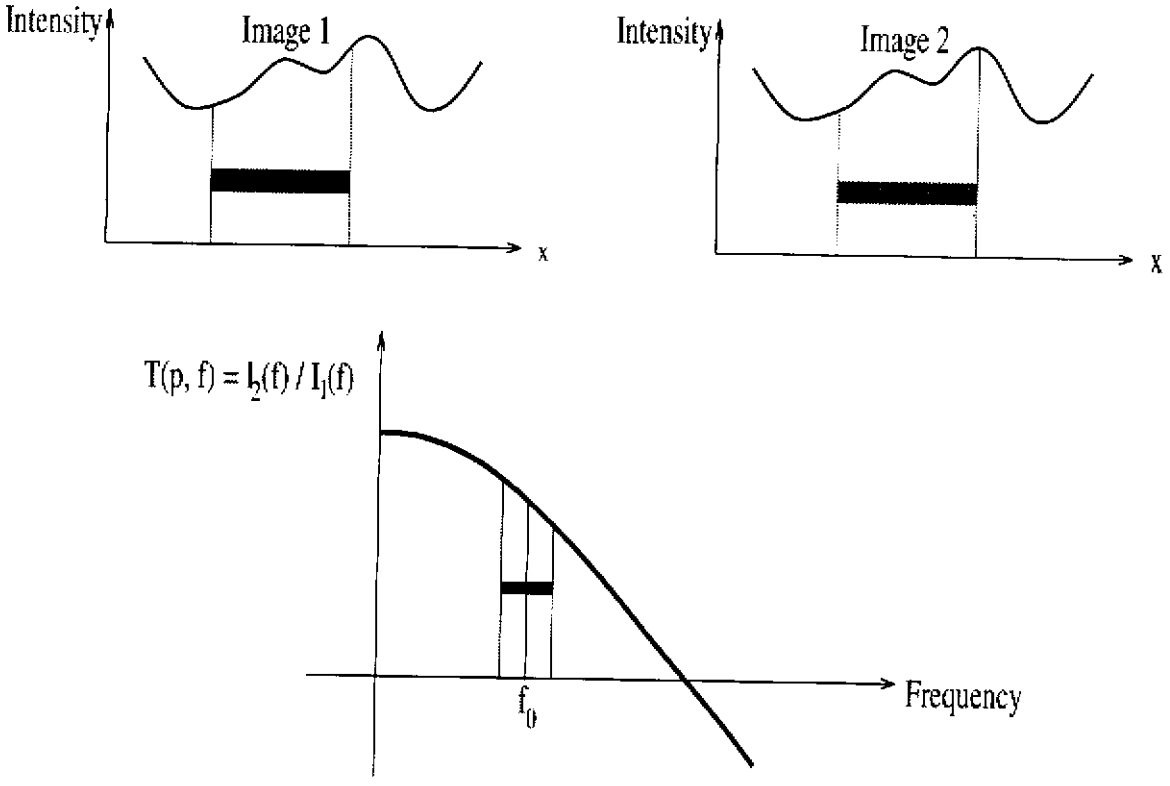


Figure 1.2: Window Effects

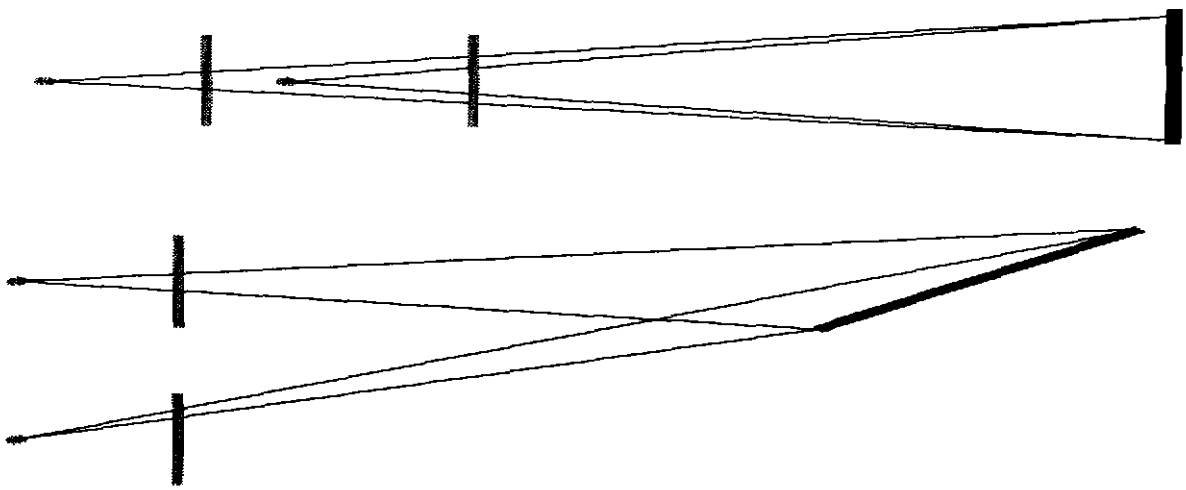


Figure 1.3: Image Deformation Caused by Shift Variance Effects

could be shift-variant. For example, if the transformation is spatial translation, and the magnitudes and directions of the translations are different at different locations, then the original image is deformed in that some parts of the image are stretched while others are compressed. In stereo, this is usually referred to as "foreshortening" effects. Figure 1.3 shows two examples of image deformation caused by shift variance effects.

The image matching problem is usually formulated as minimizing an objective function. There have been two major strategies in finding the matching parameters: the "search" method and the "direct" method. The search method samples the parameter space, artificially transforms one image patch using the parameter samples, and finds the sample which minimizes the discrepancy between the artificially transformed image patch and the other real image patch. The search method is often used when the derivatives of the objective function with respect to the matching parameters are not available or are hard to compute. Additionally if the sampling of the parameter space is very dense, the window effects could be minimal. The direct method, on the other hand, uses the information in the two image patches to directly compute the parameters. Usually the direct method requires a reasonably good initial estimation of the parameters, and the derivatives of the objective function with respect to the matching parameters are used. The approaches we will propose in the thesis are all direct methods.

In this thesis, we will first examine the traditional Gabor filter based direct methods in focus matching and optical flow. The window effects will be shown to cause

instability and errors. We then propose to use "moment" and "hypergeometric" filters to solve the general image matching problem. Due to the unique recursiveness of these filters in the spatial and frequency domains, the window effects are represented in the filter space as high order terms in Taylor expansions, and the foreshortening effects are represented as linear modifications to the original image. When both effects are taken into consideration in computing matching parameters, we achieve very high precision.

1.1.1 Matching Primitives

The first question in designing a matching algorithm is what kind of primitives we should use in order to obtain higher precision. In other words, how can we represent the image information contained in a image patch as a vector of matching primitives such that it benefits the computation of matching parameters? If the length of the vector is N, the goal of matching is to find the matching parameters which minimize the distance between the two matching vectors in the N-dimensional space.

In computer vision, the following primitives have been used for image matching:

1. *Intensity Values*: Using the intensity values of pixels in the window as matching primitives is simple and usually fast. The dimension N is usually large. But since the intensity values do not make frequency information explicit, they are less flexible than the filter space primitives but have the advantage of less computations.

2. *Filter Space Primitives*: We can also convert an image patch into the filter space by convolving it with a set of filters. The matching vector is composed of outputs from the convolutions. Specifically, the finite-duration discrete Fourier transform of the image patch is an example of filter space representation. Since the filters usually have one single pass band, the information at different frequency bands is represented separately in the matching primitives. Additionally, many filter space primitives provide derivative information with respect to the matching parameters. Thus they can usually be used in direct methods.

3. *Features*: The feature-based methods classify the information in the image patches as either non-feature or one specific type of feature, e.g., edge. Though more information about a feature such as its orientation or sharpness could also be

used for matching purposes, the feature-based methods compress the image information significantly, i.e., the dimension of the matching vector is usually only one or two. Since most image patches would be classified as non-features, the feature-based methods usually obtain only sparse matching information.

In this thesis, we will consider only intensity value primitives and filter space primitives. In fact, in some cases the two representations are theoretically the same because of the duality between the spatial and Fourier domains. For example, the sum of the square difference (SSD) of intensity values is exactly the same as the sum of the norm difference of Fourier coefficients. On the other hand, the intensity value primitives could also be regarded as a special type of filter space primitives where the filters are delta functions.

Considering the transformation $T(f;p)$ in Eq. 1.1 as an operator which operates on $I_0(f)$ as $\mathbf{T}_p[I_0(f)] = I_1(f)$, we can see that apparently delta functions in the Fourier domain are the eigenfunctions of this operator, i.e.,

$$\mathbf{T}_p[\delta(f_0)] = T(f_0;p)\delta(f_0) = c\delta(f_0). \tag{1.4}$$

In other words, when an eigenfunction is transformed by the operator, it is mapped to itself multiplied by a constant. Since a delta function in the Fourier domain is an infinitely long sinusoid in the spatial domain, the eigenfunction of the above operator is the conventional sinusoid. Unfortunately there is no similar set of eigenfunctions which have finite durations. In other words, it is impossible to find a set of finite duration filters which span the whole functional space *and* are mapped to themselves by the operator $\mathbf{T}$.

From this point of view, the research in representing image information in different filter spaces could be interpreted as finding a set of finite duration filters which have approximately the same properties as the eigenfunctions. Gabor filters are one example of such approximation. But as we will show, this approximation is a major source of computation errors. In this thesis we will demonstrate that both the moment and hypergeometric filters proposed here have very similar properties to the eigenfunctions of the operator. They are orthogonal and can be used to decompose a signal. And instead of the self mapping as in Eq. 1.4, these novel filters are mapped by the operator $\mathbf{T}$ as a linear combination of themselves, i.e.,

$$\mathbf{T}_p[H_i(f)] = \sum_{j=0}^{+\infty} c_{ij} H_{i+j}(f). \tag{1.5}$$

We can then use the above equation to replace Eq. 1.4 in measuring the similarity of the two image patches.

1.1.2 From Gabor Filter to Moment Filter to Hypergeometric Filter

The most commonly used filters to represent image information as filter space primitives are Gabor filters. Usually the peak frequencies of the Gabor filters are arranged such that their effective bandwidths do not overlap much, i.e., the information extracted from different Gabor filters is not redundant. When Gabor filters are used for matching between two images, the usual scheme is to compare the filter outputs of two images at the same frequency band to estimate the matching parameter. But since Gabor filters, like any other finite-duration filters, have non-zero passbands, they incur the window effects as explained in the previous section. The precision of the estimated matching parameter is then damaged. Though the window effects have been noticed to cause large errors as in [27, 22], there has been no effective way proposed in the literature to correct the errors.

There is also what we refer to as the "shift variance" effects which are harmful to high precision image matching. The shift variance effects are caused by the fact that the matching parameter is not a constant within the filter width. In stereo, it is also referred to as foreshortening effects. The available approaches to this type of image deformation problems were mostly based on image warping which is extremely expensive since it involves exhaustive high dimension search.

Based on Gabor filters, we propose in this thesis a new type of filters referred to as "moment" filters. The moment filters have some unique mathematical properties which relate them to each other while Gabor filters are independent of each other. These unique mathematical properties enable the moment filters to model both window and shift variance effects analytically. Once those effects are considered in the image matching, we can achieve higher precision.

But for both Gabor and moment filters, we need to sample the frequency domain manually if we need a set of filters in order to use all the information in an image. The dilemma is that we can sample it neither too densely nor too sparsely. If we sample too densely, the information contained in different frequency bands is redundant which brings difficulties in combining results from different frequency bands. But if we sample too sparsely, we abandon some information in the image. This frequency

sampling problem was never explicitly addressed in the computer vision literature. In order to solve this dilemma, we propose another type of filters referred as "hypergeometric" filters based on moment filters. The hypergeometric filters have the same mathematical properties as the moment filters such that the window and shift variance effects can also been modelled by hypergeometric filters. In addition, the set of hypergeometric filters is a complete and orthogonal functional basis which solves the frequency sampling problem.

1.2 Depth from Focus

1.2.1 Focus Matching

The focus matching problem is one example of the general matching problem in Eq. 1.1 with the Gaussian blurring model

$$T(f;\sigma_0) = e^{-f^2\sigma_0^2/2}, \tag{1.6}$$

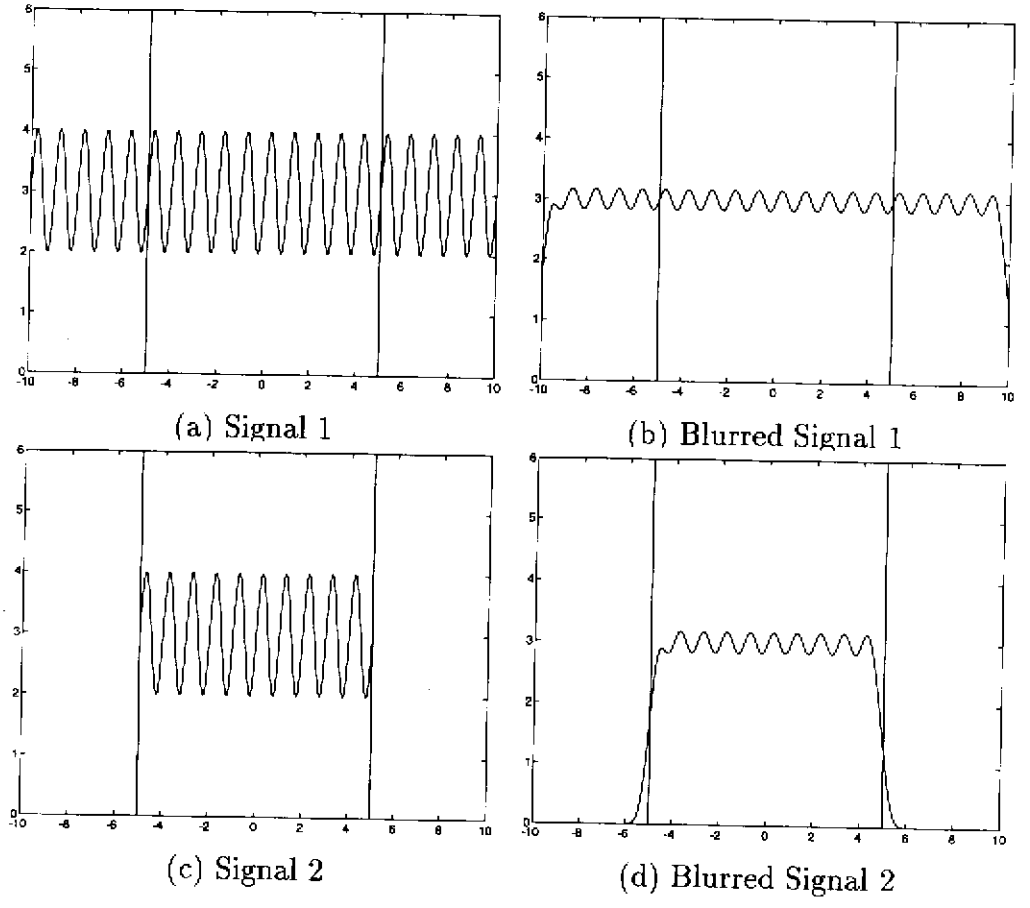
or the pill-box blurring model [12]

$$T(f;\sigma_0) = 2\frac{J_1(f\sigma_0)}{f\sigma_0}, \tag{1.7}$$

where $J_1(x)$ is a Bessel function of the first order.

Figure 1.4 illustrates the window effects in focus. The window is between the two vertical lines ($[-5,5]$) in Figure 1.4. Even though the windowed versions of Signal 1 and 2 are identical, their windowed blurred versions within the two vertical lines are very different near the window edges. If we perform finite-duration Fourier transform for the windowed signals, Signal 2 in (c) and blurred Signal 2 in (d) do not hold the original relation in Eq. 1.6. The same effects happen no matter what kind of shape the window has.

Computing depth from defocus was first proposed by Pentland in [69], where he used Laplacian filters to estimate high-frequency content in images; by comparing the high-frequency contents from two images, he could compute the blurring difference represented by σ_0. From the Gaussian law of lenses, the blurring difference could then be mapped to absolute depth. Though the computation was simple and fast as demonstrated in [68], the precision was very low due to the wide pass-bands of the Laplacian filters.



(a) Signal 1 (b) Blurred Signal 1

(c) Signal 2 (d) Blurred Signal 2

Figure 1.4: Window Effects in Focus

To improve the precision in focus matching, Subbarao [82, 84] and Bove [12, 13] proposed to compute σ_0 using Fourier coefficients by fitting a curve to all frequency bands in order to minimize errors. They still assumed that Eq. 1.1 was valid for individual frequency bands as long as their magnitudes were much higher than the noise level in the image. Compared to the Laplacian filter based approach, the FFT-based approaches did improve precision due to narrower pass bands and the larger number of filters. But all the FFT-based approaches are still suffering from the same problem, i.e., the window effects caused by the non-zero pass band of filters.

Having realized the limitations of the FFT-based approaches, Ens and Lawrence [21], Subbarao and Surya [83] explored the possibility of directly using image intensity values to compute the blurring difference. Ens and Lawrence proposed a matrix-based solution which was similar to the traditional image deconvolution. But since image deconvolution is unstable in the presence of noise, they had to impose some artificial constraints, e.g., smoothness constraints, in order to obtain stable deconvolution. In the meantime, Subbarao and Surya proposed to use spatial image moments to compute the blurring difference. Because they assumed that the local image patches could be represented by a low order polynomial, a small number of moments should have been able to represent the local image information satisfactorily. Even though both matrix-based and moment-based spatial approaches did report reasonably good results, i.e., the relative RMS depth error was 1.7% at distance of 0.9 meter reported in [22] and 2.3% at distance of 0.6 meter reported in [83], and were relatively faster than the traditional FFT-based approaches, the precision of results were basically in the same range because the window effects which haunted the FFT-based approaches reappeared in another form.

In retrospect, it is somewhat odd that very few people had suggested search methods, i.e., searching the space of σ_0 and finding the one value of σ_0 which minimizes the difference between two image patches. A possible explanation for the lack of search methods is that the artificial transformation in a search method may be too expensive for depth from defocus. In fact, Ens and Lawrence in [22, 20] actually formulated a SSD-based objective function for the focus matching problem, although they still intended to solve the objective function directly instead of by searching the parameter space. Recently Xiong and Shafer [98, 99] combined the FFT-based approach and the search approach to minimize the number of artificial transformations while using the power of the FFT-based approach to estimate σ_0. The proposed "maximal resemblance" method alternatively used a FFT-based approach to estimate σ_0 and

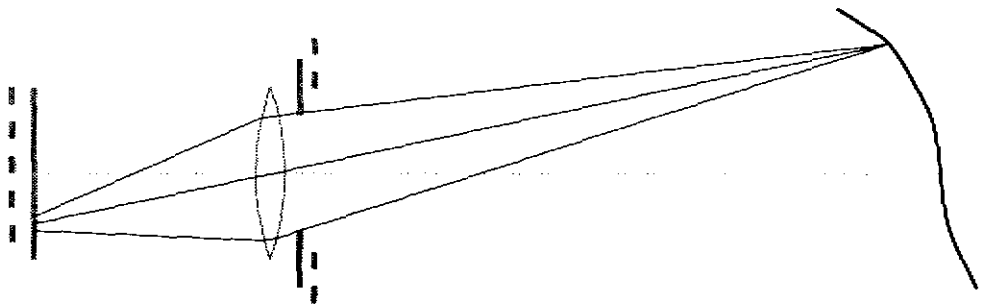


Figure 1.5: Depth from Focus

artificially blurred the image by using estimated σ_0. Note that if the search space is sampled densely enough, the window effects could be very small. The maximal resemblance minimized the window effects by artificial blurring, and avoided the exhaustive search using the FFT-based method to predict the blurring difference. The relative RMS depth error of 0.5% at about 3.0 meter was reported in [99].

The progress in focus matching has been from frequency domain approaches to spatial domain approaches, from direct methods to search methods. Despite all of the efforts, the fundamental limitation of window effects was never fully solved in previous approaches. By using the moment filters or hypergeometric filters, the precision is limited only by the noise level in images and the accuracy of the transformation model.

1.2.2 Depth from Focus Matching

The depth from focus method employs the relation between the focus quality and depth to compute how far a point is. As illustrated in Figure 1.5, either placing the image plane at different distance from the lens or changing the size of the aperture will results in a corresponding change in the focus quality. The two images used for matching are taken under those two different camera settings. In the case of changing aperture size for an ordinary camera or changing the image plane position for a telecentric lens camera [66], the only difference between two images is their focus quality. The task of focus matching presented in the previous section was to measure such change precisely.

The new task here is how to map the measured focus quality differences into absolute depth values. The exact modelling of the focus quality difference as a function of depth depends on the modelling of the optics. The most primitive and yet the

most powerful method is to build a lookup table by putting an object in different depths and measuring the focus quality difference under the camera settings. Such calibration data could be further simplified by fitting a model. For example, in our earlier work [98], we proposed the following as the relation between the focus quality difference and the absolute depth value:

$$\sigma_0^2 = \left(k_1 + \frac{k_2}{z + k_3}\right)^2 + k_4^2, \tag{1.8}$$

where σ_0^2 is the focus quality difference between the two images given the Gaussian blurring model, k_1, k_2, k_3 and k_4 are fitted parameters and z is the depth.

1.3 Depth from Stereo

1.3.1 Stereo Matching

The stereo matching problem is another example of the general matching problem in Eq. 1.1 with

$$T(f; d) = e^{jfd} \tag{1.9}$$

i.e., the transformation is a phase shift in the frequency domain or one dimensional translation in the spatial domain.

Stereo matching is one of the most sophisticated and mature fields in computer vision. Here we do not intend to present a thorough survey of this field, but rather to provide an interpretation of the progress in this field from the perspective of precision. Even though the ambiguity problem is also a very important driving force behind the development of various stereo matching algorithms, our work addresses the precision issue only.

Early stereo matching algorithms targeted pixel-level precision only, i.e., the disparity was always an integer multiple of inter-pixel distance. The Marr-Poggio theory [55] of stereo matching extracted zero-crossing locations as features, and used the sign and orientation information to disambiguate possible candidates. In order to avoid possible matching ambiguity caused by using low-dimensional matching primitives, the theory was later extended by imposing continuity constraints [31, 32, 57], disparity gradient limits [70], or using iterative relaxation methods [10].

One of the major motivations for using low-dimension matching primitives such as features is their low computational cost. On the other hand, high dimensional

matching primitives such as intensity values contain much richer information such that they can be used for *dense* and *subpixel* estimation, though the amount of computation is much higher. The area-based stereo matching algorithms [59, 60, 52, 5, 56, 42] use correlation between two image patches as a measurement of similarity. Usually the maximal similarity is detected using a simple exhaustive search strategy. The results are much more precise than the traditional feature-based approaches.

The equivalent approaches were based on Fourier coefficients or a dense set of filters. Though the Fourier transformation or filter convolutions incurs even more computation, the matching primitives are more flexible than the intensity values used in area-based approaches because of the explicit decomposition of image information into different frequency bands. Many techniques such as instantaneous frequency or stability tests could be used to improve precision or detect corrupted information. Jones and Malik [40, 39] use a set of filters in difference scales to incorporate information in different scales into a single matching. Sanger [76], Langley *et al* [48], Jepson and Jenkin [38], Weng [94, 91, 92], and Fleet *et al* [27, 24] propose to use phase information in Gabor filters or Fourier coefficients to directly compute disparity. It was pointed out in [24] that abandoning the magnitude information could actually be beneficial.

But both area-based approaches and filter-based approaches failed to address the window effects thoroughly. Even though some filter-based approaches indeed address the stability problem caused by window effects, there has been no attempt to correct the errors caused by them.

1.3.2 Depth from Stereo Matching

The depth from stereo method uses the relation between the disparity, i.e., the magnitude of the translation and the depth, to map a disparity map into a depth map. If the optical axes of the two cameras are parallel with each other as in Figure 1.6, and lens distortions and other nonlinearity of the cameras are ignored, the relation between disparity and depth is simply

$$z = \frac{fb}{d}, \tag{1.10}$$

where z is depth, f is the focal length, b is the baseline length, and d is the disparity. In cases where the orientations of the two cameras are skewed to each other, the above relation is much more complicated as explained in [23]. But no matter

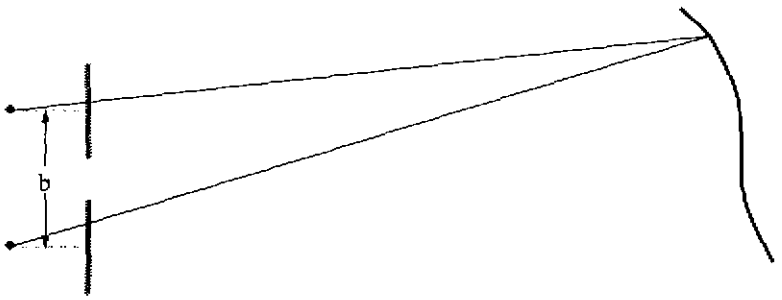


Figure 1.6: Depth from Stereo

how complicated the relation between disparity and depth is, we can always map a disparity value to a depth value through a definite mathematical function.

1.4 Structure from Motion

1.4.1 Optical Flow

The optical flow problem is another example of the general matching problem in Eq. 1.1 with

$$T(\vec{f};\vec{d}) = e^{j\vec{f}\cdot\vec{d}}, \tag{1.11}$$

which is the 2D extension of the stereo matching problem.

The major difference between traditional stereo matching and optical flow algorithms is that matching ambiguity is a much less serious problem for an optical flow because it is usually computed between close frames in an image sequence. The relative motions between these frames are small.

Besides those algorithms [6, 51, 61, 63, 90, 78, 79, 73, 26], which were direct generalizations from stereo matching algorithms, there were also some algorithms which are specific for the optical flow computation. Adelson and Bergen [2], Heeger [34] and Fleet *et al* [25] used 3D spatiotemporal filters to detect flow directions and velocities. In other words, the 2D motion problem is solved by finding orientation in 3D spatiotemporal space, assuming the motion fields are smooth in all three dimensions. Barron *et al* [11] provided excellent comparisons in methods and performance among those algorithms.

As in focus matching and stereo matching, the precision in computing optical flow is still limited by the window effects in the above mentioned approaches.

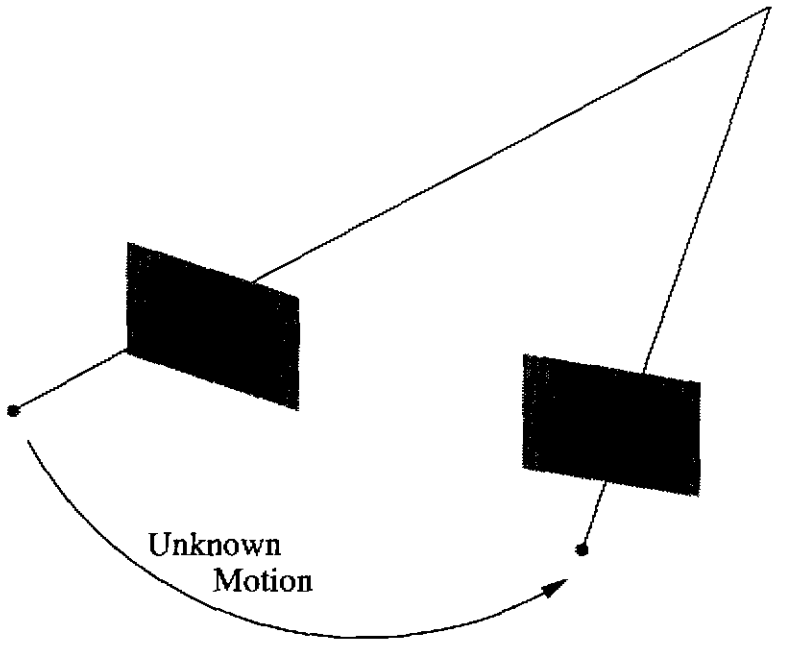


Figure 1.7: Structure from Motion

1.4.2 Structure from Optical Flow

Recovering structural information from a motion sequence is a very challenging task even after the optical flows have been computed. The goal is to use 2D image motion information to derive both the 3D structural information of the scene and the camera motion itself as illustrated in Figure 1.7. The major difficulties lie with two factors: the relation among optical flows, motion of the camera and shape of objects in the scene is nonlinear, and it is difficult to separate the motion and the shape such that they could be estimated sequentially if both of them are unknown. From this point of view, the general depth from stereo problem is a special case of structure from motion in which the motion is known.

To avoid the problem of nonlinearity, Longuet-Higgins [49], Tsai and Huang [89], Nagel [62] and Weng *et al* [95] replaced the original nonlinear constraint with a linear constraint called "E matrix", which could then be solved easily. But as pointed out in [35], since the linear constraint based on E-matrix was essentially a weaker version of the original nonlinear constraint, it introduced more structural ambiguity.

If either motion or shape is *exactly* known *a priori*, it is trivial to compute the the other. Given the 2D optical flows, the dependence of motion on shape or shape on motion could be easily specified. Since we can never recover the *exact* optical

flows, the motion and shape could only be computed to a certain precision. In other words, both motion and shape will be uncertain. And because of the inter-dependence between motion and shape, they must be correlated with each other. The correlation between motion and shape makes it difficult to separate them. In the literature, there are two ways to estimate motions and structural information simultaneously: the recursive method and the batch method. Broida *et al* [16] and Azarbayejani *et al* [7, 8] propose to use Extended Kalman Filtering to recursively estimate motions and structural information of tracked features. Examples of the batch method include [87] and [85]. Tomasi [87] factorized the 2D image motion matrix into multiplication of a shape matrix and a motion matrix, assuming orthogonal projection. Szeliski and Kang [85] tried to compute motions and structural information by directly minimizing a nonlinear constraint.

In all the above mentioned methods, including E-matrix methods, recursive methods and batch methods, features are supposed to be identified and tracked throughout the sequence. And structural information only at feature locations is recovered. In general, good features as defined in [77] are those which allow precise computation of image flow velocities. But quantifying image information as either feature or non-feature is a gross simplification because there is a spectrum of uncertainty in computing image flow velocities. Based on this observation and the fundamental difficulty of recovering dense structural information using feature-based approaches, we attempt to use dense optical flows to recover motions and structural information.

For a feature-based approach, it is a reasonable assumption that the image velocity of those features could be estimated fairly precisely. But it is unlikely that dense optical flows could be precise everywhere. Therefore, those flow-based approaches such as [71, 3, 43, 44, 58], which did not explicitly take into considerations the varying degree of uncertainty in the optical flow information cannot be applied easily to real optical flow sequence.

Our system is the first one which successfully recovers *dense* structure from image sequences by employing optical flow information of various degrees of uncertainty. The EKF-based recursive approach proposed in the thesis uses all optical flows and their uncertainty information to estimate dense 3D structure. We compress the $N \times N$ covariance matrix of the 3D structure by keeping only weighted principal components. By applying other mathematical techniques, we will show that the system is capable of delivering dense and accurate 3D structure information from an optical flow sequence.

1.5 Structure of the Thesis

The main theme of the thesis is image matching and shape recovery based on image matching. We begin our investigation of image matching based on traditional Gabor filter in Chapter 2. We demonstrate that the Gabor filter approach can be uniformly applied to focus and stereo/optical flow problems. Unfortunately, all information from Gabor filter convolutions is contaminated by window effects to various extents. Since window effects are zero when the underlying image is composed of distinct sinusoids, we use the similarity between the filter outputs and sinusoid as a measurement of how severely the information is contaminated by window effects. Therefore, we can guarantee that the information we use to compute matching parameters is only moderately contaminated by window effects.

In Chapter 3, we look further into the Gabor filter, and find that the window effects could be analytically represented as an infinite sum of moment filter outputs. Thus we can incorporate the window effects into an objective function, which yields much more precise results when minimized than does the traditional Gabor filter approach. In addition, the shift variance effects can also be represented analytically so that they can also be corrected in matching. We demonstrate the effectiveness of the moment filter approach on focus and stereo problems, and the high precision of results.

In both the Gabor filter and the moment filter approaches, the selection of filters, i.e., the sampling of the frequency domain, is arbitrary. There is no systematic way to ensure that information contained in different frequency bands is independent, nor that all information in an image can be represented by such a set of filters. In Chapter 4, we further extend the idea of moment filters to hypergeometric filters which sample the frequency domain and represent image information in a canonical fashion. Thus, the set of hypergeometric filters constructs an orthogonal and complete basis to decompose any signal. Of course, since the hypergeometric filters possess similar properties as the moment filters, both window and shift variance effects can also be analytically represented and incorporated into an objective function to compute matching parameters. We demonstrate the hypergeometric filter approach on focus, stereo, optical flow and affine matching problems.

Having obtained precise image matching parameters, we usually need to convert those parameters into, for example, 3D shape information. In Chapter 5, we look into the problem of obtaining 3D structure information from a sequence of optical flows. The major difficulty of recovering dense structure information is the compu-

tation and memory required to maintain correlated uncertainty information. We will demonstrate an EKF-based system which incrementally builds a dense structure from an optical flow sequence. Finally, in Chapter 6, we summarize the major contributions of this thesis work and future directions.

Chapter 2

Gabor Filter Approach

2.1 Introduction

In order to extract spatially local information, many researchers have proposed to use Gabor filters due to their minimal uncertainty [28]. In this chapter, we address two problems in using Gabor filters to extract information from images. The first problem is the window size problem, in which we use a set of variable window Gabor filters to extract information at each frequency band. The benefit of this set of filters is that we no longer need to adjust the window size parameter for every experiment as we will show later. The other problem we will address is the stability problem caused by finite windows. We will extend Fleet's 1D phase stability analysis to the 2D phase and amplitude stability analysis, and develop a more general stability constraint. Finally, we will demonstrate the use of the tools in solving two important vision problems: focus and correspondence.

In traditional approaches, the window size for low level operators is tuned manually for specific experiments. Such a tuning process is usually undesirable for an algorithm to be flexible and stable. Intuitively, to extract information at low frequency, large windows are essential for the information to be stable, while for higher frequency components, smaller windows are preferred to preserve locality. This suggests a variable window scheme which decomposes a fixed window into a set of windows whose sizes are directly related to their frequencies. A set of self-similar, rotation and translation invariant 2D Gabor filters, which cover the whole spatial frequency plane up to the Nyquist frequency, has been used to extract information from images. By combining results from different frequency bands, a focus algorithm based on such decomposi-

tion demonstrates greater flexibility and adaptiveness than those based on the fixed window size scheme.

It is well known that windowing in the spatial domain is equivalent to convolving in the frequency domain, which limits overall frequency resolution. When amplitude and phase information is extracted from windowed filtering, it is possible that they are severely contaminated by the convolution. Therefore, if any algorithm makes use of the extracted information implicitly or explicitly without any further examination of the stability, it is subject to either large error or total failure. One goal of this chapter is to provide a set of constraints which are capable of identifying the contaminated information, and a generic framework for using the extracted phase or amplitude information.

Focus quality measurement is a typical problem using Fourier amplitude information. The problem can be formulated as measuring the change of amplitude between two images. Applying the set of 2D variable window Gabor filters and the technique of identifying unstable amplitude information, we can show that the performance of the algorithm can be improved in stability, adaptiveness and precision.

Optical flow matching is another typical problem which can be formulated using phase information. Because shifting in the spatial domain is equivalent to phase change in the frequency domain, we can measure phase changes to derive the spatial shift between two images. Just as we do in focus quality measurement, we will apply the set of filters and the technique of stability analysis to the problem, and thus improve the performance of the algorithm.

2.2 Related Research

As an alternative way of performing visual computing in spatial domain, the spatial frequency approach has been favored by many researchers for the applicability of various signal processing techniques [45] and biological evidence [19]. Previous research on visual computing in using Gabor filters has been concentrating on four areas: motion analysis, stereo matching, texture analysis, and focus measure.

Adelson and Bergen [2] and Heeger [34] modelled motions in 2D image space as orientations in 3D spatiotemporal space, and therefore introduced 3D oriented filters to measure image velocities. More recently, Fleet and Jepson [26] modelled the normal velocity as a function of local phase changes, and then used Gabor filters to measure changes of phase at every pixel location. For the stereo matching problem, Sanger

[76], Fleet *et al* [27], and Langley *et al* [48] proposed to use filters to extract phase information, and then compute disparities from them, while Jones and Malik [40] applied a set of linear spatial filters to images, and used responses from those filters as matching features. The spatial frequency approach also achieved great success in texture segmentation [14, 37, 47], and shape recovery from texture [46, 53].

One of the major disadvantages of the spatial frequency approach is the artifact introduced by windowing, which may cause substantial error if unnoticed. The usual way to overcome this problem has been to use large windows so that the artifact is negligible, but the price is severely reduced resolution. Fleet and Jepson [24] provided an excellent way to analyze the stability of phase information of Gabor filters. One of the major goals of this chapter is to generalize their work to a stability analysis method for both phase and amplitude information.

Measuring focus quality through spatial frequency analysis has been proposed in the literature [99, 12, 82, 69]. Few of the reported results have addressed the stability problem of amplitude information, which is the only information used in focus quality measurement. We will show that the stability analysis is capable of eliminating unstable amplitude information, therefore improving the performance of focus quality measurement.

2.3 Variable Window Gabor Filters

Limited by the uncertainty principle [28], any filter must compromise between spatial resolution Δx and spatial frequency resolution Δf [1]. The Gabor filter, which is a complex sinusoid modulated by a Gaussian function, is the one which can achieve minimal product of spatial uncertainty and frequency uncertainty:

$$G(x, f; \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} e^{-jfx}, \tag{2.1}$$

where the spatial extent σ determines the tradeoff between the spatial and the frequency resolutions.

If the spatial extent σ is constant across different frequency bands, i.e., we compute the spectrogram [99], the filters for low frequency bands will have much smaller numbers of waves than those for high frequency bands. Therefore, the spatial localization will be effectively reduced at high frequency, while in spectral domain, the

[1] In the thesis, f always refers to an angular frequency, i.e., the wavelength $\lambda = \frac{2\pi}{f}$

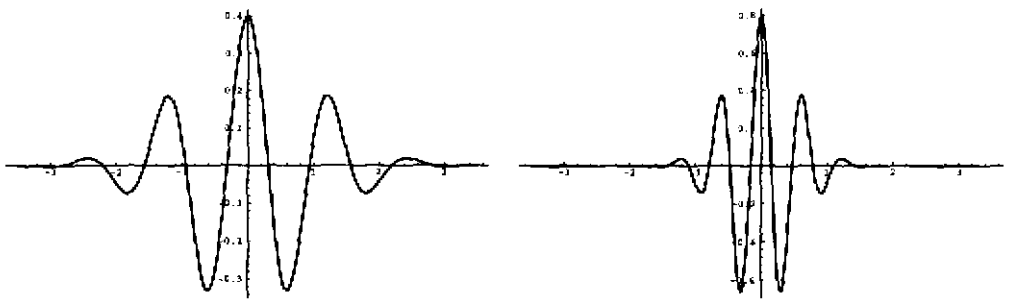


Figure 2.1: Variable Window Gabor Filters

spectral localization in terms of octaves (logarithmic frequency band) will be effectively reduced at low frequency. Suppose the spatial extent σ vary linearly with the wavelength of the tuned frequency, i.e.,

$$\sigma = \frac{k_1}{f}, \tag{2.2}$$

we then obtain a set of variable window Gabor filters:

$$G(x, f) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2 f^2}{2k_1^2}} e^{-jfx}. \tag{2.3}$$

Figure 2.1 shows two such filters.

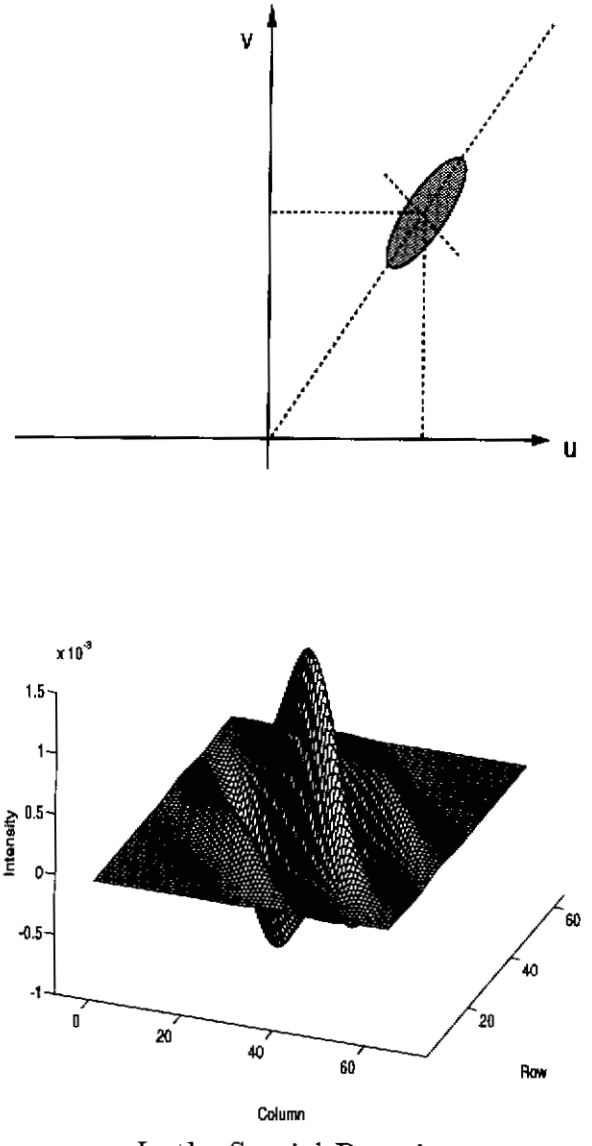
Extending the 1D variable window Gabor filters to 2D, we have the 2D Gabor filter:

$$G(x, y, f_x, f_y) = g(x, y)e^{j(f_x x + f_y y)}, \tag{2.4}$$

where $g(x, y)$ is an elliptical 2D Gaussian function in general, and (f_x, f_y) is the 2D peak frequency of the filter.

The radial frequency of this filter is $f = \sqrt{f_x^2 + f_y^2}$ and orientation $\theta = \tan(f_y/f_x)$. Also it is usually more convenient to have the modulating elliptical Gaussian with the same orientation θ as the filter. Figure 2.2 illustrates the peak frequency position, the spectral extent of one filter in the frequency domain, and the real part of the filter in the spatial domain. A remaining free parameter is the aspect ratio of the elliptical Gaussian. Inspired by some biological evidence reported in [19], we choose the aspect ratio as two-thirds.

As we did for the 1D Gabor filter, we constrain the spatial extent in the θ direction of the 2D filter to be proportional to the radial wavelength, we then obtain a set of 2D Gabor filters which are translated, rotated, and dilated or contracted versions of each other. In the experiments we show below, we used a set of 120 2D filters, which



In the Spatial Domain

Figure 2.2: Oriented 2D Gabor Filter

Aspect Ratio (k_2)	2/3
k_1	π
Angular Sampling	15°
Radial Sampling	$1/\sigma$
Minimal Frequency	$\pi/10$

Table 2.1: Sampling of the Frequency Domain by Gabor Filters

have 10 different radial frequencies and 12 orientations. The k_1 in Eq. 2.3 is set to π. The spectral extents of filters are described in Figure 2.3, assuming the extent of $g(x;\sigma)$ is from $-\sigma$ to σ. Table 2.1 shows "magic" numbers used in forming the set of Gabor filters.

As we will show in experiments, the benefit of using such a set of filters with different sizes of spatial support is that when results from different frequency bands are combined, if the image has strong high frequency components, the final result will be strongly influenced by results from high frequency bands; therefore, it is based more on very local information. On the contrary, if the image contains strong low frequency components, the final result will be based upon less local information. We no longer need to tune window size every time when processing new images.

2.4 Stability of Filtering

Though we obtain an amplitude value and a phase value at every pixel location by convolving the image with one filter, both values may be so severely contaminated by either windowing or noise that it is no longer valid to use them as approximations of the *real* amplitude and phase values. Therefore, any algorithm using them without discretion could potentially result in substantial errors. The goal of this section is to provide a way to quantify such a bias from real values and, subsequently, reduce errors caused by those contaminations.

2.4.1 Window Contamination

The origin of the window contamination is the convolution in the spatial frequency domain caused by window multiplication in the spatial domain. In other words, we are extracting amplitude and phase values not just at a single frequency, but at a weighted sum of a band of frequencies. The behavior of the sum may or may not be

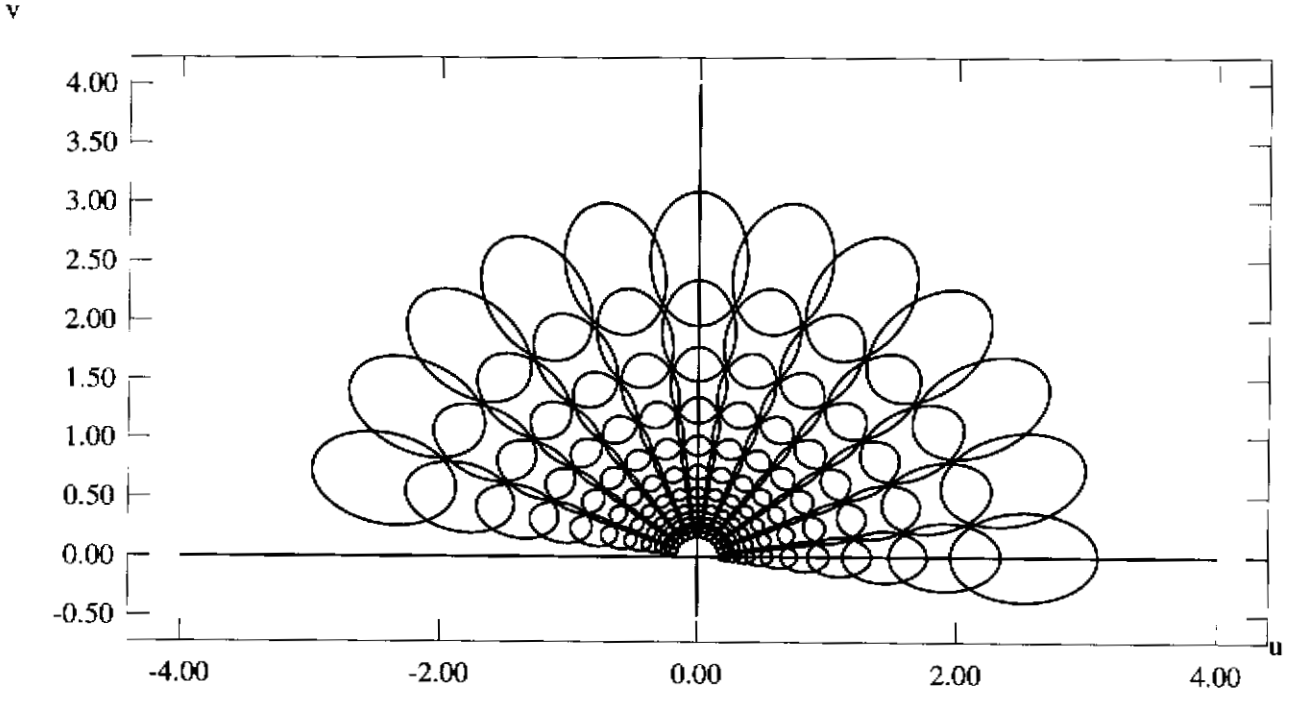


Figure 2.3: The Set of 2D Variable Window Gabor Filters

similar to that of a single sinusoid. The behavior of the sum is identical to that of a single sinusoid only when the underlying signal has zero amplitude for all frequencies in the passband except for the tuning frequency.

For simplicity, let us first consider the 1D case. For the sum of different frequency components to behave like a single frequency component, we ask the *local* behavior of the sum to satisfy following criteria:

1. The phase should change linearly with respect to the position, i.e., the derivative of the phase with respect to the pixel position should be the frequency:

$$\frac{\partial \phi}{\partial x} = f_0. \tag{2.5}$$

2. The phase should be stable with respect to the tuning frequency, i.e., the phase should be constant when the tuning frequency is shifted slightly. Since there is only one non-zero component in the passband, shifting the tuning frequency slightly should not change the phase value:

$$\frac{\partial \phi}{\partial (\log f)} = 0. \tag{2.6}$$

3. The amplitude should be stable with respect to the position, i.e., the derivative of amplitude with respect to the pixel position should be zero:

$$\frac{1}{a}\frac{\partial a}{\partial x} = 0. \tag{2.7}$$

4. The amplitude should be stable with respect to the tuning frequency, i.e., the amplitude should be constant when the tuning frequency is shifted slightly. Since only the tuning frequency component is non-zero in the passband, and the top of the Gaussian function has zero derivative, shifting the tuning frequency slightly should not change the amplitude value:

$$\frac{1}{a}\frac{\partial a}{\partial (\log f)} = 0. \tag{2.8}$$

Obviously, a complex sinusoid satisfies all the criteria perfectly[2]. Therefore, for the sum to approximate a single sinusoid locally, it has to approximately satisfy those

[2]For criterion 4, the amplitude is constant only when the tuning frequency is approximately equal to the frequency of the sinusoid.

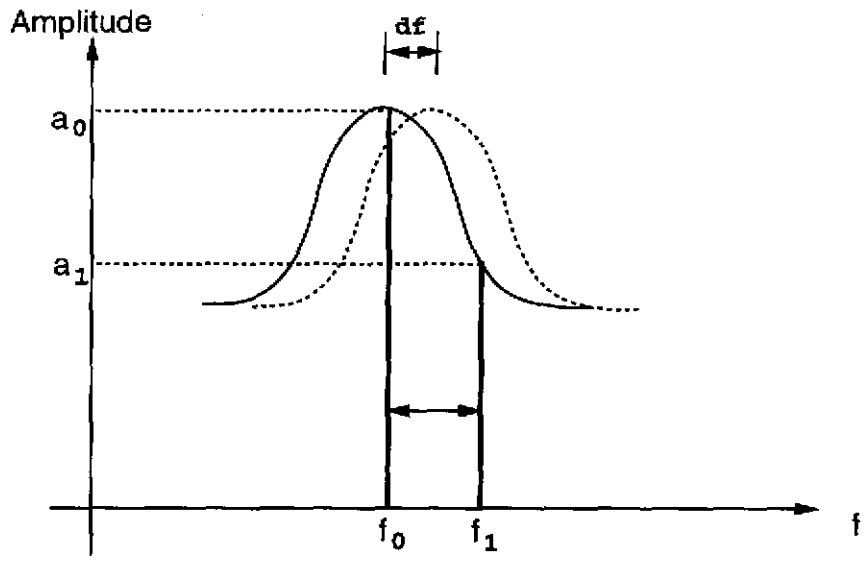


Figure 2.4: Stability Analysis

criteria locally. On the other hand, if the sum does satisfy all the constraints, then in the spatial domain, the band-passed signal can be well approximated by a sinusoid locally. Consequently, we regard locally unstable signals resulting from bandpass filtering as those which can not be well approximated by a single sinusoid.

Then the remaining task is to find one or more constraints to indicate to what extent those criteria are satisfied. Let us further simplify the problem by assuming that the signal is a sum of two independent sinusoids of frequency f_0 and f_1, and weighted amplitude a_0 and a_1. Then the sum is:

$$ae^{j\phi(x)} = a_0 e^{j\phi_0(x)} + a_1 e^{j\phi_1(x)}, \tag{2.9}$$

where $\phi(x), \phi_0(x)$ and $\phi_1(x)$ are phases, and $\frac{\partial\phi_0(x)}{\partial x} = f_0$, $\frac{\partial\phi_1(x)}{\partial x} = f_1$.

Let $\frac{a_1}{a_0} = \lambda$, $\Delta\phi = \phi_1 - \phi_0$ and $\Delta f = f_1 - f_0$, after some manipulations, the criteria 1 and 3 can be expressed as:

$$T_1 = \left|\frac{\partial\phi}{\partial x} - f_0\right| = \left|\frac{\lambda\Delta f(\lambda + \cos\Delta\phi)}{\lambda^2 + 2\lambda\cos\Delta\phi + 1}\right| \tag{2.10}$$

$$T_3 = \left|\frac{1}{a}\frac{\partial a}{\partial x}\right| = \left|\frac{\lambda\Delta f\sin\Delta\phi}{\lambda^2 + 2\lambda\cos\Delta\phi + 1}\right|. \tag{2.11}$$

When the tuning frequency is shifted as in Figure 2.4, it will affect the amplitude

ratio of two components

$$\frac{d\lambda}{df} = \frac{\frac{a_1 e^{-(\Delta f - df)^2\sigma^2/2}}{e^{-(\Delta f)^2\sigma^2/2}} - \frac{a_1}{a_0}}{df} = \frac{\lambda(e^{(\Delta f)(df)\sigma^2} - 1)}{df} = \lambda \Delta f \sigma^2, \tag{2.12}$$

where σ is the spatial extent of the filter.

Using Equation 2.12 and $d(\log f) = \frac{df}{f}$, we can express the criteria 2 and 4 as:

$$\begin{aligned} T_2 = \left|\frac{\partial \phi}{\partial f} f\right| &= \left| f \frac{\partial \phi}{\partial \lambda} \frac{d\lambda}{df} \right| \\ &= \left| \frac{\sigma^2 f \Delta f \lambda \sin \Delta\phi}{\lambda^2 + 2\lambda \cos \Delta\phi + 1} \right|, \end{aligned} \tag{2.13}$$

$$\begin{aligned} T_4 = \left|\frac{f}{a}\frac{\partial a}{\partial f}\right| &= \left| \frac{f}{a} \frac{\partial a}{\partial \lambda} \frac{d\lambda}{df} \right| \\ &= \left| \frac{\sigma^2 f \Delta f \lambda(\lambda + cos\Delta\phi)}{\lambda^2 + 2\lambda \cos \Delta\phi + 1} \right|. \end{aligned} \tag{2.14}$$

Summing those criteria together, we obtained an overall criterion for the stability of the signal. The criterion is composed of two independent sinusoids:

$$T^2 = T_1^2 + T_2^2 + T_3^2 + T_4^2 = (1 + \sigma^2 f)^2 (T_1^2 + T_3^2). \tag{2.15}$$

If $T^2 \approx 0$, the four criteria listed previously are approximately satisfied. In other words, the local behavior of the sum is similar to that of a single sinusoid when T^2 is small.

The rightmost expression in Eq. 2.15 provides another benefit for the computation of T, i.e., we do not actually need to compute T_2 or T_4, which usually require very dense sampling of the frequency domain, because they are derivatives with respect to the frequency. In practice, the computations of T_1 and T_3 are straightforward. Note the difference between the criterion proposed here and those in [24]. The criterion proposed here is more general in that only when Eq. 2.3 is satisfied, the two criteria are equivalent and the criterion in Eq. 2.15 is for both amplitude and phase information.

Figure 2.5 illustrates the stability criterion when applied to the sum of a fixed frequency sinusoid and a chirp signal, i.e., the frequency increases linearly with respect to pixel locations with the same magnitude ($\lambda = 1$). The upper two graphs show Δf and $\Delta\phi$ in Eq. 2.15; the lower left graph is the signal itself in spatial domain; and the lower right graph shows the computed stability criterion T. Obviously, the spikes in the stability criterion are caused by asynchronous sinusoids, i.e., those sinusoids are

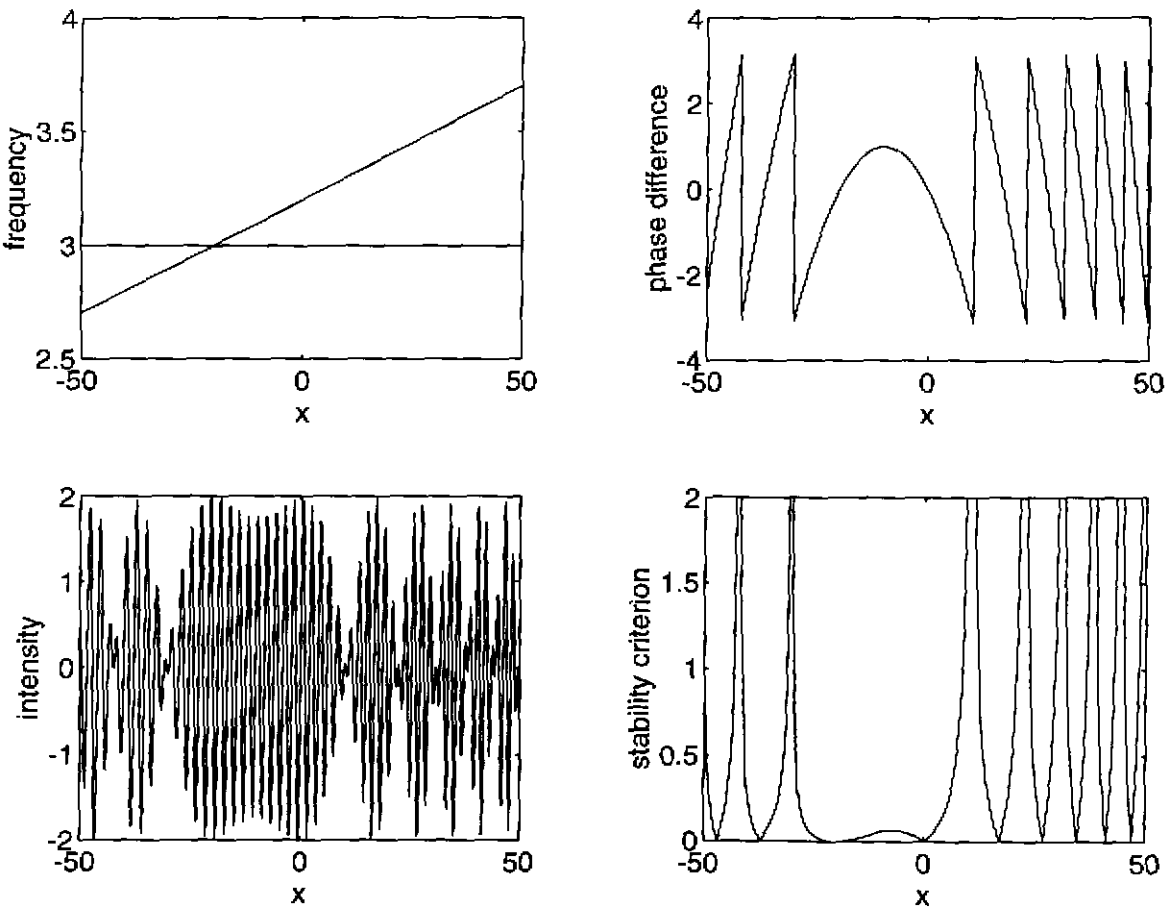


Figure 2.5: Stability Criterion

cancelling each other because the phase difference is approaching $\pm\pi$. If we threshold the stability criterion, we can detect the parts of the signal where the two sinusoids cancelled each other.

As proven in Appendix A, Eq. 2.15 is also valid for an arbitrary signal. Generalizing Eq. 2.15 to any signal, and taking the spatial extent into consideration, we obtain the criterion function T' as in Eq. 2.16:

$$T' = \frac{T}{k_1} \approx \sqrt{\left(\frac{\partial \phi}{\partial x} - f_0\right)^2 + \left(\frac{1}{a}\frac{\partial a}{\partial x}\right)^2}\,\sigma, \tag{2.16}$$

where k_1 is the constant in Eq. 2.3, and f_0 is the tuning frequency.

Elimination of unstable information can be done by simple thresholding, i.e., if at any location and any frequency band, T' exceeds a certain threshold, then the amplitude and phase information in that frequency band is regarded as unstable.

Notice that the threshold should be a constant with respect to different frequency bands.

In case of 2D Gabor filters proposed previously, let us assume the X axis is coincident with the radial orientation $\mathbf{f}$ of a filter, the four criteria can be expressed as:

$$\mathbf{T_1} = \nabla_r \phi - \mathbf{f} = \frac{\lambda(\lambda + \cos\Delta\phi)}{\lambda^2 + 2\lambda\cos\Delta\phi + 1} \begin{pmatrix} \Delta f_x \\ \Delta f_y \end{pmatrix}, \tag{2.17}$$

$$\mathbf{T_2} = f\nabla_f \phi = \frac{\lambda f\sigma^2 \sin\Delta\phi}{\lambda^2 + 2\lambda\cos\Delta\phi + 1} \begin{pmatrix} \Delta f_x \\ \frac{\Delta f_y}{k_2^2} \end{pmatrix}, \tag{2.18}$$

$$\mathbf{T_3} = \frac{1}{a}\nabla_r a = \frac{\lambda\sin\Delta\phi}{\lambda^2 + 2\lambda\cos\Delta\phi + 1} \begin{pmatrix} \Delta f_x \\ \Delta f_y \end{pmatrix}, \tag{2.19}$$

$$\mathbf{T_4} = \frac{f}{a}\nabla_f a = \frac{\lambda f\sigma^2(\lambda + \cos\Delta\phi)}{\lambda^2 + 2\lambda\cos\Delta\phi + 1} \begin{pmatrix} \Delta f_x \\ \frac{\Delta f_y}{k_2^2} \end{pmatrix}, \tag{2.20}$$

where ∇_r is the gradient operator in spatial domain, ∇_f is the gradient operator in spatial frequency domain, $\mathbf{f}$ is the peak tuning frequency, Δf_x and Δf_y are the frequency difference in X and Y directions, and k_2 is the aspect ratio of the filter. Similarly the overall stability criterion can be expressed as:

$$\|T'\|^2 = \sigma^2(\|\mathbf{ST_1}\|^2 + \|\mathbf{ST_3}\|^2), \tag{2.21}$$

Where the $\mathbf{S}$ is the aspect scaling matrix

$$\mathbf{S} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{k_2^2} \end{pmatrix}.$$

2.4.2 Noise Contamination

Due to different window sizes of filters, the effects of noise on the amplitude and phase values are also different. Assuming additive white noise, the noise component included in filter output can be expressed as:

$$N(f_0) = \int_{-\infty}^{+\infty} e^{-(f-f_0)^2\sigma^2/2} n e^{j\phi_n} df = \frac{\sqrt{2\pi}n}{\sigma} \int_{-\infty}^{+\infty} g(f - f_0; \frac{1}{\sigma}) e^{j\phi_n} df, \tag{2.22}$$

where $g(f - f_0; \frac{1}{\sigma})$ is a Gaussian function with center at f_0 and extent $\frac{1}{\sigma}$, n is the magnitude of noise at every frequency, ϕ_n is the random phase of noise.

Because we assumed white noise, the expected value of the rightmost integral term in Eq. 2.22 is assumed to be a complex number with constant magnitude and random phase. Therefore, using Eq. 2.2, we obtain the relative noise level in every frequency band:

$$\|N(f_0)\| = k_3 f_0, \tag{2.23}$$

where k_3 is a constant as

$$k_3 = \frac{\sqrt{2\pi}n}{k_1}. \tag{2.24}$$

The estimation of noise level in filter outputs will be used to merge results from different filters.

2.5 Applications

The way of using the Gabor filter to compute matching parameters from two images is illustrated in Figure 2.6. The filter outputs from two images are compared to compute the matching parameters when they are not severely contaminated by window effects or noise. The general framework developed in this chapter is applicable to any vision task which makes use of either amplitude or phase information. We will explain two specific applications, focus quality measurement and 2D correspondence. The reason we choose these two applications is, as we will see, focus quality measurement makes use of amplitude information only, and 2D correspondence makes use of phase information only.

2.5.1 Focus Quality Measurement

As explained in [99, 69], the key problem in focus quality measurement can be stated as, given two images which are blurred to different extents, how to measure *locally* the difference of blurring at each pixel location. Modelling the blurring as a convolution with a Gaussian, we have [69]:

$$\sigma_0 = \frac{1}{f}\sqrt{|\ln \|I_0(f)\|^2 - \ln \|I_1(f)\|^2|}, \tag{2.25}$$

where σ_0 represents the blurring difference, and $\|I_0(f)\|$ and $\|I_1(f)\|$ are amplitude values of two images at frequency f.

We can apply the set of 2D variable window Gabor filters to extract amplitude information at each frequency band, eliminating unstable amplitude information by

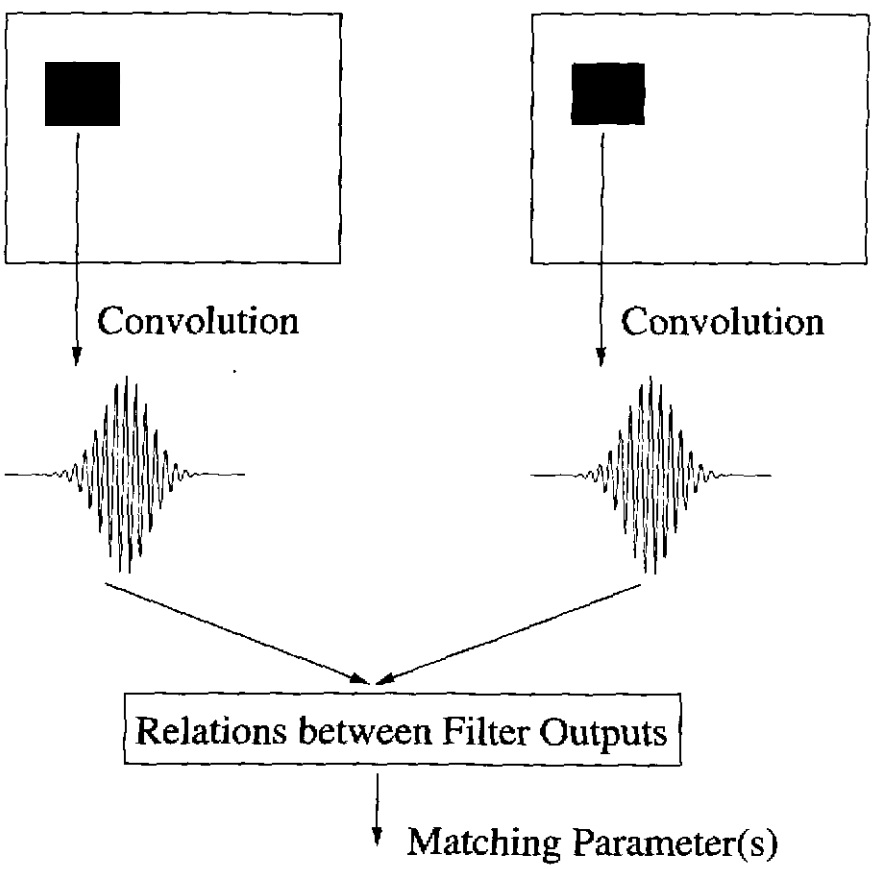


Figure 2.6: Matching by Gabor Filters

thresholding in Eq. 2.21, then fit all other stable amplitude values and their included noise level (Eq. 2.23) into the following linear (with respect to σ_0^2) equation[98].

$$\sigma_0^2\|\mathbf{f}\|^2 = \left(\ln\|I_0(\mathbf{f})\|^2 - \ln\|I_1(\mathbf{f})\|^2\right) + c, \tag{2.26}$$

where the constant c is used to compensate for illumination difference between two images and, instead of the peak tuning frequency, $\|\mathbf{f}\|^2$ can better approximate the average of two instantaneous frequencies $\|\frac{1}{2}(\nabla_r\phi_0 + \nabla_r\phi_1)\|^2$. Ordinary χ^2 estimation can be applied to the fitting with the uncertainty of the right side expressed as [98]:

$$\frac{\|N(\mathbf{f})\|}{\|I_0(\mathbf{f})\|} + \frac{\|N(\mathbf{f})\|}{\|I_1(\mathbf{f})\|}. \tag{2.27}$$

The iterative estimation in [99] can be also efficiently implemented as convolutions in the spatial frequency domain:

$$I(x) \otimes g(x;\sigma_0) \otimes f(x) = (I(x) \otimes f(x)) \otimes g(x;\sigma_0), \tag{2.28}$$

where $f(x)$ is a bandpass filter.

2.5.2 Optical Flow

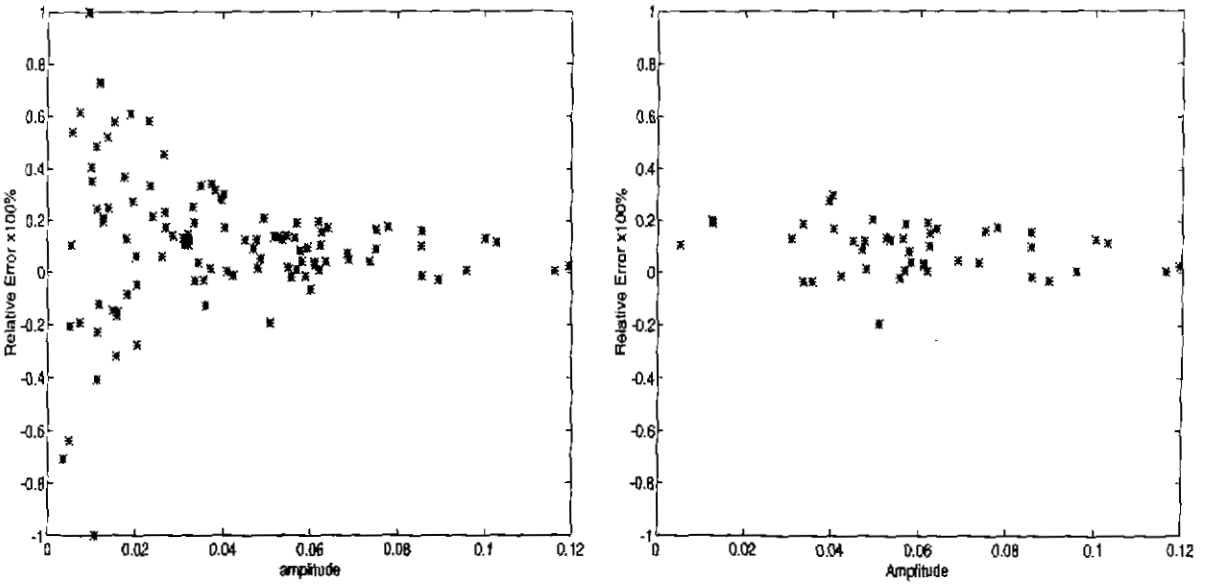
The key problem in the optical flow computation is to find the spatial translation between two images at each pixel location. Because a spatial translation is equivalent to a phase shift in frequency domain, we can derive the spatial translation from phase difference [76, 27, 48].

An obvious approach similar to that of focus quality measurement is to apply the set of 2D variable window Gabor filters to extract phase information at each frequency band, eliminate unstable phase information by thresholding in Eq. 2.21, then minimize the following to find disparity $\mathbf{r}$:

$$\sum_{\phi_0\phi_1 \text{ stable}} \frac{((\phi_1 - \phi_0 - \mathbf{f}\cdot\mathbf{r}) \bmod 2\pi)^2}{\left(\frac{\|N(\mathbf{f})\|}{\|I_0(\mathbf{f})\|} + \frac{\|N(\mathbf{f})\|}{\|I_1(\mathbf{f})\|}\right)^2}, \tag{2.29}$$

where $N(\mathbf{f})$ is the noise component in Eq. 2.23.

The iterative estimation in [27] can also be done by simply translating one image locally according to previously estimated disparities.



Before Elimination After Elimination

Figure 2.7: Elimination of Unstable Amplitude Information

2.6 Experimental Results

2.6.1 Elimination of Unstable Information

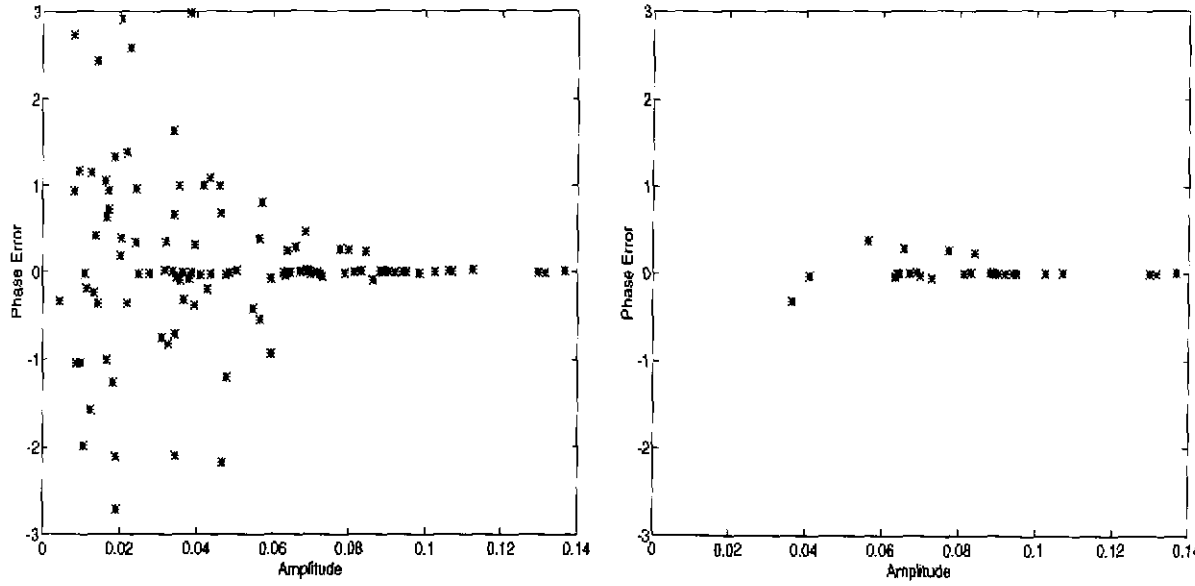
First we artificially convolved an image with a Gaussian function $g(x; \sigma_0 = 1.0)$, then convolved the set of filters with original and blurred images, and analyzed the relations between amplitudes of various frequencies at an arbitrarily chosen pixel location. The left half of Figure 2.7 shows the relative error of estimating σ (Eq. 2.26) from a single frequency band without eliminating unstable amplitude information[3], and the right half shows the error after thresholding by $T' = 1.25$ in Eq. 2.21.

To illustrate the effectiveness of eliminating unstable phase information, we spatially shifted an image by $\mathbf{r}$, and then the phase difference φ represents the error of phase with respect to ideal sinusoids:

$$\varphi = \left(\mathbf{r} \cdot \frac{\nabla\phi_1 + \nabla\phi_2}{2} - (\phi_1 - \phi_2) \right) \bmod 2\pi, \tag{2.30}$$

where ϕ_1 and ϕ_2 are the two phase values resulting from convolving a filter with the original and shifted images. Ideally, φ should be zero at any frequency band. At an

[3]In this case c in Eq. 2.26 is known to be zero.



Before Elimination After Elimination

Figure 2.8: Elimination of Unstable Phase Information

arbitrarily chosen pixel location, the left half of Figure 2.8 shows φ of every frequency band without eliminating unstable phase information, and the right half shows φ after thresholding by $T' = 1.25$ in Eq. 2.21.

As illustrated in Figure 2.7 and Figure 2.8, the identification of unstable information is indeed very accurate. Depending on a specific application, the threshold can be changed to satisfy either a looser or tighter requirement for the output of a bandpass filter to resemble a sinusoid.

As illustrated in Figure 2.7 and Figure 2.8, there also exists a strong correlation between instability and amplitude values. This suggests an amplitude thresholding scheme ([98, 76]), which assumes unstable amplitude and phase information is caused by low amplitude values. Even though this amplitude thresholding scheme could indeed work well in some cases, when we compared its results with those from the stability thresholding, we found that it usually required that images had strong high frequency component because otherwise weak high frequency components were always below the threshold even though they *were* stable with respect to the stability criterion. On the other hand, if we lower the amplitude threshold, a large amount of unstable information will get through. In the experiments described below, the amplitude threshold was set to 0.06 (refer to the horizontal axes in Figure 2.7 and Figure 2.8), assuming images were normalized so that their DC components had

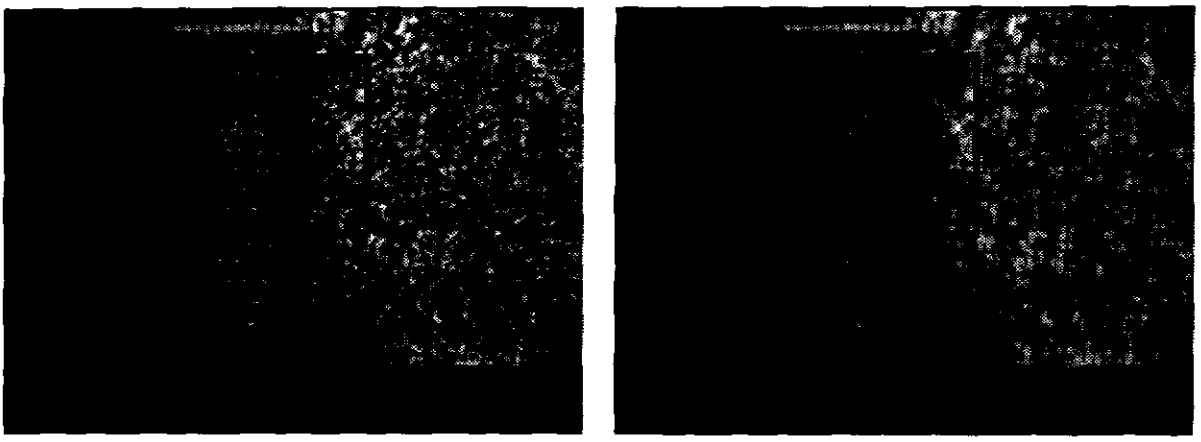


Figure 2.9: Two Images Taken under Different Lens Apertures

amplitude value of 1.0.

It should be noted that in the above synthetic experiments, there was *no* noise and *no* shift variance effects in the images. All errors shown in Figure 2.7 and Figure 2.8 are caused solely by window effects. The stability criterion used in this chapter could identify those cases where window effects are overwhelming. But it cannot *correct* the error caused by window effects because even the stable information still has a certain amount of error. Still, the overall performance of vision algorithms such as focus and stereo is improved. Because we measure many different filters at each pixel location, eliminating the unstable ones allows us to attend to the remaining stable filter outputs.

Figure 2.9 shows two images of two textured surfaces at different depths. The images were taken under different lens apertures, which made the difference of focus quality a function of depth [98]. We applied the set of filters to the two images, then eliminated unstable amplitude information by thresholding the stability criterion, and fitted a line against Eq. 2.26 to obtain σ_0 at each pixel location. Figure 2.10 shows the difference of focus quality in the rectangular region in Figure 2.9. As we can see, the two surfaces were obviously separated, and the depth discontinuity was well located. Figure 2.11 shows the computed focus quality difference by amplitude thresholding. The areas with strong high frequency components were recovered correctly, while in other areas, the results showed instability.

Figure 2.12 shows two images of a rotating ball. We applied the techniques of filtering and stability analysis to the two images, and used Eq. 2.29 to find 2D image motion. Figure 2.13 shows the optical flow between the two images. Intrinsic to the

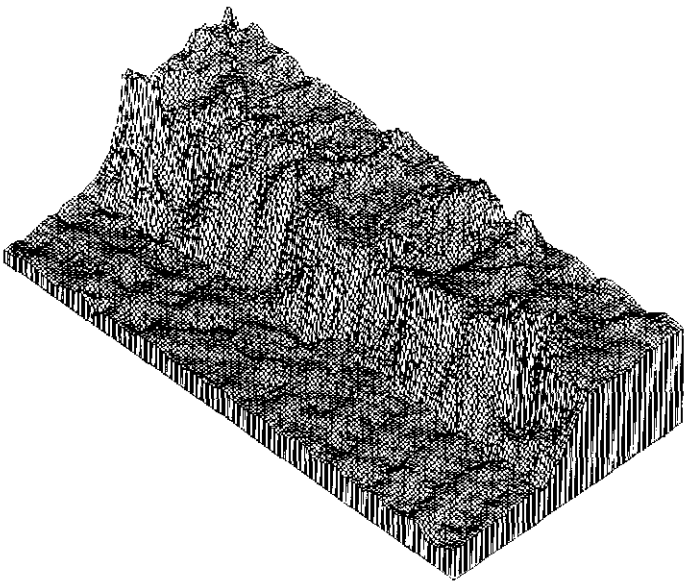


Figure 2.10: Difference of Focus Quality Using Stability Threshold

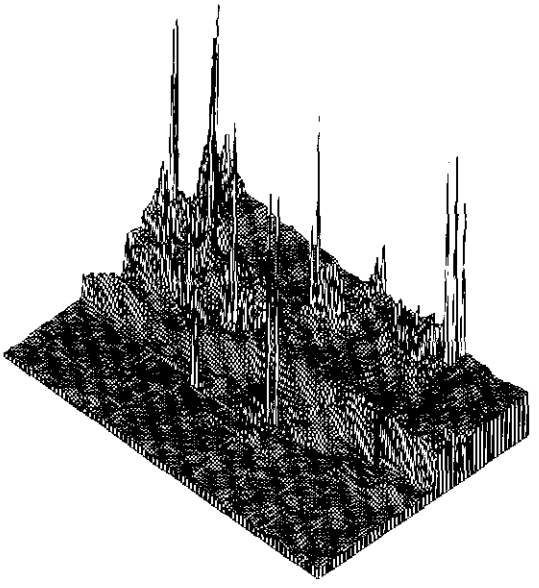


Figure 2.11: Difference of Focus Quality Using Amplitude Threshold

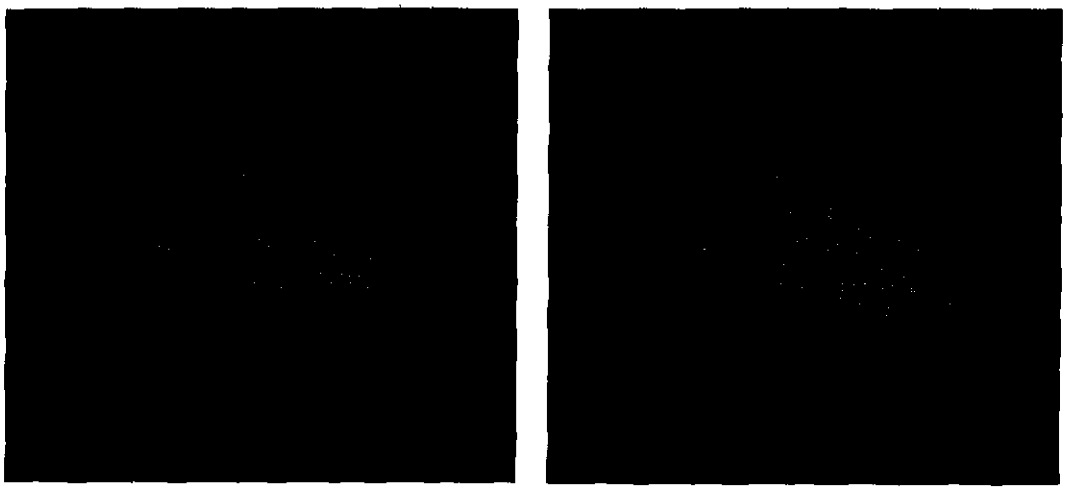


Figure 2.12: Two Images of a Ball

algorithm itself, sub-pixel accuracy was achieved without any interpolation. Given sparse features such as those on the ball, the algorithm could automatically avoid unstable information obtained in featureless areas. Figure 2.14 shows the optical flow using the amplitude threshold. Around the boundary of the ball, where the contrast was lower, the algorithm generated unstable results.

2.6.2 Fixed Window versus Variable Window

The selection of window size for any window operation is a compromise between resolution and stability. While large windows reduce effects of noise, and make results more stable, the resolution is reduced at the same time. On the other hand, small windows have the advantage of preserving locality, i.e., high resolution; they are potentially unstable. Therefore a proper window size has to be *ad hoc* to a specific problem. An implicit rule of choosing a window size is to select a window which is as small as possible while the error caused by noise and windowing is still within the acceptable range.

An alternative other than selecting a window size every time when processing new images, is to adopt the variable window scheme proposed in this chapter. Because window sizes are proportional to wavelengths as in Table 2.1, this variable window scheme uses large windows when frequencies are low, small windows when frequencies are high. An intuitive explanation of using variable windows is that all the filters have the same number of wavelets.

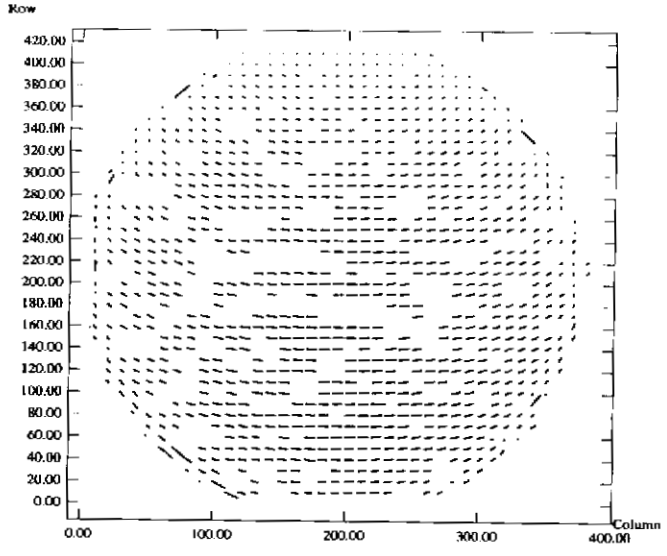


Figure 2.13: Optical Flow Using Stability Threshold

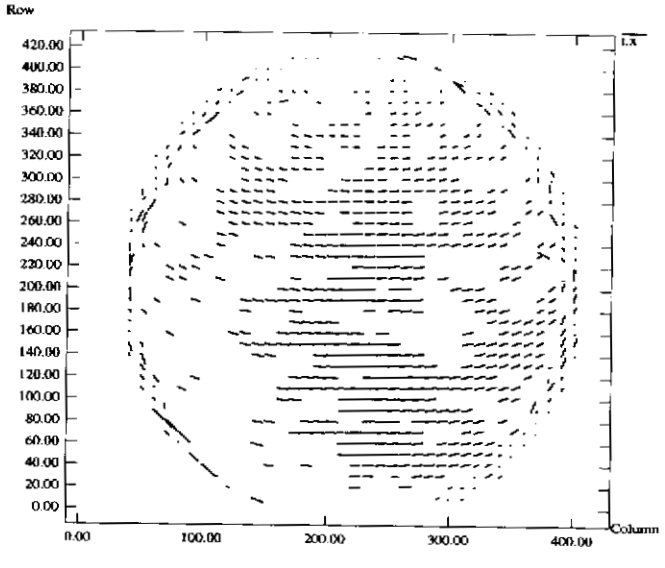


Figure 2.14: Optical Flow Using Amplitude Threshold

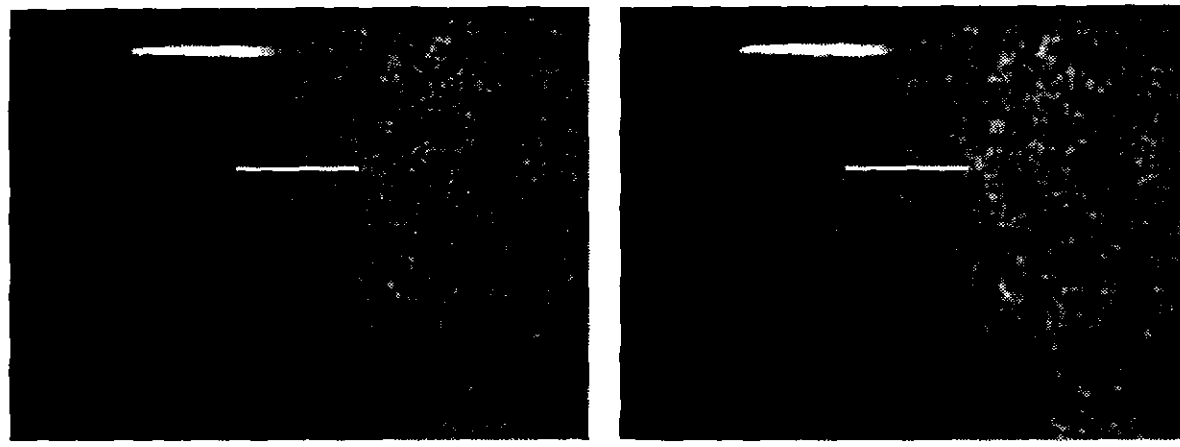


Figure 2.15: Nearly Focused Image Pair

As we will see in the next chapter, the variable window scheme ensures that the estimation errors caused by window effects are statistically of the same magnitude to all frequency bands, while for the fixed window scheme, the errors caused by window effects are larger for low frequency bands than for high frequency bands.

Figure 2.15 and Figure 2.16 show examples of input images with different contents. The images in Figure 2.15 contain a considerable amount of high frequency information, while the images in Figure 2.16 contain only low frequency information. The black lines in the images are the locations where focus quality differences are shown. Focus quality measurements are done using the fixed window scheme [99] with a large window ($\sigma = 20.0$) and a small window ($\sigma = 10.0$), and the variable window scheme described in previous sections.

Figure 2.17 shows focus quality difference of a single scan line between images in Figure 2.15. The large window scheme resulted in a large slope around the depth discontinuity. The small window scheme displayed more noise in results before and after the discontinuity even though it located the discontinuity more precisely. Figure 2.18 showed the results of focus quality difference between images in Figure 2.16. This time, the small window scheme generated unstable results, and the large window scheme produced blurred, but stable results. In both Figure 2.17 and Figure 2.18, the variable scheme resulted in a meaningful compromise between resolution and precision. Table 2.2 shows root mean square errors of all cases. The variable window scheme performed significantly better than the large window scheme in the nearly focused image pair, and the small window scheme in the defocused image pair.

Figure 2.16: Defocused Image Pair

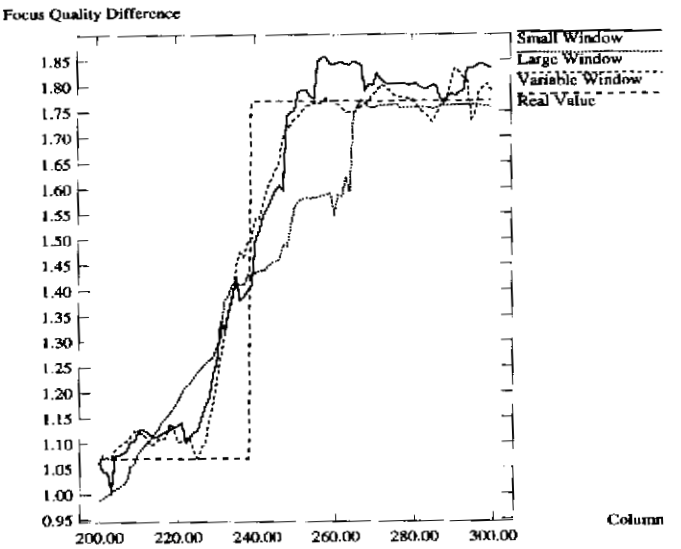


Figure 2.17: Focus Quality Difference of the Nearly Focused Images

	Nearly Focused Image Pair	Defocused Image Pair
Small Fixed Window	0.1218	0.2708
Large Fixed Window	0.1677	0.1947
Variable Window	0.1150	0.1664

Table 2.2: RMS Errors of Different Window Schemes

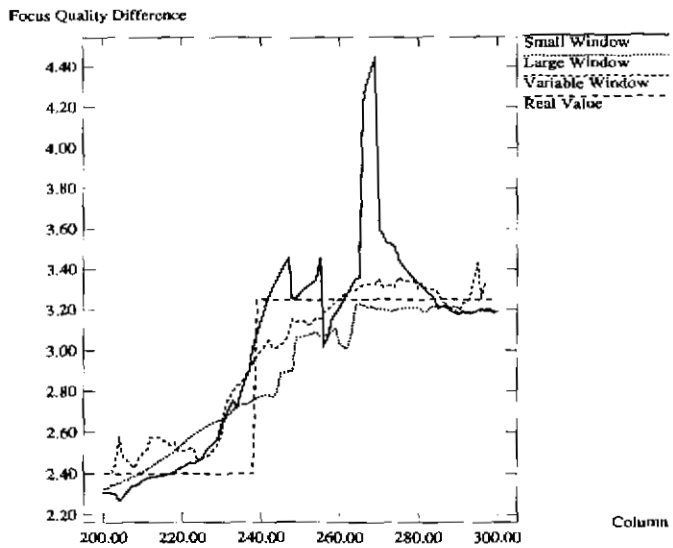


Figure 2.18: Focus Quality Difference of the Defocused Images

2.7 Summary

In this chapter, we examined the errors caused by window effects in the traditional Gabor filter approach. Since the fundamental assumption of using Gabor filters to compute matching parameters is that the behaviors of the filter outputs are similar to those of infinitely long sinusoids, we formulated the stability criterion as a measure of such similarity. We showed that such a criterion was indeed very effective in finding information which was overwhelmingly contaminated by window effects. We also showed that the variable window scheme could improve the adaptiveness of the matching algorithm.

The information which satisfies the stability criterion is still contaminated by window effects even though it is not as serious as the unstable information. In the next chapter, we will look into the origination of the window effects, and try to eliminate them completely.

Chapter 3

Moment Filter Approach

3.1 Introduction

Finite-width windows are widely used in various vision algorithms to extract spatially local information from images. While the choice of window width has a profound impact on the quality of results, no quantitative way to measure and eliminate the negative effects of finite-width windows has been proposed in computer vision literature. We propose a new set of filters: "moment" filters. Due to their recursiveness, the effects of finite-width windows can be explicitly analyzed and eliminated.

Many low level visual computing problems such as focus, stereo, etc. can be formulated as problems of extracting one parameter of a "non-stationary" transformation between two images. By non-stationary we mean that the parameter is position dependent in the image. For example, we can formulate a general correspondence problem as one in which we extract disparity as one parameter of a position-dependent phase-shifting transformation. The non-stationary nature of the parameter forces us to use finite-width filters to extract local information from images. Despite the fact that most algorithms compute the parameters by either explicitly convolving images with the finite-width filters (e.g., phase-based stereo) or by implicitly combining the outputs of finite-width filters in the spatial domain (e.g., Sum of Squared Difference in stereo), the effects of finite window width are seldom analyzed. Often the information extracted from finite-width filters is regarded as an approximation of that from infinite-width filters as shown in the previous chapter. But such an approximation makes the following computations much less accurate or even totally impossible.

The most fundamental difference between finite- and infinite-width filters in the

Fourier domain is that the bandwidth of finite-width filters must be non-zero, while that of infinite-width filters is zero. Since a non-stationary transform in the Fourier domain can be an arbitrary curve within the passband of a finite-width filter, the approximation in ignoring the effects of finite width is equivalent to using a constant to approximate the arbitrary curve in the passband as in Figure 1.2. The moment filters approximate the curve by a polynomial, whose order can be arbitrarily high so that the approximation can be as accurate as a specific task requires. Therefore, the extraction of the parameter based upon the polynomial approximation is much more accurate.

We apply the technique of moment filters to the problems of focus and stereo. Traditionally, people estimate the parameter by comparing the outputs of convolutions of the two images with the same filter – usually a Gabor filter – as we did in the previous chapter. Our moment filter approach provides a new insight into this traditional approach by showing that the convolution of one image with the Gabor filter can be represented as an infinite sum of convolutions of the other image with moment filters. From this point of view, the traditional Gabor filter approach is simply the zeroth order truncation of the infinite sum, which, in many cases, is the major source of inaccuracy. By using higher-order moment filters, we are able to reduce the inaccuracy.

When extracting a parameter using finite-width filters, people implicitly assume that the parameter is constant within the width of filters. But another aspect of the non-stationarity is that the parameter actually changes *within* the effective width of the filters. We call this the shift variance problem, but the same effect is also referred to as foreshortening in stereo. Unlike shift invariant transformations, which are well defined in the Fourier domain, this shift variant transformation usually cannot be represented in the Fourier domain as we will show later. The recursiveness of the moment filters in the spatial domain allow modelling this effect as a modification of the Gabor filter convolution by convolutions of the higher order moment filters. Applying this technique to the focus and stereo problems, we demonstrate that we can not only eliminate the negative effect of foreshortening, but also estimate the degree of foreshortening quantitatively.

In the literature, to deal with the errors and instability caused by finite width filters, two techniques were used to deal with window effects: stability criterion and instantaneous frequency. The stability criterion, as explained in the previous chapter, detects the situation where the filter output is overwhelmingly dominated by window

contamination. The instantaneous frequency [27] as the spatial derivative of phase can further improve accuracy of the results, though its actual meaning is not clear. As we will show later, both techniques are actually special cases of the moment filter technique. Both the stability criterion and instantaneous frequency can be directly computed from the first moment filter. From this point of view, the moment filter approach generalizes these two techniques, and provides a clear interpretation of the concept of the instantaneous frequency and its relationship to stability.

3.2 Theory of Moment Filters

3.2.1 Definitions

We define the ith order moment filter as: $(i = 0, 1, 2, ...)$

$$m_i(x) = p_i(x)\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}}e^{-jf_0x}, \tag{3.1}$$

in which

$$p_i(x) = \begin{cases} \frac{2^n n!}{\sigma^{2n}}L(n, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) & i = 2n \\ j\frac{2^n n!}{\sigma^{2n+2}}xL(n, \frac{1}{2}, \frac{x^2}{2\sigma^2}) & i = 2n+1, \end{cases} \tag{3.2}$$

where n is an integer, j is $\sqrt{-1}$, and $L(n, \alpha, x)$ is the generalized Laguerre polynomial [67], with $L(0, c, x) = 1$.

By the above definition, the zeroth order moment filter is the Gabor filter $G(x; f_0)$ with the peak frequency f_0:

$$m_0(x) = G(x; f_0) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}}e^{-jf_0x}. \tag{3.3}$$

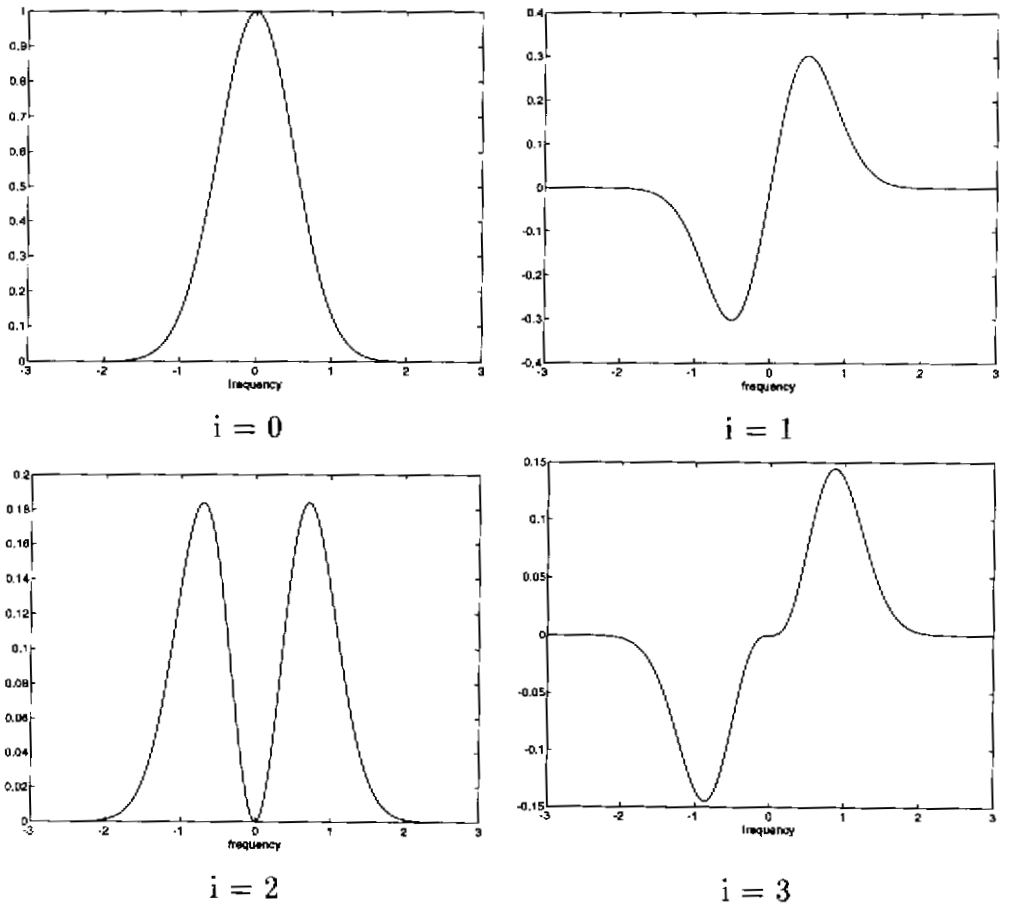
In the Fourier domain, the moment filter is:

$$M_i(f) = \mathcal{F}[m_i(x)] = (f - f_0)^i e^{-(f-f_0)^2\sigma^2/2}. \tag{3.4}$$

Therefore, the peak frequency of the ith moment filter is:

$$\text{Peak}[M_i(f)] = f_0 \pm \frac{\sqrt{i}}{\sigma}, \tag{3.5}$$

which moves away from the original peak frequency as i increases. Figure 3.1 illustrates the pass bands of moment filters as i increases, assuming $f_0 = 0$.



i = 0 i = 1

i = 2 i = 3

Figure 3.1: The Pass Band Moves as i Increases

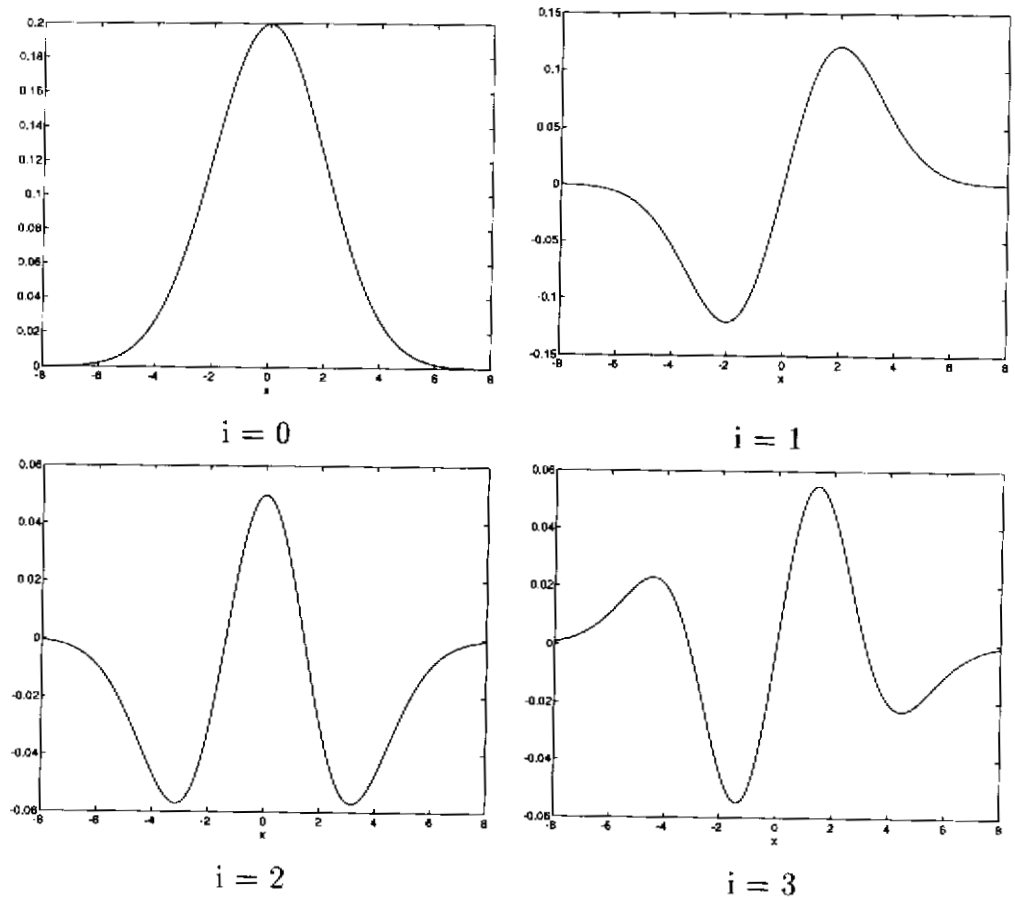


Figure 3.2: The Moment Filters in Spatial Domain when $f_0 = 0$

Another way to understand the moment filters is that the profiles of the filters, i.e., $m_i(x)$ when $f_0 = 0$, can be generated by differentiating the Gaussian function $g(x)$.

$$\mathcal{F}\left[\frac{d^n}{dx^n}g(x)\right] = (jf)^n\mathcal{F}[g(x)] = j^n f^n e^{-f^2\sigma^2/2} \tag{3.6}$$

From Eq. 3.4 and Eq. 3.1, we then have

$$m_i(x) = j^{-i}e^{-jf_0x}\frac{d^i}{dx^i}g(x) \tag{3.7}$$

Figure 3.2 illustrates some moment filters in the spatial domain when $f_0 = 0$, and Figure 3.3 illustrated some moment filters in the spatial domain when $f_0 \neq 0$.

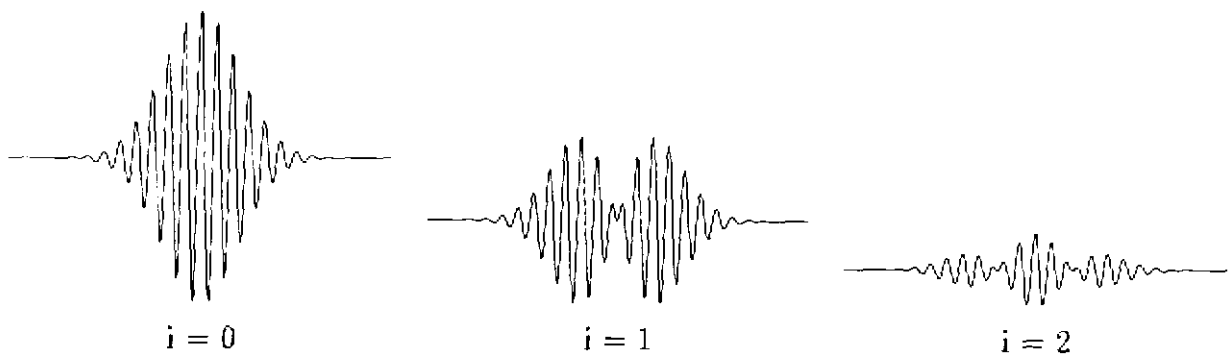


Figure 3.3: The Moment Filters in Spatial Domain when $f_0 \neq 0$

3.2.2 Decomposition by Moment Filters

In this section, we consider moment filters based upon *one* Gabor filter, which has the peak frequency f_0. We also limit our discussion to one specific spatial location $x_0 = 0$ unless we advise otherwise.

Let

$$k_i(x) = p_i^*(x)e^{jf_0x}, \tag{3.8}$$

and define the inner product of two arbitrary functions $g(x)$ and $h(x)$ over the functional space as

$$< g(x), h(x) >= \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{+\infty} g(x)h^*(x)e^{-\frac{x^2}{2\sigma^2}}dx, \tag{3.9}$$

where $*$ denotes complex conjugate, we then have

1. The functions k_i in Eq. 3.8 are orthogonal with each other under the definition of the inner product, i.e.,

$$< k_i(x), k_j(x) >= 0, \text{when} i \neq j. \tag{3.10}$$

2. For any function $\xi(x)$ which has finite energy, we can decompose the function into a weighted sum of k_i. In other words, the set of function k_i is a complete and orthogonal basis set as

$$w_i = < \xi(x), k_i(x) >, \tag{3.11}$$

$$\xi(x) = \sum_{i=0}^{+\infty} \frac{w_i}{< k_i(x), k_i(x) >} k_i(x). \tag{3.12}$$

The proof can be found in Appendix B.

Let us look into the inner product in Eq. 3.11

$$\begin{aligned} w_i &= \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{+\infty}\xi(x)k_i^*(x)e^{-\frac{x^2}{2\sigma^2}}dx \\ &= \int_{-\infty}^{+\infty}\xi(x)p_i(x)\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}}e^{-jf_0x}dx \\ &= \int_{-\infty}^{+\infty}\xi(x)m_i(x)dx. \end{aligned} \tag{3.13}$$

This equation can also be represented in the Fourier domain as

$$\begin{aligned} w_i &= \mathcal{F}[\xi(x)] \otimes \mathcal{F}\left[p_i(x)\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}}\right] \\ &= \int_{-\infty}^{+\infty}\mathcal{F}[\xi(x)](f_0-f)^i e^{-(f_0-f)^2\sigma^2/2}df \\ &= (-1)^i\int_{-\infty}^{+\infty}\mathcal{F}[\xi(x)]M_i(f)df. \end{aligned} \tag{3.14}$$

In other words, the coefficients w_i from convolutions[1] of the signal with the moment filters can be used to reconstruct the original signal without loss. Therefore, the decomposition of the original signal by moment filters is complete and non-redundant. Eq. 3.13 and Eq. 3.14 show the computation of w_i in the spatial and the Fourier domains. We will use these two equivalent representations in our later analysis.

3.3 Other Important Properties of Moment Filters

Here are some useful properties of moment filters given without proof. Interested readers may refer to Appendix C.

- The moment filters are recursive in the Fourier domain, i.e.,

$$(f-f_0)M_i(f) = M_{i+1}(f). \tag{3.15}$$

[1] In this thesis, we use the term "convolution" loosely. In this example, it is actually a correlation. But since all filters are either symmetric or anti-symmetric, the only difference between correlation and convolution is the sign.

- The moment filters are recursive in the spatial domain, i.e.,

$$xm_i(x) = j(im_{i-1}(x) - \sigma^2 m_{i+1}(x)). \tag{3.16}$$

- The moment filters are recursive with respect to the differential operation, i.e.,

$$m_i'(x) = j(m_{i+1}(x) - f_0 m_i(x)). \tag{3.17}$$

- The instantaneous frequency is defined as the spatial derivative of the phase in [76, 24]. Replacing the peak frequency of the Gabor filter by instantaneous frequency will increase the accuracy as claimed in [24]. The instantaneous frequency can also be represented by moment filters as

$$\frac{d}{dx_0}\phi(x_0) = f_0 - \mathrm{Re}\left(\frac{w_1}{w_0}\right), \tag{3.18}$$

  where Re means the real part, w_0 is the convolution of the signal with the zeroth order moment filter, and w_1 is that with the first order moment filter.

- The stability criterion in the previous chapter was used to locate where the information extracted by finite-width filters is substantially different from that extracted by infinite-width filters. In other words, the extracted information is overwhelmingly contaminated by window effects when the stability criterion is not satisfied. We can represent the stability constraint as

$$T^2 = \left(\frac{1}{a}\frac{da}{dx_0}\right)^2 + \left(\frac{d\phi}{dx_0} - f_0\right)^2 = \frac{\| w_1 \|^2}{\| w_0 \|^2}. \tag{3.19}$$

The three recursive properties establish the interdependence among moment filters. They are the most important ones. The last two properties demonstrate that both instantaneous frequency and stability criterion techniques are actually *ad hoc* utilization of information from the first-order moment filter.

3.4 Moment Filters for Focus

3.4.1 Problem Definition

Depth from defocus methods obtain depth information using a quantitative measure of difference of focus between two images. For simplicity, we assume a Gaussian model

of the blurring function, even though, as we will see later, the approach is applicable to any blurring model. Let us denote two images as $i_0(x)$ and $i_1(x)$ in the spatial domain and $I_0(f)$ and $I_1(f)$ in the Fourier domain. The relations between these two images are:

$$i_1(x) = i_0(x) \otimes \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{x^2}{2\sigma_0^2}} = \int_{-\infty}^{+\infty} i_0(t) \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x-t)^2}{2\sigma_0^2}} dt \tag{3.20}$$

$$I_1(f) = I_0(f) e^{-f^2\sigma_0^2/2} \tag{3.21}$$

The difference of focus between two images is characterized by σ_0. If σ_0 is not a constant within the window, i.e., the surface is not front parallel, but changes linearly in the neighborhood of the window center $x = 0$, i.e.,

$$\sigma_0 = s(1 + \mu x), \tag{3.22}$$

we can expand the second integral term in Eq. 3.20 as

$$\frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x-t)^2}{2\sigma_0^2}} \approx \frac{1}{\sqrt{2\pi}s} e^{-\frac{(x-t)^2}{2s^2}} + \mu x \frac{(x-t)^2 - s^2}{\sqrt{2\pi}s^3} e^{-\frac{(x-t)^2}{2s^2}}, \tag{3.23}$$

where we truncated higher order terms of μx assuming μ is small. Replacing the above equation into Eq. 3.20 as

$$i_1(x) = i_0(x) \otimes \frac{1}{\sqrt{2\pi}s} e^{-\frac{x^2}{2s^2}} + \mu x \left(i_0(x) \otimes \frac{x^2 - s^2}{\sqrt{2\pi}s^3} e^{-\frac{x^2}{2s^2}} \right), \tag{3.24}$$

we can see that if σ_0 changes linearly, the relation between $i_0(x)$ and $i_1(x)$ is no longer a simple spatial convolution, and therefore cannot be represented in the Fourier domain as in Eq. 3.21. In fact, if we perform the infinite-duration Fourier transform on both sides of Eq. 3.24, the second term in the right side cannot be easily represented based on $I_0(f)$ because of the multiplication by x. It also shows that there exists no way to represent the shift variance problem in the same format as Eq. 3.21 no matter what transformation is used. Therefore, we need to solve the following two distinct problems:

- *Finite Window Problem*: Since Eq. 3.21 is based upon an infinite window, how can we compute σ_0 when the window is actually finite?
- *Shift Variance Problem*: Since Eq. 3.21 is based on the assumption that σ_0 is a constant within the window, how can we compute s in Eq. 3.22 when σ_0 changes linearly within the window?

Another way to understand Eq. 3.22 and 3.23 is that s represents the blurring difference at the center of the window and μ the gradient of the blurring difference within the window width. When the blurring difference s is converted into depth, the gradient μ is converted into the surface slope. Therefore, solving the shift variance problem is equivalent to simultaneously computing depth and shape.

In the following discussions we will only use a single Gabor filter with the peak frequency f_0 and the moment filters based on the Gabor filter. In practice, however, we can optimally merge results from multiple Gabor filters by Kalman filtering [80] or minimizing a least square objective function composed of multiple bands as explained later.

3.4.2 Finite Window Problem

In this section, we will limit our discussion to the situation when $\mu = 0$, i.e., σ_0 is a constant within the window. We define U_i and V_i as the convolution of the first and the second image with the ith order moment filter at $x = 0$, i.e.,

$$\begin{aligned} U_i &= \int_{-\infty}^{+\infty} i_0(x) m_i(x) dx \\ &= (-1)^i \int_{-\infty}^{+\infty} I_0(f) M_i(f) df, \end{aligned} \tag{3.25}$$

$$\begin{aligned} V_i &= \int_{-\infty}^{+\infty} i_1(x) m_i(x) dx \\ &= (-1)^i \int_{-\infty}^{+\infty} I_1(f) M_i(f) df \\ &= (-1)^i \int_{-\infty}^{+\infty} I_0(f) e^{-f^2\sigma_0^2/2} M_i(f) df. \end{aligned} \tag{3.26}$$

By Taylor expansion, in the neighborhood of f_0 we have

$$e^{-f^2\sigma_0^2/2} \approx e^{-f_0^2\sigma_0^2/2} \left(1 - f_0\sigma_0^2(f - f_0) - \frac{\sigma_0^2(1 - f_0^2\sigma_0^2)}{2}(f - f_0)^2\right), \tag{3.27}$$

where higher order terms are truncated.

Replacing Eq. 3.27 into Eq. 3.26 and using the recursiveness of moment filters in the Fourier domain, we obtain

$$\begin{aligned} V_0 &\approx \int_{-\infty}^{+\infty} I_0(f) \left(1 - f_0\sigma_0^2(f - f_0) - \frac{\sigma_0^2(1 - f_0^2\sigma_0^2)}{2}(f - f_0)^2\right) e^{-(f_0 - f)^2\sigma^2/2} df \\ & \quad e^{-f_0^2\sigma_0^2/2} \\ &= e^{-f_0^2\sigma_0^2/2}(U_0 + f_0\sigma_0^2 U_1 - \frac{\sigma_0^2(1 - f_0^2\sigma_0^2)}{2} U_2). \end{aligned} \tag{3.28}$$

Therefore, we can compute σ_0^2 from this equation:

$$\sigma_0^2 = \frac{2}{f_0^2}\left(\ln(U_0 + f_0\sigma_0^2 U_1 - \frac{\sigma_0^2(1 - f_0^2\sigma_0^2)}{2}U_2) - \ln V_0\right). \tag{3.29}$$

Comparing with the equation from Eq. 3.21 and Eq. 2.25, i.e.,

$$\sigma_0^2 = \frac{2}{f_0^2}(\ln I_0(f_0) - \ln I_1(f_0)), \tag{3.30}$$

we can see that if we make the assumption that the effects caused by window are negligible, i.e.,

$$\ln\frac{U_0}{V_0} = \ln\frac{I_0(f_0)}{I_1(f_0)}, \tag{3.31}$$

this may result in substantial errors if either U_1 or U_2 happens to be not very small in comparison with U_0.

A more intuitive way of looking at the difference between Eq. 3.29 and Eq. 3.30 is illustrated in Figure 3.4. The solid line is the transformation curve $e^{-f^2\sigma_0^2/2}$, with $\sigma_0 = 1.0$. The peak frequency f_0 is 0.6, and the passband of the Gabor filter is $(0.4, 0.8)$. The dotted curve is the zeroth order approximation, i.e., Eq. 3.30 is equivalent to using a constant to approximate the transformation curve within the passband. The first order approximation (dash-dot curve), i.e., Eq. 3.29 with only the linear term, is equivalent to using a line to approximate the transformation curve within the passband. The second order approximation (dashed curve), i.e., Eq. 3.29, is equivalent to using a quadratic curve to approximate the transformation curve within the passband. Generally, if we use up to the nth order moment filter in the Fourier domain, we are actually fitting the transformation curve by an nth order polynomial. Using a high order approximation will drastically decrease the error compared to the constant approximation.

The recursiveness of moment filters in the frequency domain plays the pivotal role in modelling the window effects. In fact, if we assume that the transformation curve could be approximated as, for example, a second order polynomial around the peak frequency f_0, i.e.,

$$T(f; p) = a_0(p) + a_1(p)(f - f_0) + a_2(p)(f - f_0)^2, \tag{3.32}$$

where the coefficients a_0, a_1 and a_2 are functions of p, we can use the recursiveness of moment filters in the frequency domain, i.e.,

$$(f - f_0)M_i(f) = M_{i+1}(f), \tag{3.33}$$

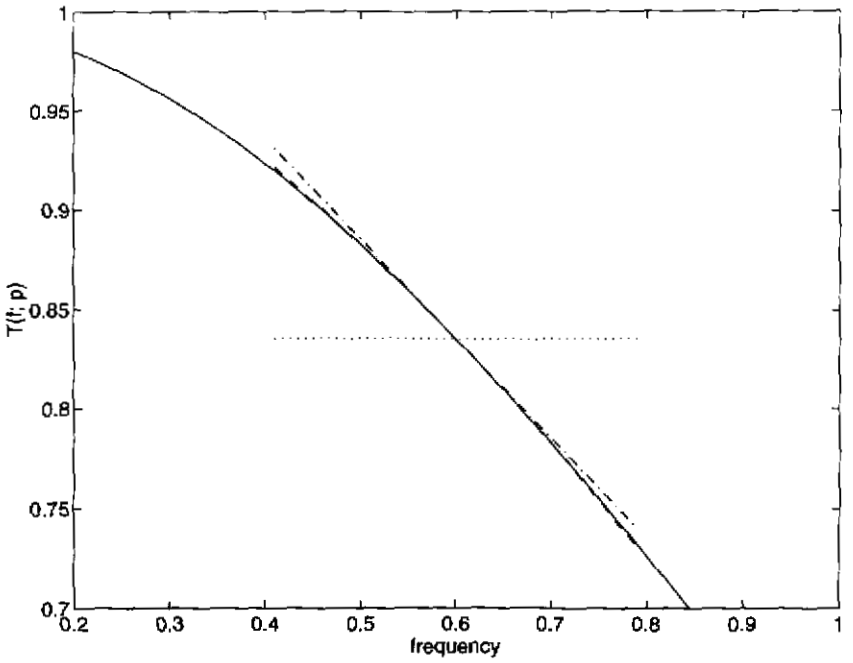


Figure 3.4: Polynomial Fitting of the Transformation Curve

to model the window effects:

$$\begin{aligned} T(f;p)M_0(f) &= \left(a_0(p) + a_1(p)(f-f_0) + a_2(p)(f-f_0)^2\right) M_0(f) \\ &= a_0(p)M_0(f) + a_1(p)M_1(f) + a_2(p)M_2(f). \end{aligned} \tag{3.34}$$

Now, let us look back at the truncation in Eq. 3.27. We truncated terms of $\frac{(f-f_0)^i}{i!}$ for $i > 2$. Correspondingly, we truncated terms of $\frac{U_i}{i!}$ for $i > 2$ in Eq. 3.28. These truncations are valid only if $\frac{U_i}{i!}$ generally decreases in magnitude as i increases. Since U_i is generated by convolving images with $m_i(x)$ as in Eq. 3.13, the expected magnitude of U_i will be proportional to the square root of the energy of the filter:

$$\begin{aligned} E_i &= \left(\int_{-\infty}^{+\infty} \| m_i(x) \|^2 \, dx\right)^{1/2} \\ &= \left(\frac{1}{2\pi}\int_{-\infty}^{+\infty} \| M_i(f) \|^2 \, df\right)^{1/2} \\ &= \sqrt{\frac{\Gamma(i+\frac{1}{2})}{2\pi}}\sigma^{-(i+1/2)}, \end{aligned} \tag{3.35}$$

where $\Gamma(x)$ is the Gamma function. Apparently, the expected magnitude of $\frac{U_i}{i!}$ will decrease monotonically as long as the window width σ is much larger than one pixel

size, which is usually satisfied. Figure 3.3 shows the decreasing magnitude of moment filters.

Solving Eq. 3.29 is not as trivial as solving Eq. 3.30. But since the expected magnitude of U_0 is larger than those of U_1 and U_2, we propose to use Eq. 3.29 as an iterative procedure to find the qth time estimation of $\sigma_0^{2(q)}$ such as:

$$\begin{aligned}\sigma_0^{2(q)} &= \frac{2}{f_0^2}\left(\ln(U_0 + f_0\sigma_0^{2(q-1)}U_1 - \frac{\sigma_0^{2(q-1)}(1 - f_0^2\sigma_0^{2(q-1)})}{2}U_2) - \ln V_0\right) \\ &= T(\sigma_0^{2(q-1)}).\end{aligned} \tag{3.36}$$

And the sufficient condition for the iteration to converge is

$$\| T'(\sigma_0^{2(q)}) \| = \| \frac{2}{f_0^2}\frac{f_0U_1 - \frac{1}{2}(1 - 2f_0^2\sigma_0^{2(q)})U_2}{U_0 + f_0\sigma_0^{2(q)}U_1 - \frac{1}{2}\sigma_0^{2(q)}(1 - f_0^2\sigma_0^{2(q)})U_2} \| < 1. \tag{3.37}$$

For some locations, the convergence condition is not satisfied. We can think of the convergence condition as a generalized stability criterion. In fact, if we consider only the linear term in the Taylor expansion, this convergence condition is the same as the stability criterion in the previous chapter.

3.4.3 Shift Variance Problem

When $\mu \neq 0$ in Eq. 3.22 and Eq. 3.23, we must consider the effect of the shift variance. Let us define

$$V_i = \int_{-\infty}^{+\infty} i_1(x)m_i(x)dx, \tag{3.38}$$

$$\mathcal{V}_i = \int_{-\infty}^{+\infty}\left(\int_{-\infty}^{+\infty} i_0(t)\frac{1}{\sqrt{2\pi}s}e^{-\frac{(x-t)^2}{2s^2}}dt\right)m_i(x)dx. \tag{3.39}$$

In other words, $\mathcal{V}_i$ represents the filter output of a virtual image which is shift invariant blurring of $i_0(x)$, while V_i represents the filter output of the original real image $i_1(x)$, which is shift variant blurring of $i_0(x)$. In order to use Eq. 3.36 to compute the blurring difference at the center of the window accurately, we need to compute $\mathcal{V}_i$ from V_i. Replacing Eq. 3.23 into Eq. 3.38, we have

$$\begin{aligned}V_i = &\int_{-\infty}^{+\infty}\left(i_0(x) \otimes \frac{1}{\sqrt{2\pi}s}e^{-\frac{x^2}{2s^2}}\right)m_i(x)dx + \\ &\mu\int_{-\infty}^{+\infty}\left(i_0(x) \otimes \frac{x^2 - s^2}{\sqrt{2\pi}s^3}e^{-\frac{x^2}{2s^2}}\right)xm_i(x)dx.\end{aligned} \tag{3.40}$$

Using the recursiveness of the moment filters in the spatial domain Eq. 3.16 in Eq. 3.40, and converting it into the Fourier domain, we have

$$\begin{aligned} V_i &= \mathcal{V}_i - \mu \int_{-\infty}^{+\infty} I_0(f)(-f^2 s^2 e^{-f^2 s^2/2}) j(\sigma^2 M_{i+1}(f) - i M_{i-1}(f)) df \\ &= \mathcal{V}_i + j s^2 \mu(\sigma^2(\mathcal{V}_{i+3} + 2f_0\mathcal{V}_{i+2} + f_0^2\mathcal{V}_{i+1}) - \\ &\quad i(\mathcal{V}_{i+1} + 2f_0\mathcal{V}_i + f_0^2\mathcal{V}_{i-1})). \end{aligned} \tag{3.41}$$

In other words, the effect of the linear shift variance is that the original filter outputs $\mathcal{V}_i$ of the shift invariance transformation are recombined linearly to form V_i.

If the magnitude of μ is small, we can then ignore the terms of second or higher order of μ, and have the following approximation from Eq. 3.41:

$$\begin{aligned} \mathcal{V}_i &\approx V_i - j s^2 \mu(\sigma^2(V_{i+3} + 2f_0 V_{i+2} + f_0^2 V_{i+1}) - \\ &\quad i(V_{i+1} + 2f_0 V_i + f_0^2 V_{i-1})) \\ &= V_i + s^2 \mu P_i, \end{aligned} \tag{3.42}$$

where P_i is used for simplicity of notation. In other words, when μ is small, we can reverse the effect of the linear shift variance, and obtain the original filter output $\mathcal{V}_i$ when the transformation is shift invariant. Once we have computed $\mathcal{V}_i$, we can compute s by the iteration proposed in the previous section:

$$s^2 = \frac{2}{f_0^2}\left(\ln(U_0 + f_0 s^2 U_1 - \frac{1}{2}s^2(1 - f_0^2 s^2)U_2) - \ln(V_0 + s^2 \mu P_0)\right). \tag{3.43}$$

Now the problem is that we do not know μ. Therefore, we also need an algorithm to estimate μ. There are two ways to estimate μ. One is to estimate s and μ at the same time by minimizing the sum of the norm difference of all frequency bands:

$$\sum_{\text{All bands}} \| V_0 + s^2 \mu P_0 - e^{-f_0^2 s^2/2}(U_0 + f_0 s^2 U_1 - \frac{s^2(1 - f_0^2 s^2)}{2} U_2) \|^2 . \tag{3.44}$$

The other method is to take the ratio of the zeroth and the first moments using Eq. 3.28 and Eq. 3.42 in one single frequency band:

$$\frac{V_1 + s^2 \mu P_1}{V_0 + s^2 \mu P_0} \approx \frac{U_1 + f_0 s^2 U_2 - \frac{1}{2}s^2(1 - f_0^2 s^2)U_3}{U_0 + f_0 s^2 U_1 - \frac{1}{2}s^2(1 - f_0^2 s^2)U_2} \tag{3.45}$$

Combining Eq. 3.43 and Eq. 3.45 together, we have two equations for two unknowns, i.e., s and μ. For simplicity of notation, Letting $u = s^2$ and $v = s^2\mu$,

Eq. 3.43 and Eq. 3.45 can be simplified as

$$u = \mathcal{M}(u, v), \tag{3.46}$$

$$v = \mathcal{N}(u), \tag{3.47}$$

where $\mathcal{M}$ and $\mathcal{N}$ are two functions inferred from Eq. 3.43 and Eq. 3.45. We can iteratively solve for u and v according to the two above equations. Suppose the qth time estimation of u and v are $u^{(q)}$ and $v^{(q)}$ respectively, we can approach the true value of u and v by iteration as:

$$u^{(q)} = \mathcal{M}(u^{(q-1)}, v^{(q-1)}) \tag{3.48}$$

$$v^{(q)} = \mathcal{N}(u^{(q-1)}). \tag{3.49}$$

The sufficient condition for the iteration to converge is that the magnitudes of both the complex eigenvalues E_0 and E_1 of the Jacobian matrix J are less than one, i.e.,

$$\| E_i \| < 1.0, i = 0, 1, \tag{3.50}$$

where

$$J = \begin{bmatrix} \frac{\partial \mathcal{M}}{\partial u} & \frac{\partial \mathcal{M}}{\partial v} \\ \frac{\partial \mathcal{N}}{\partial u} & 0 \end{bmatrix}. \tag{3.51}$$

Therefore, when the convergence condition Eq. 3.50 is satisfied, we can iteratively estimate depth and slope simultaneously as in Eq. 3.48 and Eq. 3.49.

3.4.4 Error Estimation

Error estimation is necessary if we need to know how accurate our results are or how to merge results from different frequency bands sequentially. The convergence conditions (Eq. 3.37 and Eq. 3.50) can identify cases of instability. But for those do converge, we still need to quantify the dependency of errors on the image contents in order to optimally merge results computed from different frequency bands.

Before we jump into the estimation of errors, we need to be clear in math. Generally, both Eq. 3.43 and Eq. 3.45 will return complex numbers for u and v, which in practice, need to be real numbers. Because we know that the imaginary part of computed u and v should be zero, we simply drop the computed imaginary parts, and therefore keep the u and v meaningful. As we are concerned about the error estimation of u and v, we will have the problem of estimating real-number errors caused by complex-number sources.

Let $\delta = \delta_r + j\delta_i$ as a zero-mean complex error source, whose magnitude $E[\delta\delta^*] = E[\delta_r^2] + E[\delta_i^2] = 2E[\delta_r^2]$ (we assume the error source is non-directional), and

$$\begin{aligned} u &= A\delta, \\ v &= B\delta, \end{aligned}$$

where A and B are constants, we then have

$$E[u_r v_r] = E[\frac{A\delta + A^*\delta^*}{2}\frac{B\delta + B^*\delta^*}{2}] = \mathrm{Re}(AB^*)E[\delta\delta^*]/2. \tag{3.52}$$

Generally, if u, v and δ are vectors, and A and B are constant matrices, we have

$$E[\mathbf{u}_r\mathbf{v}_r^T] = \mathrm{Re}(\mathbf{A}E[\delta\delta^{*T}]\mathbf{B}^{*T})/2. \tag{3.53}$$

There are two types of error sources we need to take into consideration. The first is the noises in images, and the other is the truncation error in Eq. 3.27 and Eq. 3.23. To simplify the problem, we make the following assumptions:

1. The error caused by truncation in Eq. 3.23 is negligible.

2. The noise is white additive noise.

Based on the two error sources, i.e., the truncation error in Eq. 3.23 and white noise in images, the uncertainty in estimating s and μ from Eq. 3.44 could be computed by χ^2 methods [72]. And the uncertainty in estimating u and v in Eq. 3.48 and Eq. 3.49 can be represented as (Appendix D)

$$\begin{bmatrix} du \\ dv \end{bmatrix} = \mathbf{A}_f\mathbf{W}_f, \tag{3.54}$$

where $\mathbf{A}_f$ is a 2×6 complex matrix, and $\mathbf{W}_f$ is the 6×1 error vector. Applying Eq. 3.53 to Eq. 3.54, we have the covariance of u_r and v_r, which are real parts of complex number u and v, as

$$E\begin{bmatrix} du_r du_r & du_r dv_r \\ du_r dv_r & dv_r dv_r \end{bmatrix} = \mathrm{Re}(\mathbf{A}_f E[\mathbf{W}_f\mathbf{W}_f^{*T}]\mathbf{A}_f^{*T})/2, \tag{3.55}$$

where the covariance matrix $E[\mathbf{W}_f\mathbf{W}_f^{*T}]$ is a 6×6 diagonal real matrix. Once we compute s and μ and their uncertainty estimation from every individual frequency band, we could then combine results from all bands optimally using Kalman filtering [80].

3.5 Moment Filters For Stereo

3.5.1 Problem Definition

Depth from stereo methods obtain depth information by measuring the *local* shift between two images. Assuming $i_0(x)$ and $i_1(x)$ are the two images in the spatial domain, and $I_0(f)$ and $I_1(f)$ are in the Fourier domain, we have the following relations

$$i_1(x) = i_0(x + D_0), \tag{3.56}$$

$$I_1(f) = I_0(f)e^{jfD_0}. \tag{3.57}$$

The shift between two images is characterized by the disparity D_0. The above relation in the Fourier domain is valid only when D_0 is a constant. If D_0 varies linearly in the neighborhood of the window center $x = 0$, i.e.,

$$D_0 = D + \mu x, \tag{3.58}$$

we then have

$$i_0(x + D_0) \approx i_0(x + D) + \mu x \frac{d}{dx} i_0(x + D), \tag{3.59}$$

where we truncated terms of higher order of μ based on assumption that μ is small.

The problems we need to solve are:

- *Finite Window Problem*: Since Eq. 3.57 is based upon an infinite window, how can we compute D_0 with high precision when the window is finite?
- *Shift Variance Problem*: Since Eq. 3.57 is based upon the assumption that D_0 is a constant within the window, how can we compute D when D_0 varies within the window as in Eq. 3.58?

3.5.2 Finite Window Problem

In this section, we limit our discussion to the situation when $\mu = 0$, i.e., the disparity D_0 is indeed constant within the window. We define U_i and V_i as the result of convolving the two images with the ith moment filter, i.e.,

$$\begin{aligned} U_i &= \int_{-\infty}^{+\infty} i_0(x) m_i(x) dx \\ &= (-1)^i \int_{-\infty}^{+\infty} I_0(f_0) M_i(f) df \end{aligned} \tag{3.60}$$

$$\begin{aligned} V_i &= \int_{-\infty}^{+\infty} i_1(x) m_i(x) dx \\ &= (-1)^i \int_{-\infty}^{+\infty} I_0(f) e^{jfD_0} M_i(f) df \end{aligned} \tag{3.61}$$

In the neighborhood of f_0 the transformation curve can be expanded according to Taylor expansion

$$e^{jfD_0} \approx e^{jf_0D_0} \left(1 + jD_0(f - f_0) - \frac{D_0^2}{2}(f - f_0)^2\right), \tag{3.62}$$

where higher order terms are truncated.

Replacing Eq. 3.62 into Eq. 3.61 and using the recursiveness of the moment filter in the Fourier domain, we have

$$V_0 = e^{jf_0D_0}(U_0 - jD_0U_1 - \frac{D_0^2}{2}U_2). \tag{3.63}$$

Therefore, we have an equation to compute D_0 as

$$D_0 = \frac{j}{f_0}\left(\ln(U_0 - jD_0U_1 - \frac{D_0^2}{2}U_2) - \ln V_0\right). \tag{3.64}$$

As we showed in focus, the difference between Eq. 3.64 and Eq. 3.57 is again that the moment filter approach uses a second order polynomial to approximate the transformation curve e^{jfD_0} in the neighborhood of f_0, while the traditional approach uses a constant $e^{jf_0D_0}$ to approximate the curve. As we can see, only when $U_1 = U_2 = 0$, is Eq. 3.64 the same as Eq. 3.57. Therefore, when the magnitude of U_1 or U_2 is not small, using Eq. 3.57 to compute D_0 will result into large errors.

Since the traditional Gabor filter approach is based on the assumption $U_1 = 0$, it is not a surprise to learn that the stability constraint used in the previous chapter is actually the magnitude ratio of U_1 and U_0, i.e.,

$$\sqrt{\left(\frac{\partial\phi}{\partial x} - f_0\right)^2 + \left(\frac{1}{a}\frac{\partial a}{\partial x}\right)} = \frac{\| U_1 \|}{\| U_0 \|}, \tag{3.65}$$

where ϕ is the phase, and a is the magnitude. Eq. 3.65 also justified the so-called phase singularity constraint which determines when $U_1(f_0)$ is so large compared with $U_0(f_0)$ that Eq. 3.57 is not even approximately valid. We can see that such a situation is exactly what Eq. 3.64 characterizes.

While the stability constraint locates the information severely contaminated by window effects, it does not find a specific way to correct the error caused by a finite

window. Our moment filter approach can correct this error since the window effects are analytically expressed as high order terms in Eq. 3.63. Just as we did in the problem of depth from defocus, we can use Eq. 3.64 in an iterative procedure to find the qth time estimation of D_0 as

$$\begin{aligned} D_0^{(q)} &= \frac{j}{f_0}\left(\ln(U_0 - jD_0^{(q-1)}U_1 - \frac{(D_0^{(q-1)})^2}{2}U_2) - \ln V_0\right) \\ &= T(D_0^{(q-1)}). \end{aligned} \tag{3.66}$$

And the sufficient condition for the iteration to converge is

$$\| \frac{d}{dD_0} T(D_0^{(j)}) \| < 1. \tag{3.67}$$

3.5.3 Relations to the Phase-Based Method

The phase-based approach to stereo has been advocated recently by some researchers [27, 76, 91]. This approach simply divides the phase difference by the peak frequency or the instantaneous frequency to infer the disparity. Fleet in [27] showed that when the stability criterion is small, using the instantaneous frequency would produce better results than using the peak frequency.

Suppose we truncate the Taylor expansion after the linear term, then Eq. 3.63 becomes

$$V_0 = e^{jf_0D_0}(U_0 - jD_0U_1). \tag{3.68}$$

Rewriting the above equation, we then have

$$jf_0D_0 = \ln V_0 - \ln(U_0 - jD_0U_1). \tag{3.69}$$

Because the stability constraint guarantees that $\| U_1 \| \ll \| U_0 \|$, the second term of the right side in the above equation can be approximated as

$$\ln(U_0 - jD_0U_1) \approx \ln U_0 - jD_0\frac{U_1}{U_0}. \tag{3.70}$$

Replacing Eq. 3.70 into Eq. 3.69, after some manipulations, we obtain

$$(f_0 - \frac{U_0}{U_1})D_0 = j\ln U_0 - j\ln V_0. \tag{3.71}$$

Taking the real part of the both sides in the above equation, and referring to the definition of the instantaneous frequency in Eq. 3.18, we can see that the above equation *literally* becomes that D_0 multiplied by the instantaneous frequency equals the phase difference. Therefore, we can regard the phase-based approach as a special case of the moment filter approach we propose here.

3.5.4 Shift Variance Problem

When $\mu \neq 0$ in Eq. 3.58 and Eq. 3.59, we must consider the effect of shift variance, i.e., foreshortening in stereo. Let

$$V_i = \int_{-\infty}^{+\infty} i_1(x) m_i(x) dx, \tag{3.72}$$

$$\mathcal{V}_i = \int_{-\infty}^{+\infty} i_0(x+D) m_i(x) dx \tag{3.73}$$

$$= (-1)^i \int_{-\infty}^{+\infty} I_0(f) e^{jfD} M_i(f) df. \tag{3.74}$$

In other words, $\mathcal{V}_i$ represents the filter output of a virtual image which is uniformly translated from $i_0(x)$, while V_i represents the filter output of the original real image $i_1(x)$, which is non-uniformly translated from $i_0(x)$. In order to use Eq. 3.66 to compute D accurately, we need to estimate $\mathcal{V}_i$ from V_i. Replacing Eq. 3.59 into Eq. 3.72, we have

$$V_i \approx \int_{-\infty}^{+\infty} i_0(x+D) m_i(x) dx + \mu \int_{-\infty}^{+\infty} \frac{d i_0(x+D)}{dx} x m_i(x) dx. \tag{3.75}$$

Using the recursiveness of the moment filter in the spatial domain, we can rewrite Eq. 3.75 as

$$V_i = \mathcal{V}_i - \mu(i(f_0 \mathcal{V}_{i-1} - \mathcal{V}_i) - \sigma^2(f_0 \mathcal{V}_{i+1} - \mathcal{V}_{i+2})). \tag{3.76}$$

If μ is small, we can ignore the second or higher order terms when we reverse the above equation, i.e., we can have the following approximation:

$$\begin{aligned} \mathcal{V}_i &\approx V_i + \mu(i(f_0 V_{i-1} - V_i) - \sigma^2(f_0 V_{i+1} - V_{i+2})) \\ &= V_i + \mu P_i, \end{aligned} \tag{3.77}$$

where P_i is used for simplicity of notation.

Again, there are two ways to compute μ. One is by minimizing the sum of norm difference of all frequency bands:

$$\sum_{\text{All bands}} \| V_0 + \mu P_0 - e^{jf_0 D}(U_0 - jDU_1 - \frac{D^2}{2} U_2) \|^2 . \tag{3.78}$$

And similarly the uncertainty estimation of D and μ could be done by the χ^2 method once the uncertainty of filter output are estimated.

The other way to compute μ is by combining Eq. 3.64 and Eq. 3.77. We have the equations to solve D and μ,

$$D = \frac{j}{f_0}\left(\ln(U_0 - jDU_1 - \frac{D^2}{2}U_2) - \ln(V_0 + \mu P_0)\right), \tag{3.79}$$

$$\frac{V_1 + \mu P_1}{V_0 + \mu P_0} \approx \frac{U_1 - jDU_2 - \frac{1}{2}D^2U_3}{U_0 - jDU_1 - \frac{1}{2}D^2U_2}. \tag{3.80}$$

The same iterative scheme we used in focus to solve s and μ can also be applied to the above equations to solve D and μ simultaneously.

There are two types of error sources that we need to take into consideration in error estimation as we did in focus. The first one is noises in images, and the other is the truncation error in Taylor expansion. To simplify the problem, we make the following assumptions:

1. The error caused by truncation in Eq. 3.59 is negligible.

2. The noise is white additive noise.

Based on the same two types of error sources, the uncertainty in estimating D and μ can also be estimated just as was done in the focus method. As details can be found in (Appendix D), the errors of D and μ can be represented as

$$\begin{bmatrix} dD \\ d\mu \end{bmatrix} = \mathbf{A}_s\mathbf{W}_s, \tag{3.81}$$

where $\mathbf{A}_s$ is a 2×6 complex matrix, and $\mathbf{W}_s$ is a 6×1 error vector. Applying Eq. 3.53 to Eq. 3.81, we have the covariance of D_r and μ_r as

$$E\begin{bmatrix} dD_r dD_r & dD_r d\mu_r \\ dD_r d\mu_r & d\mu_r d\mu_r \end{bmatrix} = \mathrm{Re}(\mathbf{A}_s E[\mathbf{W}_s\mathbf{W}_s^{*T}]\mathbf{A}_s^{*T})/2, \tag{3.82}$$

where the covariance matrix $E[\mathbf{W}\mathbf{W}^{*T}]$ is a 6×6 diagonal real matrix. Once we compute s and μ and their uncertainty estimation from every individual frequency band, we could then combine results from all bands optimally using Kalman filtering [80].

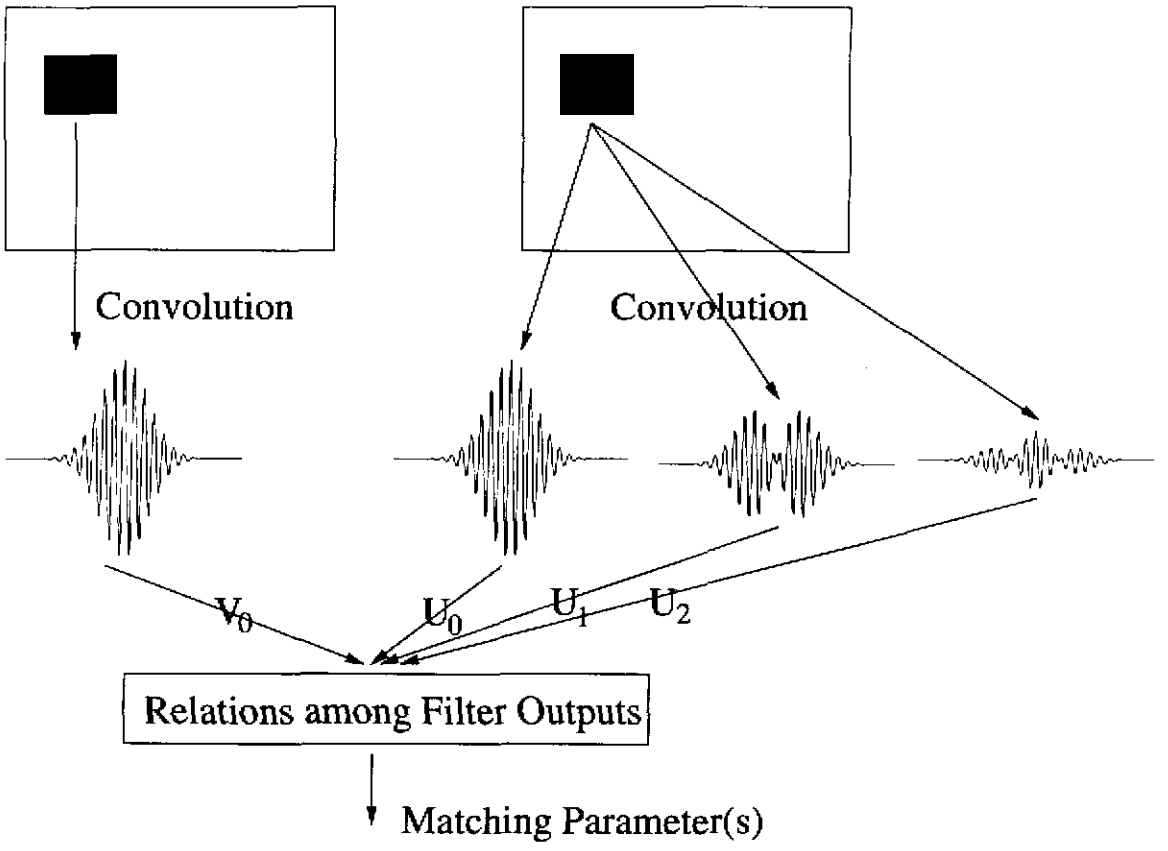


Figure 3.5: Modelling Window Effects

3.6 Moment Filters for the General Image Matching Problem

The power of moment filters lies in their ability to model window effects and shift variance effects analytically. As shown in examples of focus and stereo, we can eliminate the errors caused by those negative effects if we take those effects into consideration.

- *Window Effects*: The window effects are represented as linear combinations of high order moment filter outputs as in Figure 3.5. The window effects are eliminated numerically since the energy of moment filters decrease rapidly as their order increases. As shown in the previous sections, the ability to model window effects is the direct result of the recursiveness of the moment filters in the frequency domain.

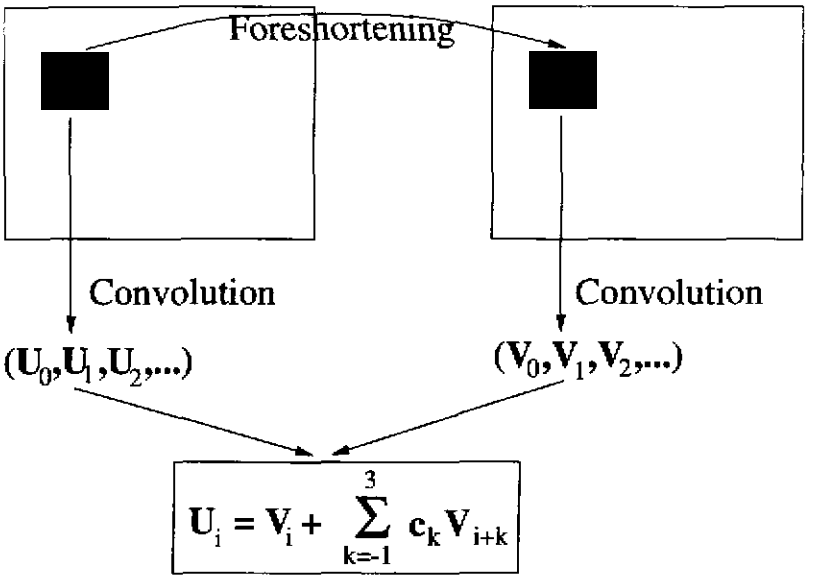


Figure 3.6: Modelling Shift Variance Effects

- *Shift Variance Effects*: The shift variance effects can also be represented as linear combinations of moment filter outputs as in Figure 3.6, in which the coefficients are functions of the shift variance parameters. As shown in the previous sections, the ability to model shift variance effects is the direct result of the recursiveness of the moment filters in the spatial domain.

3.7 New Interpretations of Window Schemes

In Chapter 2 and other literature on wavelets such as [74], we justified the variable window scheme based upon the observation that a fixed window scheme makes the number of "wavelets" in filters different, and therefore the expected "quality" of results from different frequency bands are different. In other words, the finite fixed window size has window effects of different magnitude on different bands. Since we can model window effects analytically now, it is time that we quantitatively prove this intuition.

By now, we have quantitatively represented the window effects as higher order terms in the Taylor expansions. Suppose that the Taylor expansion is truncated after the first term, i.e., the window effects are ignored; we can expect that the error caused

by the truncation is

$$\begin{aligned}\delta(\sigma_0^2) &= \frac{2}{f_0^2}\left(\ln(U_0 + f_0\sigma_0^2 U_1) - \ln U_0\right) \\ &\approx 2\sigma_0^2\frac{U_1}{U_0 f_0}\end{aligned} \tag{3.83}$$

in focus (referring to Eq. 3.29), or

$$\begin{aligned}\delta(D_0) &= \frac{j}{f_0}\left(\ln(U_0 + jD_0 U_1) - \ln U_0\right) \\ &\approx jD_0\frac{U_1}{U_0 f_0}\end{aligned} \tag{3.84}$$

in stereo (referring to Eq. 3.64).

From Eq. 3.35, we know that the ratio between the expected magnitude of U_0 and U_1 is

$$E\left[\frac{\| U_1 \|}{\| U_0 \|}\right] = \frac{E_1}{E_0} = \frac{.1}{\sqrt{2}\sigma}, \tag{3.85}$$

where E_1 and E_0 are the energy of the moment filters as in Eq. 3.35. Therefore, the expected error caused by a finite window width is

$$E[\delta(\sigma_0^2)] \approx \sqrt{2}\sigma_0^2\frac{1}{f_0\sigma} \tag{3.86}$$

for focus, or

$$E[\delta(D_0)] \approx jD_0\frac{1}{\sqrt{2}f_0\sigma} \tag{3.87}$$

for stereo. If we use the variable window scheme as in the previous chapter, i.e., $f_0\sigma$ is a constant, the errors caused by window effects are statistically of the same magnitude for different frequency bands. In other words, the expected errors are functions of only $f_0\sigma$, which is a constant in the variable window scheme but not in the fixed window scheme. Therefore, the expected quality of results from different filters is the same for the variable window scheme if window effects are ignored.

3.8 Experiments

3.8.1 Sampling of the Frequency Domain

Since the information in an image is usually distributed across all frequency bands, we have to use multiple Gabor filters and their accompanying moment filter sets in

order to sample the frequency domain. At this point, the sampling of the frequency domain by Gabor filters is still arbitrary. In our 2D implementation of the moment filter approach, the sampling is done in polar coordinates, i.e., the frequency domain is sampled in radial and angular directions. The angular sampling rate is one per 30 degrees, and the radial sampling rate is chosen as one per $0.7/\sigma$, where σ is the window size parameter. The minimum frequency is one-tenth of the Nyquist frequency, and we use only the first 48 samples in the polar sampling. Therefore, we have a total of 48 Gabor filters and 48 moment filter sets. In the experiments below, we use moment filters up to the 4th order.

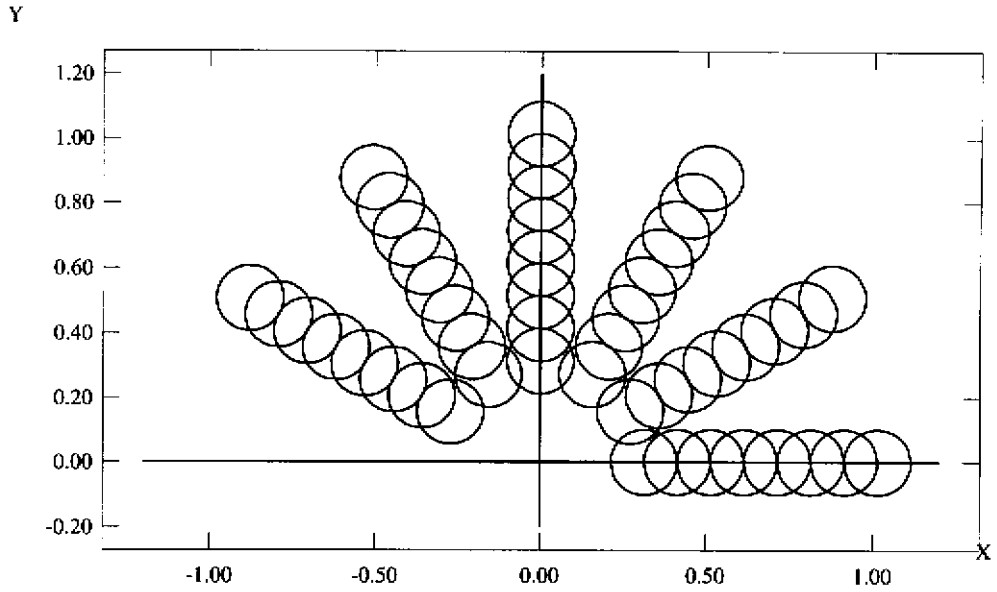
Though the moment filter approach can be applied under any frequency sampling scheme, we have to choose one for experimentation. There are two window schemes which are related to the radial frequency sampling. In the fixed window scheme, because σ are the same for all Gabor filters, the radial frequency sampling is uniform. In the variable window scheme,

$$\sigma = k\lambda, \tag{3.88}$$

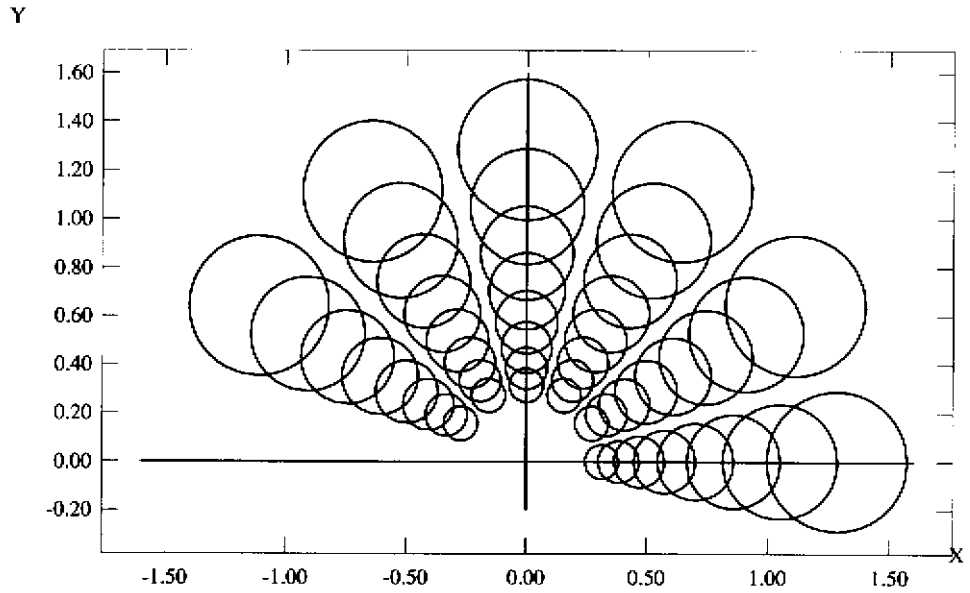
where k is a constant and λ is the peak wavelength of the Gabor filter, the radial frequency sampling is nonuniform. We will not compare these two window schemes here as we already did in the previous chapter, but rather will provide two choices for the convenience of experimentation. Figure 3.7 illustrates the sampling of the frequency plane under these two window schemes. The circles represent the effective bandwidth of Gabor filters.

In our experiments, we integrate results from different frequency bands sequentially using Kalman filters. Another possibility is to use all frequency bands together to compute the results by least square minimization as explained earlier. Due to the large amount of memory the latter method may require, we choose to use the sequential combination method.

The main difficulty in implementing the filter-based approach is the enormous amount of computation and memory required if we need to sample many frequency bands. We believe that, as parallel computers become more powerful, this problem will be much less severe. But for now we use only 48 filters, a very sparse set of samples in the frequency domain. We believe that if we sample the frequency domain in a denser fashion, we could further improve the results reported here. Current running time on MasPar is about twenty minutes for a 480×512 image pair.



Fixed Window Scheme



Variable Window Scheme

Figure 3.7: Frequency Sampling

3.8.2 Focus

We implement the moment filter approach for the focus problem using the third order Taylor expansion in Eq. 3.27. Because computing P_1 in Eq. 3.45 requires the fourth order moment filter, we need up to the fourth order moment filters for every Gabor filter. To accommodate numerical errors, the convergence thresholds in Eq. 3.37 and Eq. 3.50 were set to 0.9. Results from different frequency bands were merged optimally through Kalman filtering as in [80].

Subbarao's algorithm in [82] follows directly from Eq. 3.21, which can be regarded as a special case of our moment filter approach in which the Taylor expansion in Eq. 3.27 uses only the constant term. However, Subbarao's algorithm cannot be extended to deal with the shift variance problem.

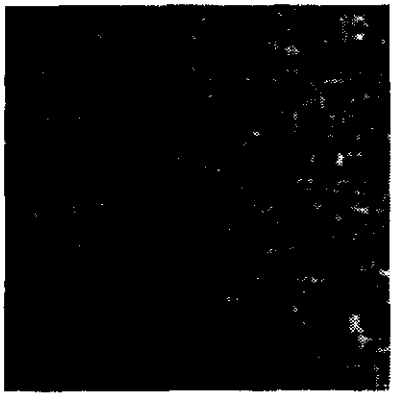
We implemented both algorithms under the variable window scheme, in which k in Eq. 3.88 was 0.8. Our experiments used both synthetic and real images. The original image is shown in Figure 3.8. We uniformly blurred the original texture image (Figure 3.8) by a Gaussian function with parameter $\sigma_0 = 1.0$ (in pixel width) to get the first synthetic image pair. We also non-uniformly blurred the original image by a Gaussian function with parameter σ_0 that increased linearly as the column increased to get the second synthetic image pair.

Since in 2D images, in Eq. 3.22, the slope of a surface has to be represented by a 2D vector instead of a scalar, we have to solve equations with three unknowns in 2D cases instead of two unknowns in 1D cases.

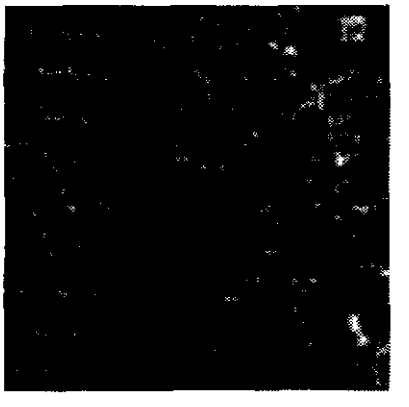
We compare three different algorithms: Subbarao's algorithm (SA), the moment filter algorithm without slope estimations (MFF1), and the moment filter algorithm with slope estimations (MFF2). Table 3.1 shows the root mean square errors of the synthetic image pairs using these algorithms. For the first pair, MFF1 and MFF2 show almost no error at all, while SA has a significant amount of error. For the second pair, the RMS error of SA is 4 times larger than that of MFF1, and 27 times larger than that of MFF2. We can see that the main source of error in Subbarao's algorithm is the truncation of information from moment filters. When the surface is slanted, i.e., foreshortening happens, the additional error depends on how slanted the surface is. The MFF2 algorithm can correct the errors caused by the foreshortening.

To illustrate the precision of different algorithms: Figure 3.9 shows the computed local blurring difference σ_0^2 of the first image pair; and Figure 3.10 shows the computed local blurring difference σ_0^2 of the second pair. In addition, Figure 3.11 illustrates the

Original Image



Uniformly Blurred Image



Nonuniformly Blurred Image

Figure 3.8: Synthetic Texture Images for Focus

	SA	MFF1	MFF2
The First Image Pair	0.070518	0.000749	0.000316
The Second Image Pair	0.190362	0.046419	0.007976

Table 3.1: RMS Errors of Different Algorithms

slope vectors computed by MFF2 for the second image pair. Note that the slope vector has been scaled by σ_0^2 on the left side of Eq. 3.47, which causes the magnitudes of slope vectors to increase. We can see that MFF2 can not only correct the errors caused by the foreshortening, but also estimate the slope of the surface.

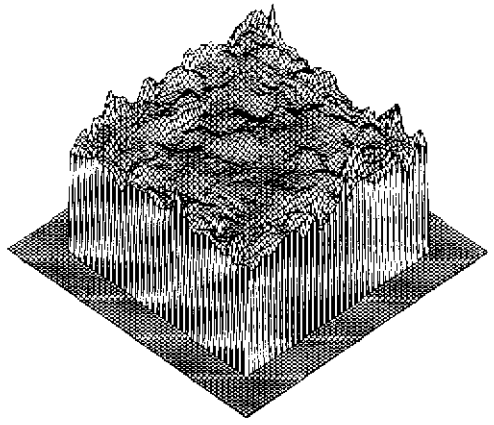
The real image pair is shown in Figure 3.12. These two pictures were taken by the Photometrics camera [97] under different aperture and exposure time settings. The toy house is about two meters away from the camera, and the size of the house itself is about 2.5 inches wide, 3.0 inches long, and 4.5 inches high. We compensated for the optical vignetting and different exposure amount between two images by using a uniform illumination source. Uniform illumination was achieved by putting a diffuser in front of the camera, and a light source far away.

Figure 3.13 shows the computed blurring difference of the house image pair using Subbarao's algorithm and MFF1. We can see that the result from SA is so noisy that the shape of the house can not be observed with confidence. But the shape of the house from MFF1 is clearly visible though the discontinuity generates a lot of spikes. Finally Figure 3.14 shows the result using MFF2. We can see that the computed blurring difference is even more smooth. Most of the slope vectors have the correct direction despite the very small slope ($\mu \approx 0.002$ in Eq. 3.22); and the discontinuity generates very large slope vectors which could potentially be used to locate the discontinuity in later processing stages.

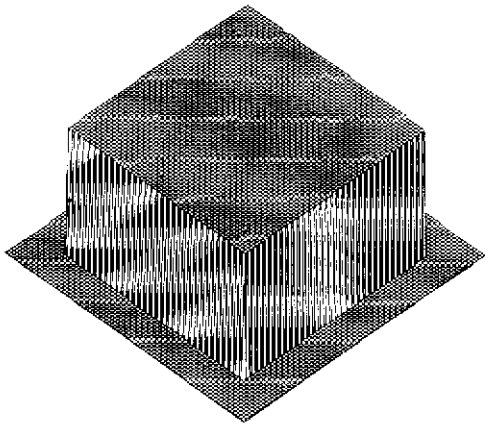
3.8.3 Stereo

Though filter-based stereo algorithms may be able to alleviate the mismatch problem, our concern here is focused on the precision of subpixel registration. We will compare our algorithms with Kanade and Okutomi's adaptive window scheme [41, 42]. A simple SSD-based matching is used to generate pixel resolution disparity, and that same initial disparity is used by both our algorithms and Kanade and Okutomi's algorithm.

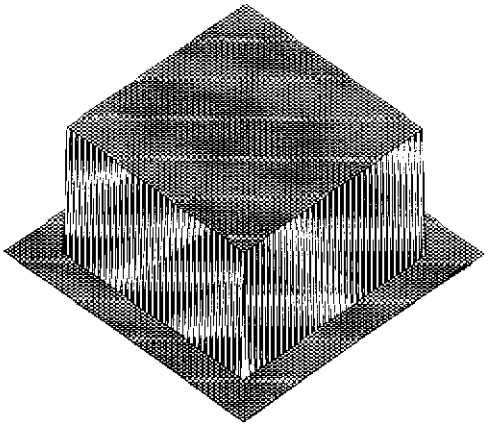
We implemented our algorithms under the fixed window scheme, in which the window size $\sigma = 7.0$. The minimal radial frequency was $\pi/10$, and the maximal radial frequency was about $\pi/3$. There were, in total, 48 filters. For Kanade and Okutomi's algorithm, the window size varies from 3 to 21 pixels, and the method iterates 5 times. For simplicity, we will denote: Kanade and Okutomi's algorithm as KOA; the moment filter approach without slope correction as MFS1; and the moment



SA

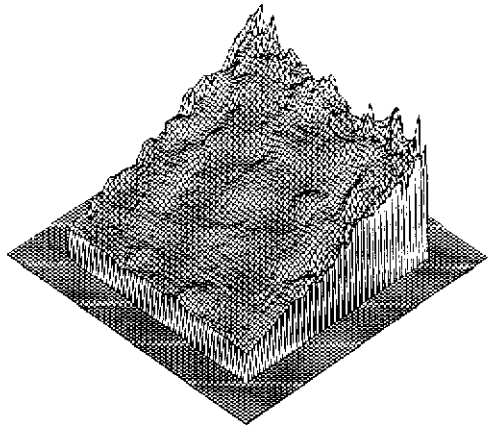


MFF1

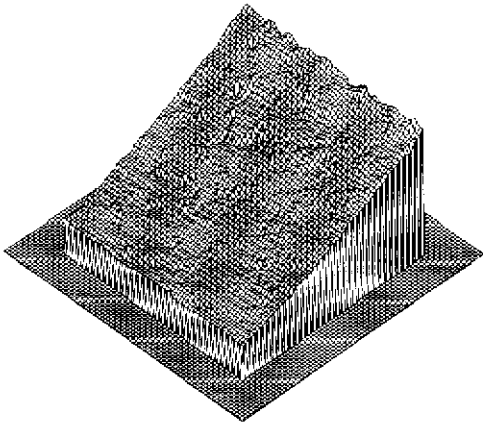


MFF2

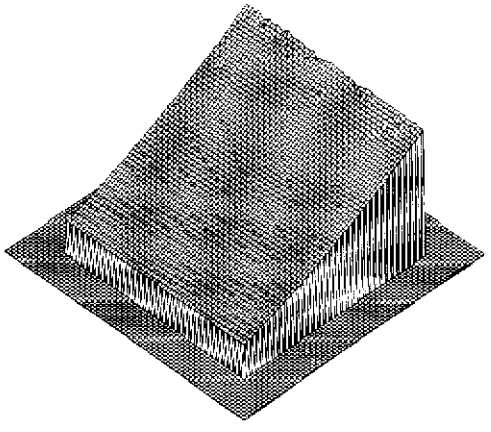
Figure 3.9: Computed Blurring Difference of the First Pair



SA



MFF1



MFF2

Figure 3.10: Computed Blurring Difference of the Second Pair

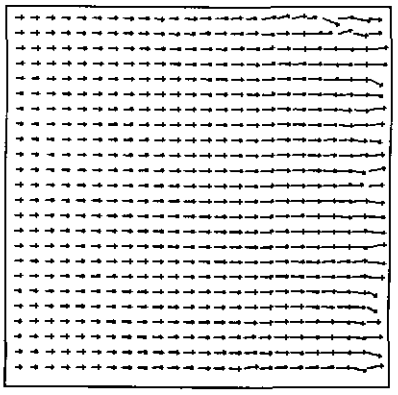


Figure 3.11: Computed Surface Slope of the Second Pair

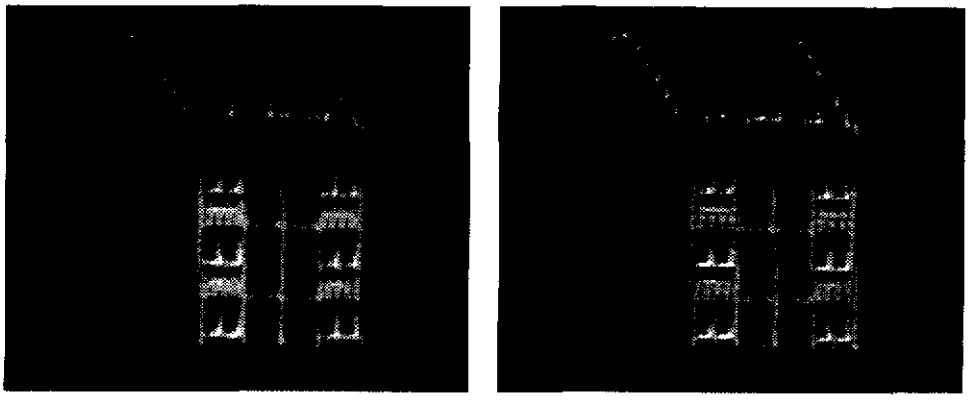
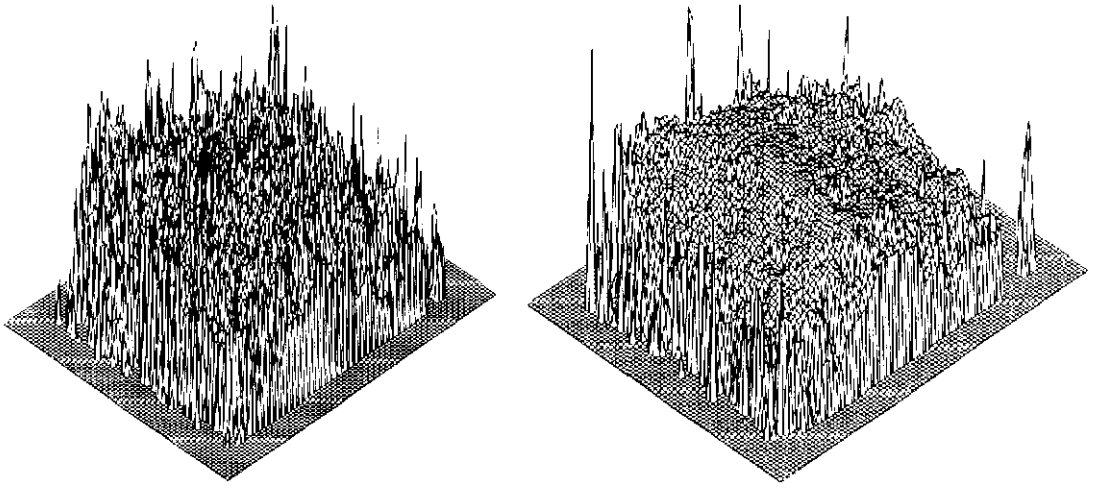
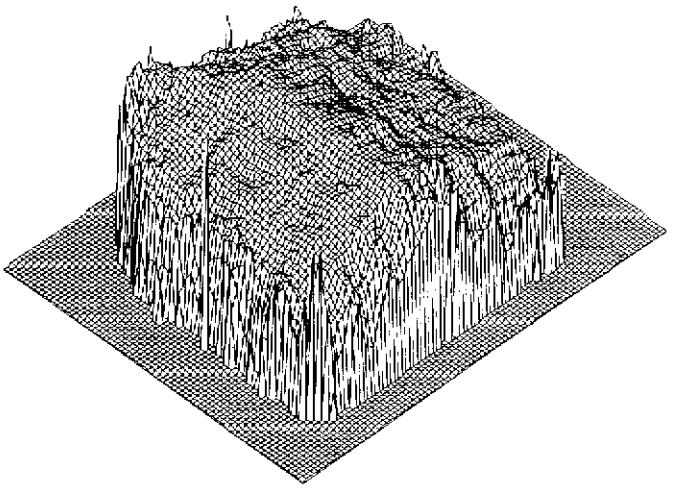


Figure 3.12: Real Image Pair of a Toy House

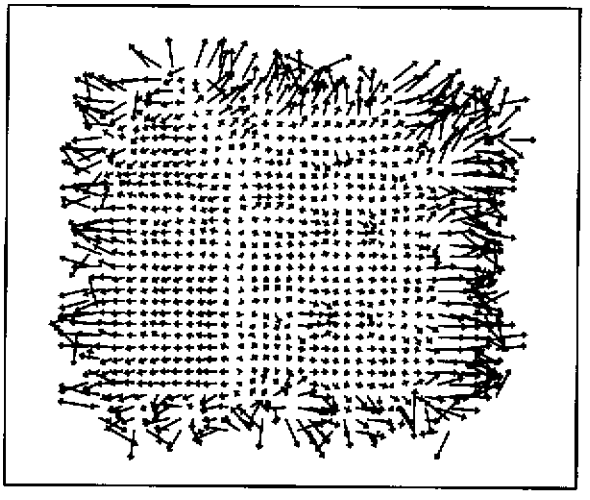


SA MFF1

Figure 3.13: Blurring Difference of House from SA and MFF1

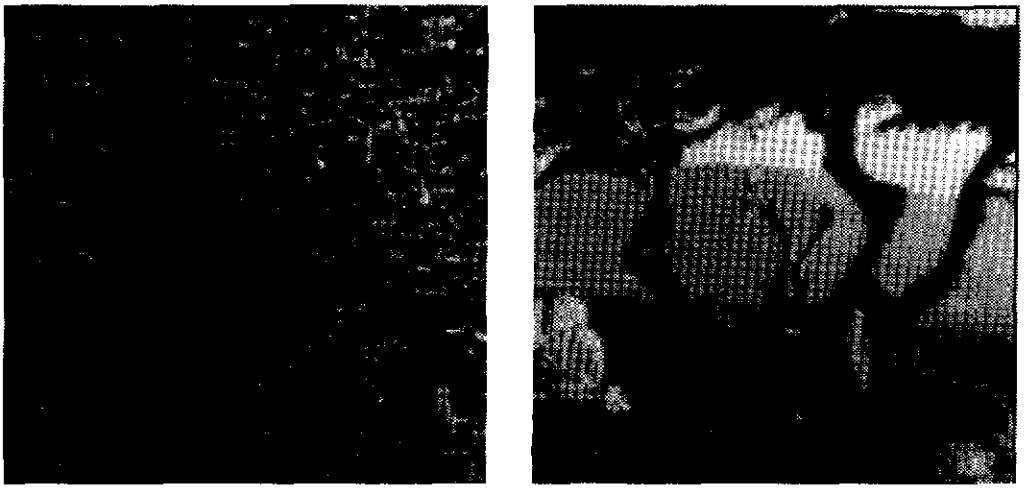


Blurring Difference



Slope

Figure 3.14: Blurring Difference and Slope of House from MFF2



(a) Texture Image (b) The 20th Frame of Tree Sequence

Figure 3.15: Images for Synthetic Experiments

filter approach with slope correction as MFS2. The RMS errors we will use below are in pixel widths.

We first tested the stereo algorithms on synthetic images. The first two test pairs were obtained by artificially shifting an original image. The first image was shifted by a uniform subpixel shift amount from the original, and the second image had a nonuniform subpixel shift, which, of course, resulted in stretching or compression. Technically, the subpixel shift was achieved by linearly interpolating the original image. Figure 3.15(a) shows the original texture image. The uniform shift was 1.55 pixels, and the nonuniform shift increased linearly from 1.0 pixel at the zero column to 3.54 pixels at the rightmost column. The third synthetic image pair included the 20th and 21st frames of the translating tree sequence [11] as in Figure 3.15(b). The scene was a planar surface textured by a picture of a tree. Because the ground truth of the motion was in the horizontal direction, we could use them as a stereo pair. The true disparity was between 1.73 and 2.26, and it increased linearly from left to right.

Table 3.2 shows the RMS errors of disparity values computed by these different algorithms. We see that for the first uniformly shifted image pair, all algorithms did very well. Because the RMS errors in this case were less than one percent of the pixel size, we believe that we are approaching the limit set by the spatial sampling. For the nonuniformly shifted pair, we see that the errors were significantly larger than those from the uniformly shifted pair. Figure 3.16 shows the disparity values of an arbitrary row for the nonuniformly shifted image pair. We see that KOA generated

	Uniformly Shifted Pair	Nonuniformly Shifted Pair	Tree Pair
KOA	0.008	0.035	0.016
MFS1	0.008	0.024	0.012
MFS2	0.006	0.020	0.010

Table 3.2: RMS Errors of Disparity from Synthetic Images

significant step-like ripples in this example. Figure 3.17 illustrates the slope computed by MFS2. For the tree pair, Figure 3.18 and Figure 3.19 show the disparity maps and slope computed by different algorithms.

We also tested these algorithms on two real image pairs which were taken by the Photometrics camera in our laboratory. The targets were about 2.5 meters from the camera, and the camera was translated horizontally by 0.5 inch between the left and right views. Because there was no vergence, the disparity map of a planar surface should still be planar. We will fit a plane to regions in the disparity map which correspond to planar surfaces, and then compared quantitatively the performance of stereo algorithms. Note that when we fitted the planar patches, we excluded those pixels near the boundaries so that the residual RMS errors computed were not influenced by the results near depth discontinuities.

The pair of images were of a textured cube shown in Figure 3.20. There were three large planar surfaces, which we denote as top patch, left patch, and right patch in Figure 3.20. The difference between the largest and the smallest disparity was about 3.0 pixel widths. Figure 3.21 and Figure 3.22 show the computed disparity maps and slope. We can see that the disparity map computed from KOA contained more ripples than those from MFS1 and MFS2. The slope estimation correctly indicated not only the direction of the slope, but also its magnitude. Table 3.3 shows the RMS error after plane fittings. We observe that: both MFS1 and MFS had better performance than KOA; and the RMS error of MFS2 remained almost constant when the surface slant increased, a fact we attribute to the slope correction performed by MFS2.

The second real pair has images of the same toy house we used in the focus experiments. Figure 3.23 shows the image pair we used for the stereo computation. There were two slightly slanted planar surfaces, which we show as the left patch and the right patch in Figure 3.23. The difference between the largest and the smallest disparity was only about 0.5 pixels, excluding the background. Figure 3.24 shows the computed disparity maps from KOA and MFS1. Again, we can see the consistent

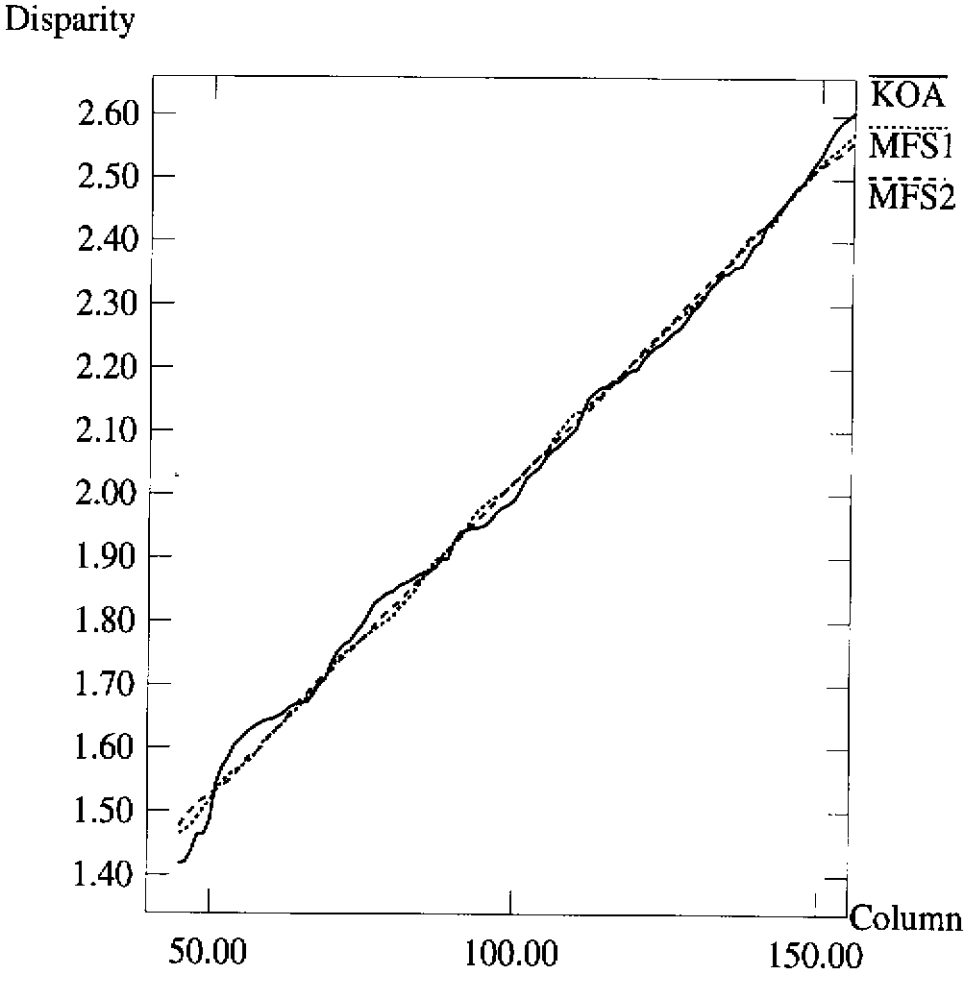


Figure 3.16: Disparity from the Nonuniformly Shifted Pair

	Top Patch (6690 Pixels)	Left Patch (16025 Pixels)	Right Patch (19625 Pixels)
KOA	0.063	0.032	0.027
MFS1	0.039	0.024	0.020
MFS2	0.018	0.020	0.016

Table 3.3: RMS Errors of the Cube

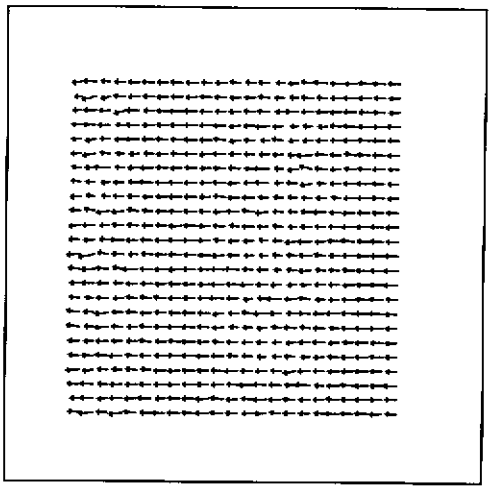
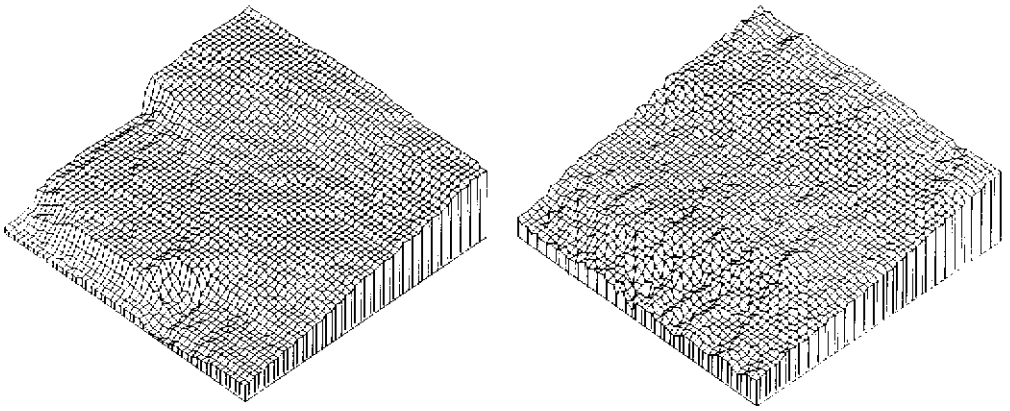
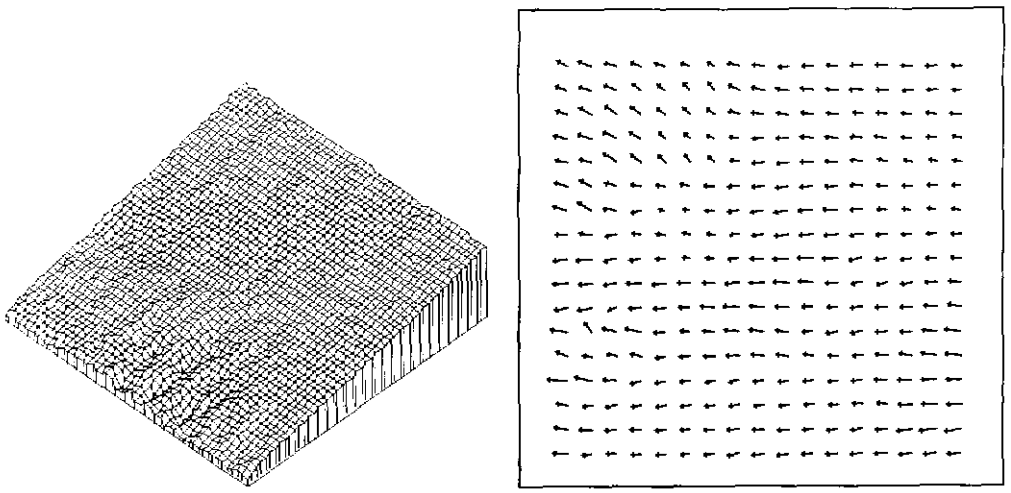


Figure 3.17: Slope of the Texture Pair Computed by MFS2



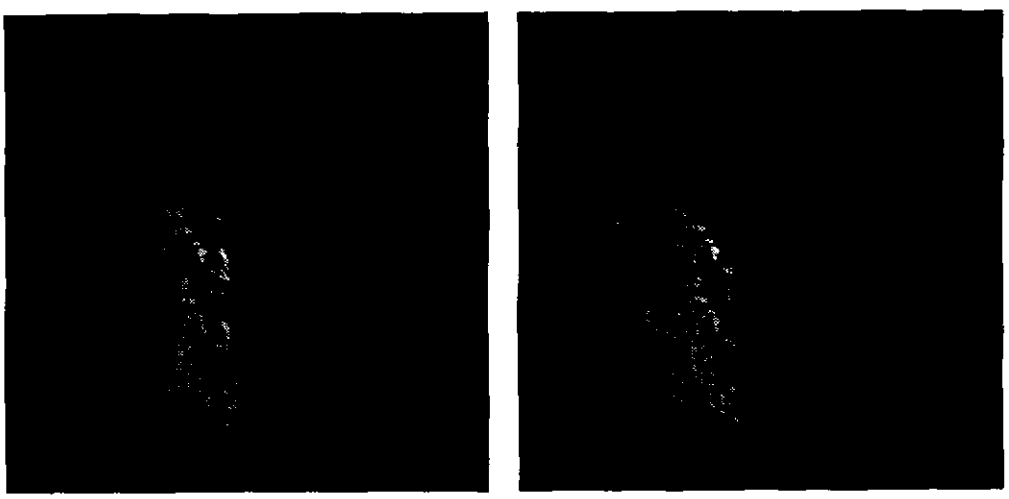
KOA MFS1

Figure 3.18: Disparity Maps of the Tree Pair from KOA and MFS1



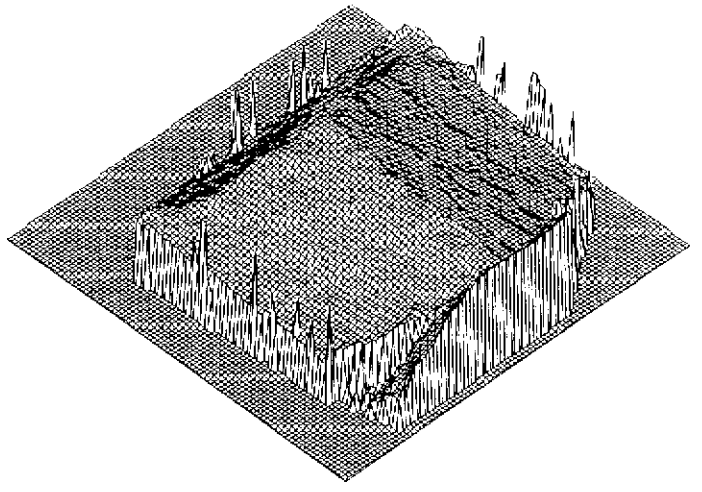
Disparity Surface Slope

Figure 3.19: Disparity and Slope of the Tree Pair from MFS2

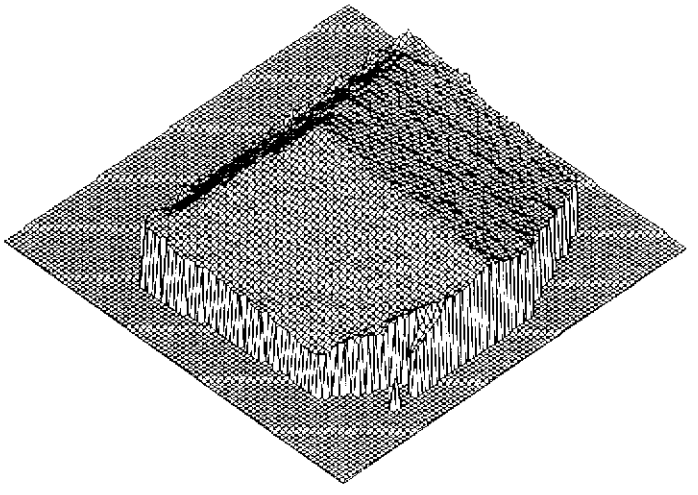


Left Image Planar Surfaces

Figure 3.20: Stereo Images of a Textured Cube

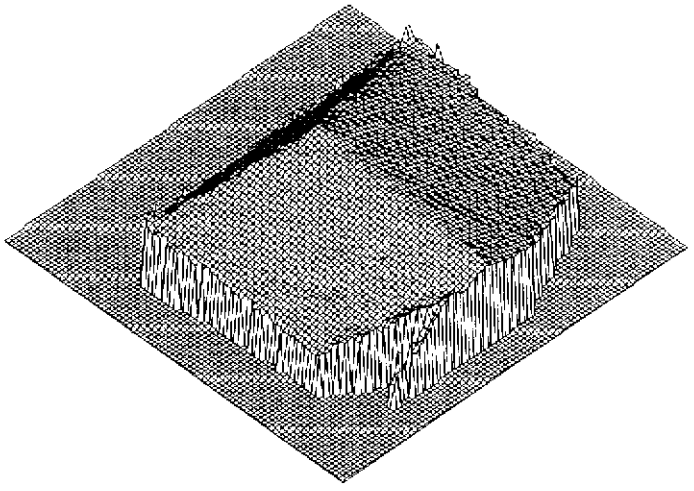


KOA

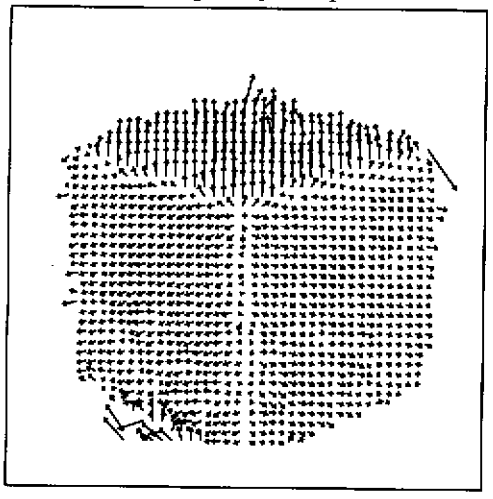


MFS1

Figure 3.21: Disparity Map of the Cube from KOA and MFS1



Disparity Map



Surface Slope

Figure 3.22: Disparity and Slope of the Cube from MFS2

	Left Patch (11260 Pixels)	Right Patch (13204 Pixels)
KOA	0.032	0.031
MFS1	0.028	0.017
MFS2	0.019	0.017

Table 3.4: RMS Errors of the Toy House

Left Image Planar Surfaces

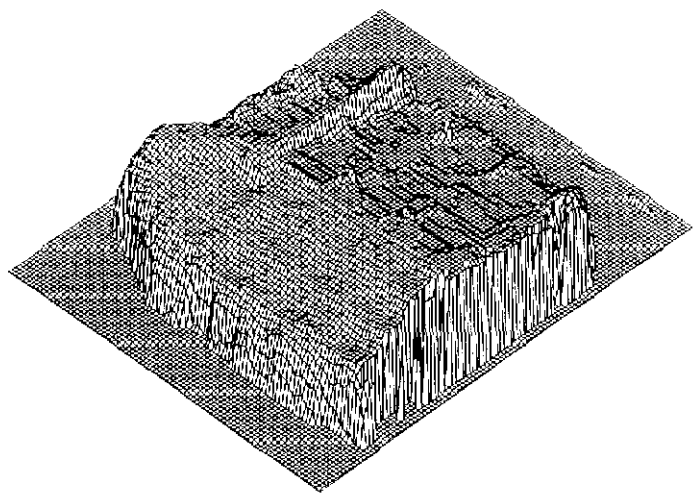
Figure 3.23: Stereo Images of a Toy House

step-like ripples from KOA. Figure 3.25 illustrates the disparity and slope computed from MFS2, which indicated magnitudes and directions of the slope of the surfaces, and located the discontinuities by large slope values. Table 3.4 shows the RMS error after plane fittings of the two patches.

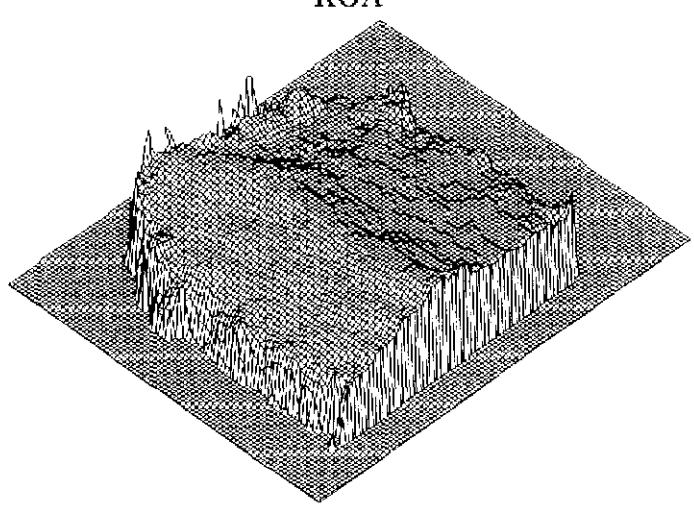
3.9 Summary

We have shown how the moment filters can be used to solve the problems of focus and stereo. In fact, our moment filter approach can handle a broad category of problems, which can be characterized as finding a parameter in the transformation between two images. As long as the transformation is well-modelled, the parameter and its gradient can be accurately estimated by the following generic algorithm:

1. In the neighborhood of the peak frequency f_0, the transformation in the Fourier domain can always be approximated as a polynomial of $(f - f_0)$ using Taylor expansion.



KOA



MFS1

Figure 3.24: Disparity Map of the House from KOA and MFS1

2. The recursiveness of the moment filters in the Fourier domain allows the convolution of the Gabor filter with one image to be expressed as the sum of convolutions of moment filters with the other image. Therefore, we obtain an equation of the parameter; and the equation can usually be solved by iteration.

3. The recursiveness of the moment filters in the spatial domain allows the modification of the filter outputs caused by linear shift variance of the parameter within the window to be approximated as a linear combination of convolutions of moment filters. This modification can be used to compute the gradient of the parameter and to correct the parameter estimation error caused by the gradient.

There is one common problem with both the traditional Gabor filter and the moment filter approaches, i.e., the frequency sampling problem. Even though we sampled the frequency domain in a way such that adjacent filters do not overlap much, we have no way of knowing whether they can represent image content either *completely* or *nonredundantly*. If we sample the frequency domain too densely, the filter outputs may be redundant. In this case, combining results or information from different frequency bands is unjustified. On the other hand, if we sample the frequency domain two sparsely, we may lose information in images.

To overcome this problem, we propose another set of filters – "hypergeometric" filters – which have recursive properties similar to those of the moment filters, and do not suffer from the above problem.

Chapter 4

Hypergeometric Filter Approach

4.1 Introduction

Hypergeometric filters have the recursive properties similar to those of the moment filters, and two additional properties:

- All hypergeometric filters have a single peak frequency. Because high order moment filters have two peak frequencies, we have to base our analysis on zero-order moment filters that have one single peak frequency. Hypergeometric filters allow us to use all high order ones.
- The set of hypergeometric filters samples the frequency domain completely and non-redundantly.

In moment filter or Gabor filter approaches, the sampling of the frequency domain is, in some sense, arbitrary as in Figure 3.7. In order to ensure that we use non-redundant and complete information contained in images, we need a set of filters which are orthogonal, complete and single-peaked. The hypergeometric filters proposed in this chapter satisfy all these requirements.

In the following sections, we will explain in detail how to apply the hypergeometric filter approach to two specific matching problems, i.e., optical flow and affine matching, though this approach is applicable to the general matching problem. There have been numerous computational models and algorithms proposed in literature to solve these two matching problems; unfortunately, almost all proposed matching algorithms share some basic limitations.

- **Finite Window Problem.** The computational models of translational matching are usually based on the assumption that the underlying signals have infinitely long duration, or the effects caused by finite window size are negligible. Such models include traditional SSD-based subpixel registration ([6]), differential techniques ([52]), energy-based methods([2]), and phase-based matching ([26]). In pursuit of high precision, the effects of finite window size have to be explicitly examined and, if possible, eliminated.

- **Affine Matching Problem.** When the two translational parameters are not constant within the size of the window, the translational model has to be replaced by the affine model. Many algorithms such as [54, 75, 39] use exhaustive searches to find those affine parameters. Generally speaking, the search is six-dimensional and, therefore, prohibitively expensive. To reduce the amount of computation, most people assume the two translational parameters to be known before the computation of the four deformation parameter is performed. But in many applications, the translational parameters can never be accurately estimated without knowing the four deformation parameters. Therefore, we need a new way to *compute*, instead of search for, the six affine parameters simultaneously.

- **General Matching Problem.** We also encounter some other matching problem, such as in depth from defocus [99, 82]. These problems bear obvious similarity in that they all try to match two images, yet these sophisticated algorithms developed for the translational matching can not be effectively extended to other matching problems.

Since the new hypergeometric filters are recursive in both the spatial and frequency domains, we will demonstrate that:

- The effects of the finite window size can be expressed as high order terms in a Taylor expansion. Ignoring these window effects is equivalent to truncating the Taylor expansion after the first term. Therefore, by truncating the expansion at a higher order, we are able to reduce, and numerically eliminate, these effects. By doing so, we can achieve much higher precision in optical flow computation.

- The effects of the affine deformation can be *analytically* expressed as a linear modification of the original image. Because these effects are analytical, we can directly *compute* the six parameters of the affine transform without resorting to exhaustive search or image warping.

- The approach can easily be applied to a wide variety of matching problems, in which the transformation between two image patches can be analytically expressed in the frequency domain.

We will also present experiments that illustrate these properties and their applications. To demonstrate the superiority of our approach, the results on optical flow are quantitatively compared to those reported in literature. The experiments on affine matching show that we can reliably extract the six affine parameters even from a very small image patch. We also implemented the hypergeometric filter approach on focus matching and stereo matching problems to demonstrate that approach's general applicability. The results on focus and stereo are quantitatively compared to other approaches.

4.2 Hypergeometric Filters

4.2.1 Definitions

We define the "hypergeometric filters" or "H filters" in the Fourier domain as ($m = 1, 2, ..., \infty$):

$$H_m(f) = \begin{cases} c_m f^m e^{-f^2\sigma^2/2} & \text{when } f \geq 0 \\ 0 & \text{when } f < 0 \end{cases} \tag{4.1}$$

$$H_0(f) = (2\sqrt{\pi}\sigma)^{1/2} e^{-f^2\sigma^2/2} = c_0 e^{-f^2\sigma^2/2} \tag{4.2}$$

$$H_{-m}(f) = \begin{cases} 0 & \text{when } f > 0 \\ c_m(-f)^m e^{-f^2\sigma^2/2} & \text{when } f \leq 0 \end{cases}, \tag{4.3}$$

where c_m is the real normalization constant:

$$c_m = 2\sigma^m \sqrt{\frac{\pi\sigma}{\Gamma(m+1/2)}}. \tag{4.4}$$

where σ is the window size constant, $\Gamma()$ is the Gamma function.

Actually $H_m(f)$ represents the positive frequency peak of the order m moment filter in Figure 3.1 with $f_0 = 0$, and $H_{-m}(f)$ represents the negative frequency peak. Therefore, we can easily verify that $H_m(f)$ is an asymmetric bell-shaped band pass filter with the peak frequency at:

$$\text{Peak}[H_m(f)] = \frac{\sqrt{m}}{\sigma}, \tag{4.5}$$

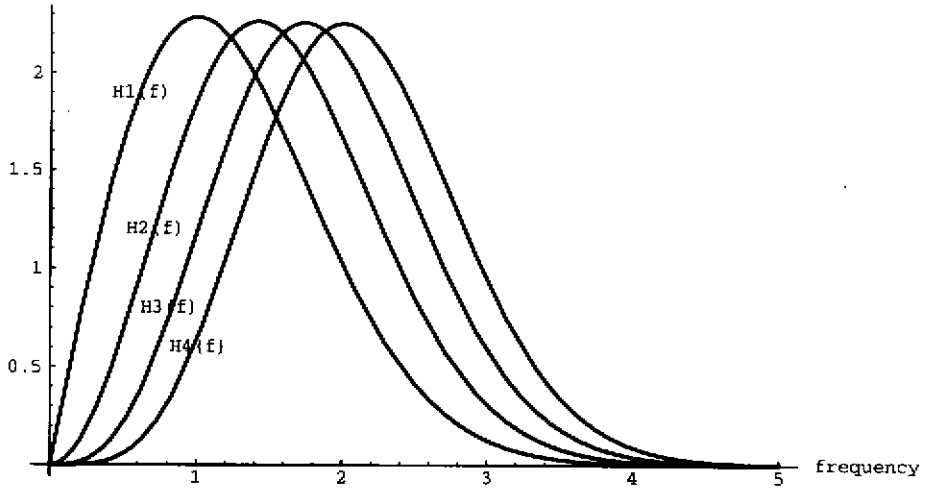


Figure 4.1: The Passbands of Hypergeometric Filters

$$\text{Peak}[H_0(f)] = 0, \tag{4.6}$$

$$\text{Peak}[H_{-m}(f)] = -\frac{\sqrt{m}}{\sigma}. \tag{4.7}$$

Figure 4.1 shows the passbands of $H_1(f), H_2(f), H_3(f)$ and $H_4(f)$.

Taking inverse Fourier transform of the H filters in Eq. 4.1 to 4.3, we have the H filters represented in the spatial domain:

$$\begin{aligned} h_m(x) &= \sqrt{\frac{2^m}{\pi\Gamma(m+1/2)\sigma}} e^{-\frac{x^2}{2\sigma^2}} \\ &\quad \left(\frac{1}{\sqrt{2}}\Gamma(\frac{1+m}{2})\Phi(-\frac{m}{2}, \frac{1}{2}, \frac{x^2}{2\sigma^2}) + j\Gamma(1+\frac{m}{2})\Phi(\frac{1-m}{2}, \frac{3}{2}, \frac{x^2}{2\sigma^2})\frac{x}{\sigma} \right) \\ &= \left(a_m \Phi(-\frac{m}{2}, \frac{1}{2}, \frac{x^2}{2\sigma^2}) + j b_m x \Phi(\frac{1-m}{2}, \frac{3}{2}, \frac{x^2}{2\sigma^2}) \right) e^{-\frac{x^2}{2\sigma^2}}, \end{aligned} \tag{4.8}$$

$$h_0(x) = (\sqrt{\pi}\sigma)^{-1/2} e^{-\frac{x^2}{2\sigma^2}}, \tag{4.9}$$

$$h_{-m}(x) = h_m^*(x), \tag{4.10}$$

where j is the complex constant $\sqrt{-1}$, $\Phi(a, c, x)$ is a special type of hypergeometric function, known as "confluent hypergeometric function" or "Kummer's function" in literature ([1]), and a_m and b_m are constants decided by m. Eq. 4.8 has already been normalized so that the energy of the filter $h_m(x)$ is one. The solid curves in Figure 4.2 are H filters of the second, fifth and tenth orders, while the dashed curves are the closest Gabor filters in the sense that they have the same peak frequency and bandwidth. Only real parts of the complex filters are shown.

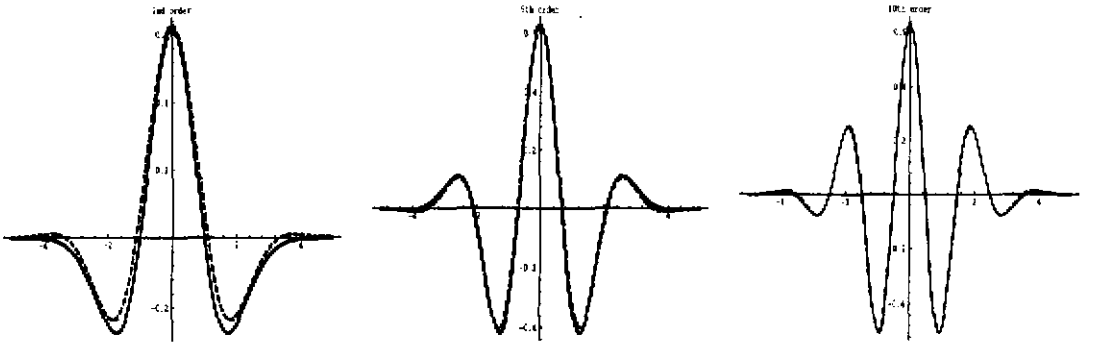


Figure 4.2: H Filters and Their Closest Gabor Filters

As shown in Appendix E, the relation between the H filters and the moment filters is

$$h_{2i}(x) + h_{-2i}(x) = k_1 L(i, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}}, \quad (4.11)$$

$$h_{2i+1}(x) - h_{-(2i+1)}(x) = jk_2 x L(i, \frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}}, \quad (4.12)$$

where k_1 and k_2 are real constants decided by i, and $L(n, \alpha, x)$ is the generalized Laguerre polynomial. Comparing Eq. 4.11 and Eq. 4.12 with Eq. 3.2, referring to the fact that moment filters are orthogonal and complete, we conclude that we can reconstruct the original signal without loss from coefficients resulted from convolving the image with H filters.

Two-dimensional extension of the H filter is straightforward:

$$h_{mn}(x, y) = h_m(x) h_n(y), \quad (4.13)$$

$$H_{mn}(f_x, f_y) = H_m(f_x) H_n(f_y). \quad (4.14)$$

Figure 4.3 shows a perspective plot of $h_{10}(x)h_{20}(y)$ when $\sigma = 7.0$.

Without loss of generality, we will demonstrate our approach using only H filters of positive peak frequency. Those of negative or zero frequency can be similarly used in the following computations.

4.2.2 Recursiveness of Hypergeometric Filters

As proven in Appendix E, the hypergeometric filters have the following properties:

- The hypergeometric filters are recursive in the Fourier domain, i.e.,

$$H_{m+1}(f) = \frac{c_{m+1}}{c_m} f H_m(f). \quad (4.15)$$

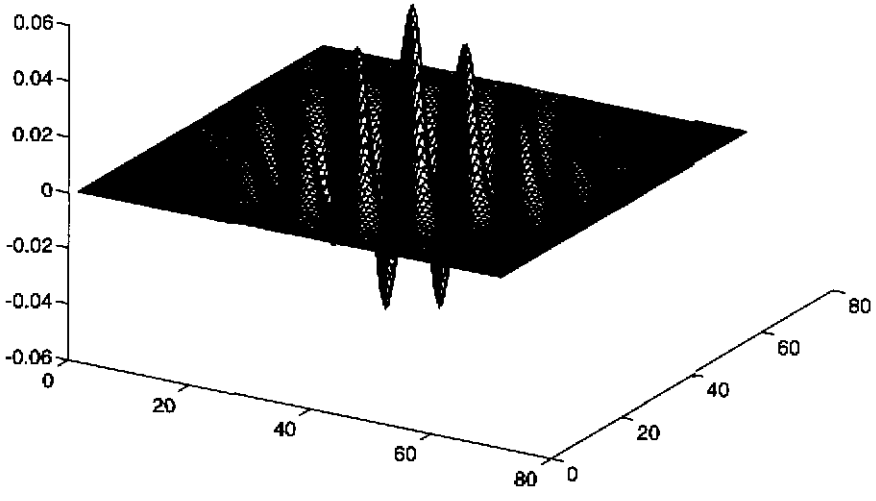


Figure 4.3: A Two-Dimensional Hypergeometric Filter

- The hypergeometric filters are recursive in the spatial domain, i.e.,

$$xh_m(x) = k_{+1}h_{m+1}(x) + k_{-1}h_{m-1}(x), \tag{4.16}$$

where

$$k_{+1} = -j\sigma\sqrt{m+1/2}, \qquad k_{-1} = j\sigma\sqrt{m^2/(m-1/2)}. \tag{4.17}$$

- The hypergeometric filters are recursive with respect to the differential operation, i.e.,

$$h'_m(x) = j\frac{c_m}{c_{m+1}}h_{m+1}(x). \tag{4.18}$$

4.2.3 Other Properties of Hypergeometric Filters

Unlike Gabor filters which have positive value at zero frequency, the DC components of H filters are zero except for $m = 0$. Therefore, the H filters have the advantage of avoiding the strong DC bias existing in Gabor filters, a bias that is extremely harmful when the peak frequency is low.

Now let us look at the effective bandwidth in the Fourier domain and effective spatial extent. According to Gabor [28], we have

$$\Delta x_m^2 = \frac{\int_{-\infty}^{+\infty} x^2 \parallel h_m(x) \parallel^2 dx}{\int_{-\infty}^{+\infty} \parallel h_m(x) \parallel^2 dx}, \tag{4.19}$$

$$\Delta f_m^2 = \frac{\int_{-\infty}^{+\infty} (f - \text{Peak}[H_m(f)])^2 \parallel H_m(f) \parallel^2 df}{\int_{-\infty}^{+\infty} \parallel H_m(f) \parallel^2 df}. \tag{4.20}$$

Figure 4.4 shows the effective bandwidth and effective spatial extent. As indicated by the well-known uncertainty principle in local frequency analysis, the effective bandwidth and effective spatial extent must satisfy the following uncertainty relation:

$$\varepsilon_m = \sqrt{\Delta x_m^2 \Delta f_m^2} \geq \frac{1}{2}. \tag{4.21}$$

And Figure 4.5 shows that, as Gabor filters have the minimal uncertainty, the uncertainties of the H filters ε_m is only slightly larger ($\varepsilon_m < 0.51$, when $m > 5$).

Since Gabor filters are the only filters with minimal uncertainty, and the uncertainty of H filters approaches the minimal value as m increases, we can predict that the H filters become more and more similar to Gabor filters as m increases. The prediction is verified by Figure 4.2, in which the higher order the hypergeometric filter is, the more similar it is to the Gabor filters.

The H filters sample the frequency space automatically as illustrated in Eq. 4.5 and Eq. 4.7. Unlike the frequency sampling in STFT (short time Fourier transform), which is uniform, the H filters sample the frequency space in a non-uniform fashion. Figure 4.6 shows the samplings of the frequency space in STFT and H filters, in case of $\sigma = 2.0$ and window size is 21. And the highest frequency is Nyquist frequency π.

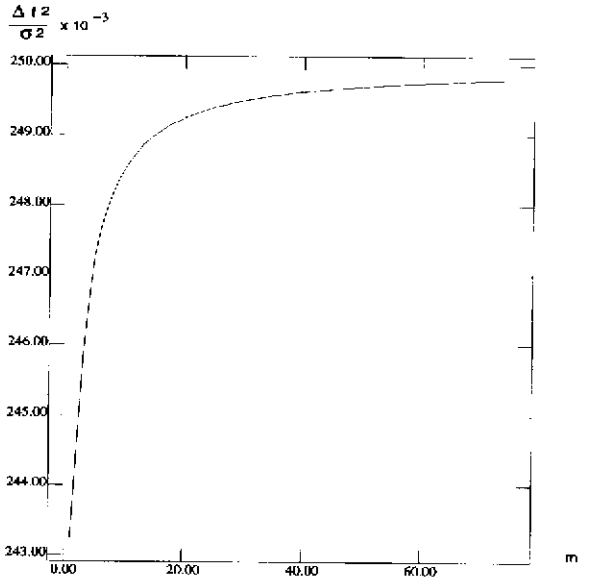
4.3 Translational Matching

In this section, we consider the traditional problem of recovering two motion parameters at every pixel, assuming translational motion i.e.,, in the spatial domain

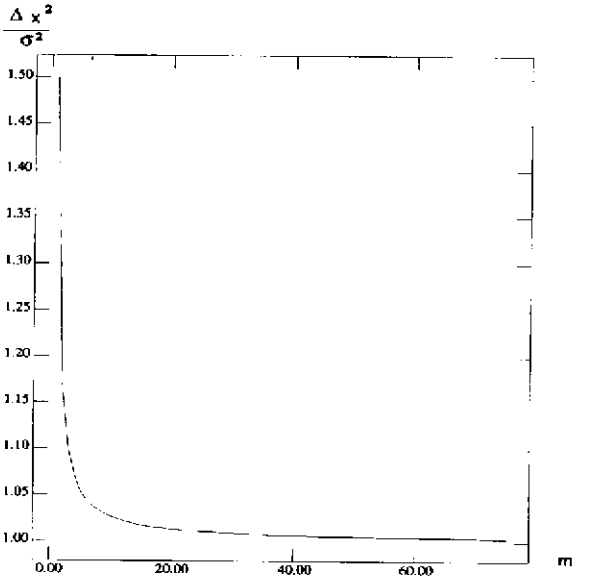
$$i_1(x, y) = i_0(x + u, y + v), \tag{4.22}$$

or in the frequency domain

$$I_1(f_x, f_y) = I_0(f_x, f_y) e^{j(f_x u + f_y v)}, \tag{4.23}$$



Effective Bandwidth



Effective Spatial Extent

Figure 4.4: Uncertainty in the Spatial and Frequency Domains

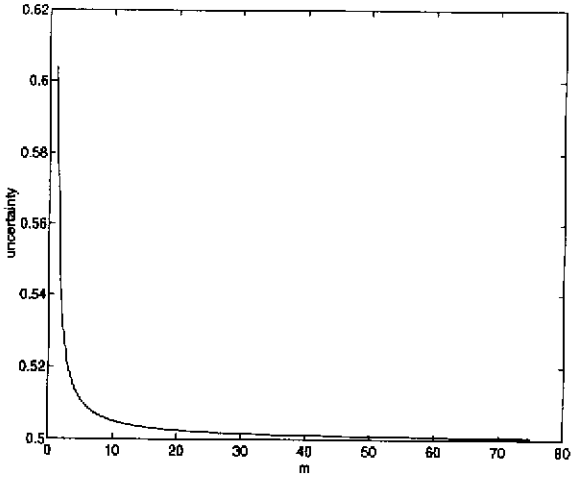


Figure 4.5: Product of the Spatial and Frequency Uncertainty

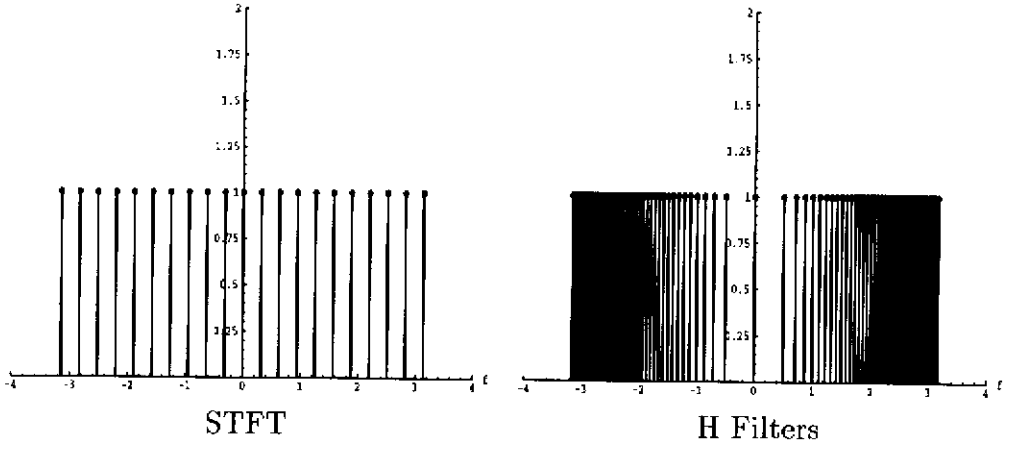


Figure 4.6: Frequency Sampling in STFT and H Filters

where $i_0(x,y)$ and $i_1(x,y)$ are the two images in the spatial domain, $I_0(f_x,f_y)$ and $I_1(f_x,f_y)$ are in the frequency domain, and (u,v) is the 2D translation.

Theoretically, if u and v are constant in the whole image, and the size of the image is infinitely large, we can precisely compute them by using correlation or differentiation techniques based on Eq. 4.22, or phase-based or energy-based techniques based on Eq. 4.23. Unfortunately none of these assumptions are true in reality. In this section, we show how to precisely compute the translation by only assuming u and v are constants within a *finite* size window. In the affine matching section, we will further relax this assumption.

Let U_{mn} and V_{mn} be results from convolving the images with $h_{mn}(x,y)$, i.e.,

$$\begin{aligned} U_{mn} &= i_0(x,y) \otimes h_{mn}(x,y) \\ &= I_0(f_x,f_y) H_{mn}(f_x,f_y), \qquad (4.24) \\ V_{mn} &= i_1(x,y) \otimes h_{mn}(x,y) \\ &= I_1(f_x,f_y) H_{mn}(f_x,f_y) \\ &= I_0(f_x,f_y) e^{j(f_x u + f_y v)} H_{mn}(f_x,f_y). \qquad (4.25) \end{aligned}$$

In the following, we will base our computation solely on U_{mn} and V_{mn}. Since $h_{mn}(x,y)$ has finite effective width, the values of U_{mn} and V_{mn} are unaffected by values of u and v outside the effective width of the H filter.

Suppose (f_{0x}, f_{0y}) is the peak frequency of $H_{mn}(f_x,f_y)$; we obtain the following Taylor expansion around the peak frequency:

$$\begin{aligned} e^{j(f_x u + f_y v)} &= \sum_{p=0}^{\infty}\sum_{q=0}^{\infty} A_1(p,q,u,v)(f_x - f_{0x})^p (f_y - f_{0y})^q \\ &= \sum_{p=0}^{\infty}\sum_{q=0}^{\infty} A_{pq}(f_{0x}, f_{0y}, u, v) f_x^p f_y^q, \qquad (4.26) \end{aligned}$$

where $A_1(p,q,u,v)$ and $A_{pq}(f_{0x}, f_{0y}, u, v)$ are from coefficients of the Taylor expansion. Using the recursiveness of the H filter in the frequency domain, we then obtain

$$\begin{aligned} & e^{j(f_x u + f_y v)} H_{mn}(f_x,f_y) \\ = & \sum_{p=0}^{\infty}\sum_{q=0}^{\infty} A_{pq}(f_{0x}, f_{0y}, u, v) f_x^p H_m(f_x) f_y^q H_n(f_y) \\ = & \sum_{p=0}^{\infty}\sum_{q=0}^{\infty} A_{pq}(f_{0x}, f_{0y}, u, v) \frac{c_m c_n}{c_{m+p} c_{n+q}} H_{m+p}(f_x) H_{n+q}(f_y). \qquad (4.27) \end{aligned}$$

Applying Eq. 4.5, we can further simplify the above equation as

$$e^{j(f_x u+f_y v)} H_{mn}(f_x, f_y) = \sum_{p=0}^{\infty}\sum_{q=0}^{\infty} B_{pqmn}(u,v) H_{(m+p)(n+q)}(f_x, f_y), \tag{4.28}$$

where

$$B_{pqmn}(u,v) = A_{pq}(\frac{\sqrt{m}}{\sigma}, \frac{\sqrt{n}}{\sigma}, u, v)\frac{c_m c_n}{c_{m+p} c_{n+q}}.$$

We can then combine Eq. 4.28 and Eq. 4.25 into

$$\begin{aligned} V_{mn} &= I_0(f_x, f_y)\sum_{p=0}^{\infty}\sum_{q=0}^{\infty} B_{pqmn}(u,v) H_{(m+p)(n+q)}(f_x, f_y) \\ &= \sum_{p=0}^{\infty}\sum_{q=0}^{\infty} B_{pqmn}(u,v) I_0(f_x, f_y) H_{(m+p)(n+q)}(f_x, f_y) \\ &= \sum_{p=0}^{\infty}\sum_{q=0}^{\infty} B_{pqmn}(u,v) U_{(m+p)(n+q)}. \end{aligned} \tag{4.29}$$

If we truncate the Taylor expansion after order N, we can compute the translation parameters u and v by minimizing

$$S(u,v) = \sum_m \sum_n \parallel V_{mn} - \sum_{p=0}^{N}\sum_{q=0}^{N} B_{pqmn}(u,v) U_{(m+p)(n+q)} \parallel^2 . \tag{4.30}$$

In practice, it usually takes less than ten evaluations of $S(u,v)$ to find minima using the Levenberg-Marquardt method when $N = 3$. Furthermore, if we assume the uncertainty of V_{mn} is caused by image noise and window truncation, we can apply the standard χ^2 analysis to estimate the covariance of (u,v).

It is interesting to note that when the Taylor expansion in Eq. 4.26 is truncated after the first term, Eq. 4.29 becomes

$$V_{mn} = e^{j(f_{0x} u+f_{0y} v)} U_{mn}, \tag{4.31}$$

which is exactly what has been used in the phase-based approach, where window effects are ignored as in [92, 27]. Thus the traditional approach with Gabor filters is a degenerate case of the method using hypergeometric filters.

Comparing Eq. 4.29 and Eq. 4.31, we conclude that the higher order terms in Eq. 4.29 represent modifications introduced by finite window size. By adding these modification terms in the minimization in Eq. 4.30, we are able to reduce, or numerically eliminate window effects.

4.4 Affine Matching

In the previous section, we assumed that the two translation parameters (u, v) were constant within the window area. In this section, we relax this assumption to let (u, v) change linearly within the window, i.e., the affine matching problem:

$$\begin{pmatrix} u \\ v \end{pmatrix} \doteq \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} + \begin{pmatrix} u_x & u_y \\ v_x & v_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}. \tag{4.32}$$

We call this affine matching problem a *derivative* of the translational matching problem because we are estimating the translation parameters and their derivatives simultaneously, i.e.,

$$\nabla u = \begin{pmatrix} u_x \\ u_y \end{pmatrix}, \tag{4.33}$$

$$\nabla v = \begin{pmatrix} v_x \\ v_y \end{pmatrix}. \tag{4.34}$$

At first, let us consider the case when $u_0 = v_0 = 0$, i.e., the translation is zero at the center of the window. Then we have

$$\begin{aligned} i_1(x, y) &= i_0(x + u_x x + u_y y, y + v_x x + v_y y) \\ &\approx i_0(x, y) + (u_x x + u_y y)\frac{\partial i_0}{\partial x} + (v_x x + v_y y)\frac{\partial i_0}{\partial y}. \end{aligned} \tag{4.35}$$

Without loss of generality, we consider the convolution of H filters with the images at pixel position $(0, 0)$, i.e.,

$$\begin{aligned} V_{mn} &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} i_1(x, y) h_{mn}(-x, -y)dxdy \\ &\approx \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} i_0(x, y) h_{mn}(-x, -y)dxdy + \\ &\quad \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (u_x x + u_y y)\frac{\partial i_0}{\partial x} h_{mn}(-x, -y)dxdy + \\ &\quad \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} (v_x x + v_y y)\frac{\partial i_0}{\partial y} h_{mn}(-x, -y)dxdy. \end{aligned} \tag{4.36}$$

Using the partial integration

$$\int udv = uv - \int vdu, \tag{4.37}$$

the limit values of H filters

$$\lim_{x \to \pm\infty} h_m(x) = 0, \tag{4.38}$$

and the recursiveness of H filter in the spatial domain

$$x h_m(x) = k_{+1} h_{m+1}(x) + k_{-1} h_{m-1}(x), \tag{4.39}$$

we have Eq. 4.36 as

$$\begin{aligned} V_{mn} &= C_{00}U_{mn} + C_{(-1)1}U_{(m-1)(n+1)} + C_{1(-1)}U_{(m+1)(n-1)} + \\ &\quad C_{20}U_{(m+2)n} + C_{11}U_{(m+1)(n+1)} + C_{02}U_{m(n+2)} \\ &= C_{00}U_{mn} + \sum_{p,q \geq -1, |p|+|q|=2} C_{pq}U_{(m+p)(n+q)}, \end{aligned} \tag{4.40}$$

where

$$C_{00} = 1 + mu_x + nv_y, \qquad C_{(-1)1} = mv_x\sqrt{\frac{n+1/2}{m-1/2}},$$

$$C_{1(-1)} = nu_y\sqrt{\frac{m+1/2}{n-1/2}}, \qquad C_{20} = -u_x\sqrt{(m+3/2)(m+1/2)},$$

$$C_{11} = -(u_y + v_x)\sqrt{(n+1/2)(m+1/2)}, \qquad C_{02} = -v_y\sqrt{(n+3/2)(n+1/2)}.$$

Eq. 4.40 shows that the effects of affine deformation can be represented as a linear modification of the original image. Similar to what we did in the translational matching section, we can form an objective function $S_1(u_x, u_y, v_x, v_y)$, which, when minimized, yields the four deformation parameters.

$$S_1(u_x, u_y, v_x, v_y) = \sum_m \sum_n \| V_{mn} - C_{00}U_{mn} - \sum_{p,q \geq -1, |p|+|q|=2} C_{pq}U_{(m+p)(n+q)} \|^2 . \tag{4.41}$$

So far we have only considered the case when the translation is zero. When the translation is not zero, we can decompose the general affine transform into two steps. The first step is a pure deformation, and the second pure translation as illustrated in Figure 4.7, where

$$i(x, y) = i_0(x + u_x x + u_y y, y + v_x x + v_y y), \tag{4.42}$$

$$i_1(x, y) = i(x + u_1, y + v_1). \tag{4.43}$$

If we let

$$\begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = \begin{pmatrix} 1 + u_x & u_y \\ v_x & 1 + v_y \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \end{pmatrix}, \tag{4.44}$$

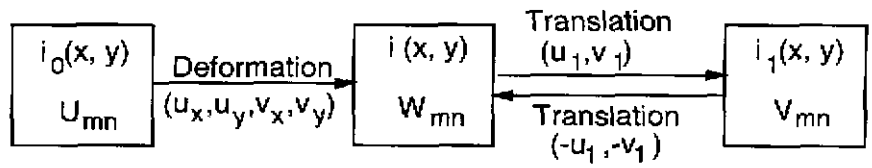


Figure 4.7: Two-Step Decomposition of an Affine Transform

we then have the original relation between $i_0(x)$ and $i_1(x)$, i.e.,

$$\begin{aligned} i_1(x,y) &= i(x+u_1, y+v_1) \\ &= i_0(x+u_0+u_x x+u_y y, y+v_0+v_x x+v_y y). \end{aligned} \tag{4.45}$$

In other words, the combination of the two transforms is equivalent to the original affine transform. Since the pure translation is also reversible as in Figure 4.7, we have

$$i(x,y) = i_1(x-u_1, y-v_1). \tag{4.46}$$

Let W_{mn} be the convolution of $i(x,y)$ with $h_{mn}(x,y)$, from Eq. 4.29 and Eq. 4.46, we have

$$W_{mn} = \sum_{p=0}^{\infty}\sum_{q=0}^{\infty} B_{pqmn}(-u_1,-v_1)V_{(m+p)(n+q)}. \tag{4.47}$$

From Eq. 4.40 and Eq. 4.42, we also have

$$W_{mn} = C_{00}U_{mn} + \sum_{p,q\geq -1, |p|+|q|=2} C_{pq}U_{(m+p)(n+q)}. \tag{4.48}$$

From these two equations, we can formulate the following objective function, which yields the six affine parameters when minimized:

$$\begin{aligned} S_2(u_1,v_1,u_x,u_y,v_x,v_y) = \sum_m \sum_n \| \; & C_{00}U_{mn} + \sum_{p,q\geq -1, |p|+|q|=2} C_{pq}U_{(m+p)(n+q)} - \\ & \sum_{p=0}^{N}\sum_{q=0}^{N} B_{pqmn}(-u_1,-v_1)V_{(m+p)(n+q)} \|^2 . \end{aligned} \tag{4.49}$$

4.5 General Matching Problem

As an abstract example, let us consider a general matching problem stated as recovering parameters $\vec{u} = (p_1, p_2, \cdots)$ in the transformation $T(f_x, f_y; \vec{u})$ between two images

$I_0(f_x, f_y)$ and $I_1(f_x, f_y)$. For example, in the above translational matching problem:

$$\vec{u} = (u, v), \tag{4.50}$$

$$T(f_x, f_y; \vec{u}) = e^{j(f_x u + f_y v)}. \tag{4.51}$$

As long as the transformation $T(f_x, f_y; \vec{u})$ can be analytically expressed, we can always expand it as a polynomial of f_x and f_y, while the coefficients of the polynomial are functions of $\vec{u}$. Using the recursiveness of H filters in the frequency domain, we always have

$$V_{mn} = \sum_{p=0}^{\infty} \sum_{q=0}^{\infty} B_{pqmn}(\vec{u}) U_{(m+p)(n+q)}, \tag{4.52}$$

where the analytical form of $B_{pqmn}(\vec{u})$ can be easily derived from the Taylor expansion of $T(f_x, f_y; \vec{u})$. A similar objective function to Eq. 4.30 can be used to compute the parameters $\vec{u}$.

When the series in Eq. 4.52 is truncated after the zeroth order term, we have

$$V_{mn} = T(f_{0x}, f_{0y}; \vec{u}) U_{mn}, \tag{4.53}$$

which is the case when window effects are ignored.

The derivative problem[1] is that we need to estimate $\vec{u}$ and its spatial derivatives $(\vec{u}_x, \vec{u}_y)$ as well. As demonstrated in the affine matching section, we can usually decompose it into two steps, where the first one represents the effects caused by non-zero spatial derivatives $(\vec{u}_x, \vec{u}_y)$ while $\vec{u} = 0$, and the second represents the effects caused by $\vec{u}$ while its derivatives are zero. Due to the recursiveness of the H filter, all these effects can be analytically expressed as linear modifications. Finally, we can formulate an objective function to compute $\vec{u}$ and $(\vec{u}_x, \vec{u}_y)$ simultaneously.

4.6 Comparisons of Moment Filters and Hypergeometric Filters

Because both moment filters and hypergeometric filters have similar recursiveness properties in the spatial and Fourier domains, they yield the same strategy for solving parameter extraction problems. This strategy uses a polynomial to fit the underlying

[1] In this sense, the derivative problem is a more general definition than the foreshortening problem in the previous chapter.

transformation curve[2], and therefore obtain results of much higher precision than the traditional approaches which use a constant to approximate the curve. In this respect the two approaches are equally effective in computing accurately and handling the foreshortening or derivative problem.

The moment filter approach is based on a single frequency band. Thus we have the freedom to sample the frequency space in any way we like. The disadvantage of this freedom is that there is no way that we can sample the frequency domain perfectly. A dense sampling scheme undermines the assumption of the Kalman filtering in combining results, while a sparse sampling scheme abandons information contained in images. On the other hand, the advantage of this freedom is that we can have more sophisticated window schemes such as the wavelet scheme. One additional disadvantage of the moment filter approach is that the expansion using moment filters has a higher dimension than is necessary although we usually did not use moment filters with order higher than five.

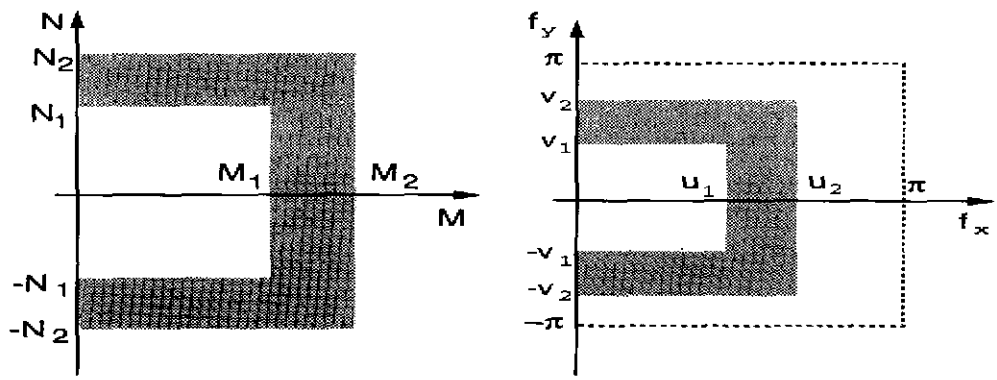
The advantages and disadvantages of the hypergeometric filter approach are exactly the opposite of those of the moment filter approach. The hypergeometric filter approach samples the frequency space in a canonical fashion such that all information is utilized. But on the other hand, we do not have the freedom to choose different window sizes for different frequency bands.

4.7 Implementation Issues and Experiments

4.7.1 Computation of H Filters

Unlike for most other filters, the numerical computation of a hypergeometric filter itself is a difficult problem. The infinite series representation of the confluent hypergeometric function in [81] (Equation 47:3:1) converges very slowly when the parameters are not very small. Furthermore, when one parameter is negative, the adjacent terms in the infinite series could conceal each other such that their sum is dominated by roundoff errors. From experiments, we found that the so-called "Algorithm 707" in [65, 64] worked well most of time in computation of the hypergeometric filters ([100]). Also, as pointed out by an anonymous reviewer, the H filters may be numerically computed by inverse Fourier transforms from Eq. (4.1).

[2]Or a transformation surface in two parameter cases and a hyper-surface in more than two parameter cases



$h_{mn}(x, y)$ The Fourier Domain

Figure 4.8: Frequency Sampling in Our Implementation

For real value images, because the Fourier transform $I(f_x, f_y)$ has the property

$$I(f_x, f_y) = I^*(-f_x, -f_y),$$

only half of the plane needs to be sampled. From Eq. 4.5, the Nyquist frequency π is reached when

$$m \approx \sigma^2 \pi. \tag{4.54}$$

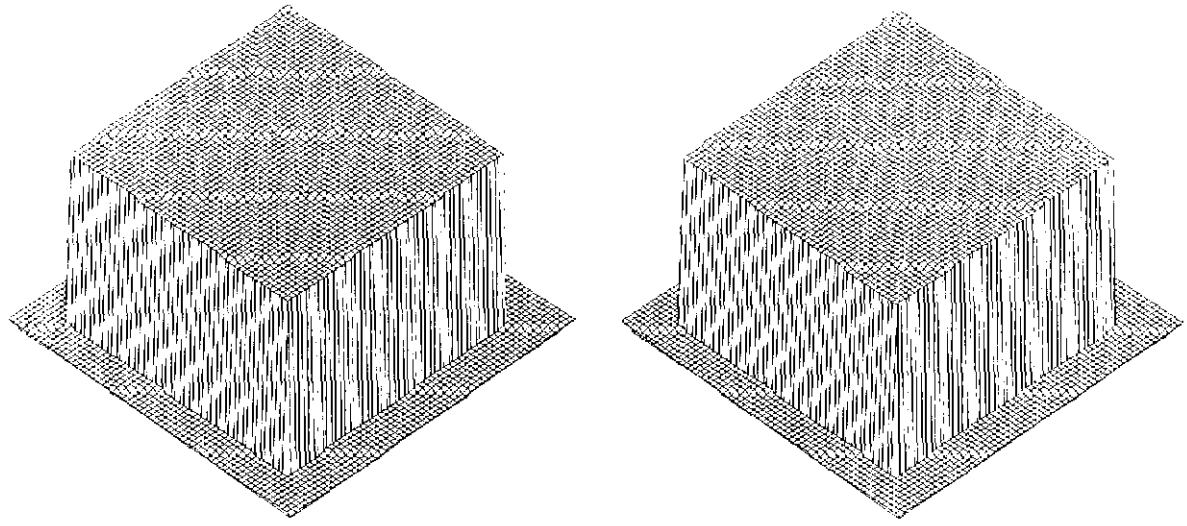
We implement our algorithm on a parallel machine (MasPar). Due to the memory limitation, we cannot use all the filters up to the Nyquist frequency in our implementation. Instead, we use (m, n), corresponding to (f_x, f_y), contained in the shadow area in Figure 4.8, in which

$$u_1 = \frac{\sqrt{M_1}}{\sigma}; \quad u_2 = \frac{\sqrt{M_2}}{\sigma};$$
$$v_1 = \frac{\sqrt{N_1}}{\sigma}; \quad v_2 = \frac{\sqrt{N_2}}{\sigma}.$$

Again, we believe that once the development of more powerful hardware allows us to fully explore the whole Fourier domain up to the Nyquist frequency, the results reported here should be improved considerably. Current running time of the hypergeometric filter approaches is about thirty minutes on MasPar.

4.7.2 Experiments on Focus

We implemented the hypergeometric filter approach for the focus problem using the algorithm sketched in Section 4.5. The transformation is modelled as a Gaussian



HFF1 HFF2

Figure 4.9: Computed σ_0^2 from the First Pair

	SA	MFF1	MFF2	HFF1	HFF2
The First Pair	0.070518	0.000749	0.000316	0.000522	0.000345
The Second Pair	0.190362	0.046419	0.007976	0.026760	0.003828

Table 4.1: RMS Errors of Hypergeometric Filter Approach

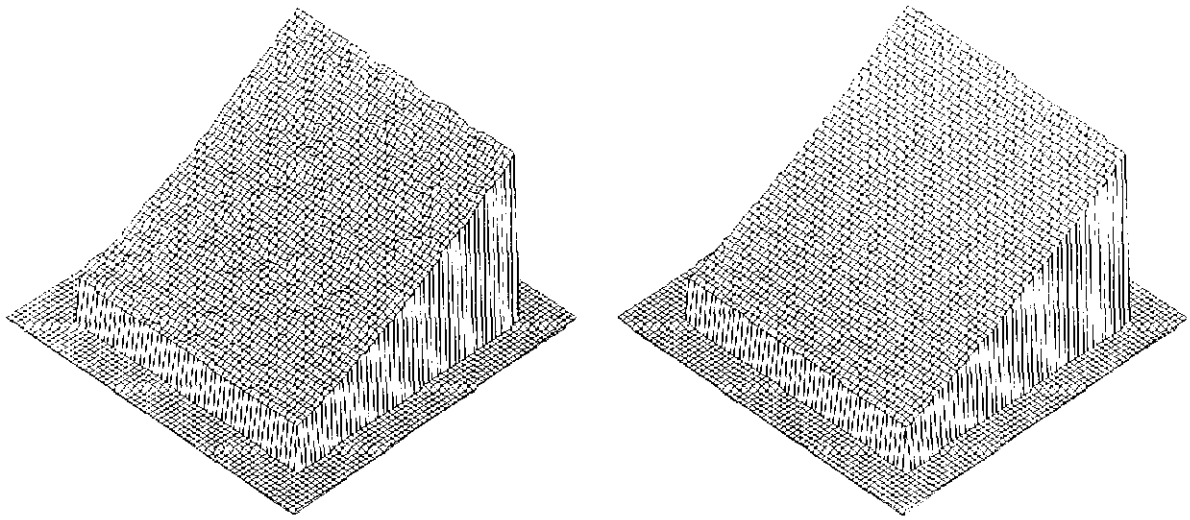
blurring:

$$T(f;\sigma_0) = e^{-f^2\sigma_0^2/2}. \tag{4.55}$$

For the synthetic image pairs shown in Figure 3.8, we used the hypergeometric filters with size $\sigma = 6.0$ and the frequency samplings $M_1 = N_1 = 0; M_2 = N_2 = 15$. Figure 4.9 and Figure 4.10 show the computed σ_0^2 from the first and second synthetic pairs by using the hypergeometric approach without foreshortening (HFF1) and with foreshortening (HFF2). Figure 4.11 illustrates the slope estimated from HFF2 for the second synthetic pair. We compare the results from different algorithms on the same image pairs in Table 4.1.

We also tested the hypergeometric filter approach on the real image pair shown in Figure 3.12. Figure 4.12 and Figure 4.13 show the computed blurring difference of the house image pair using HFF1 and HFF2, respectively. Finally, Figure 4.14 illustrates the slope estimations for the house image pair.

Compared to the results of MFF1 and MFF2 reported in the previous section, the



HFF1 HFF2

Figure 4.10: Computed σ_0^2 from the Second Pair

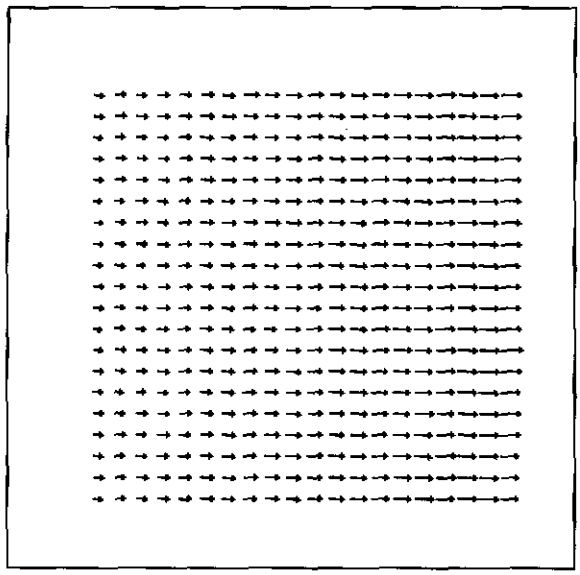


Figure 4.11: Slope Computed by HFF2 for the Second Pair

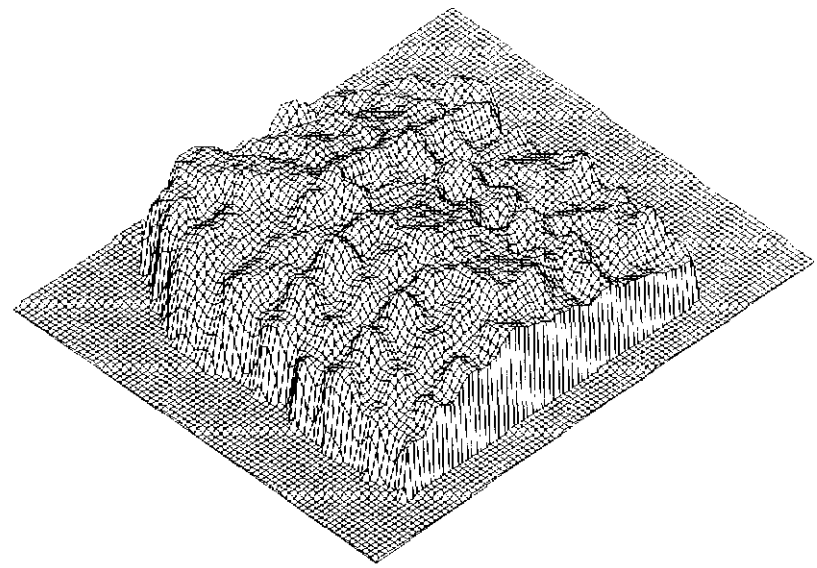


Figure 4.12: Blurring Difference from HFF1 for the House Pair

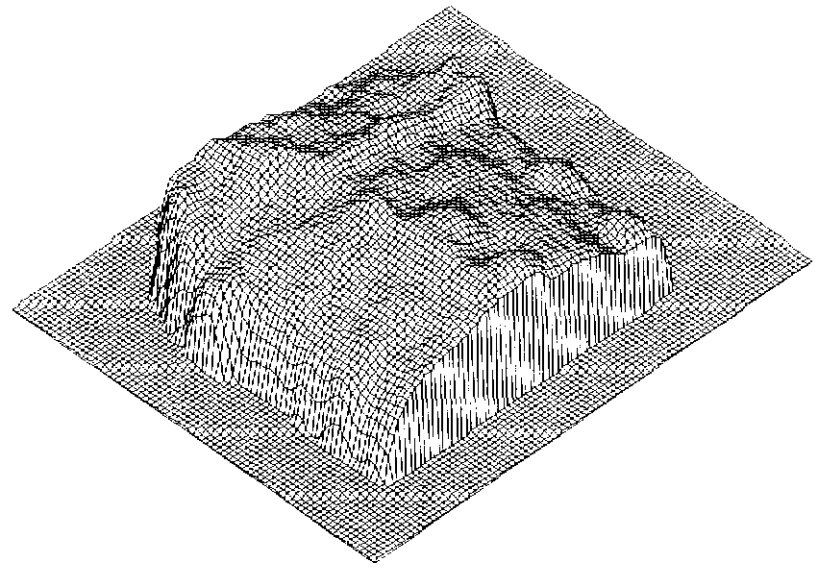


Figure 4.13: Blurring Difference from HFF2 for the House Pair

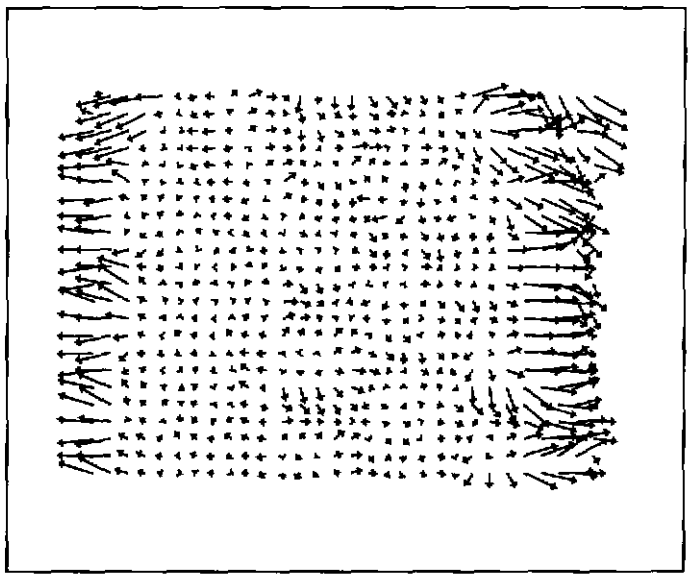


Figure 4.14: Slope for the House Pair

results of HFF1 and HFF2 are approximately in the same precision, which is consistent with our expectation that the moment filter approach and the hypergeometric filter approach are equally effective in modelling window effects and shift variance effects. The difference in results, we believe, is mainly caused by the fact the information used by the moment filter approaches and the hypergeometric approaches is at difference frequency bands. The hypergeometric filter approaches use low frequency information ($0 \leq f \leq 0.77$), while the moment filter approaches use middle frequency information ($0.31 \leq f \leq 1.30$).

4.7.3 Experiments on Stereo

We also implemented the hypergeometric filter approach for the stereo matching problem, in which the transformation is modelled as a spatial translation:

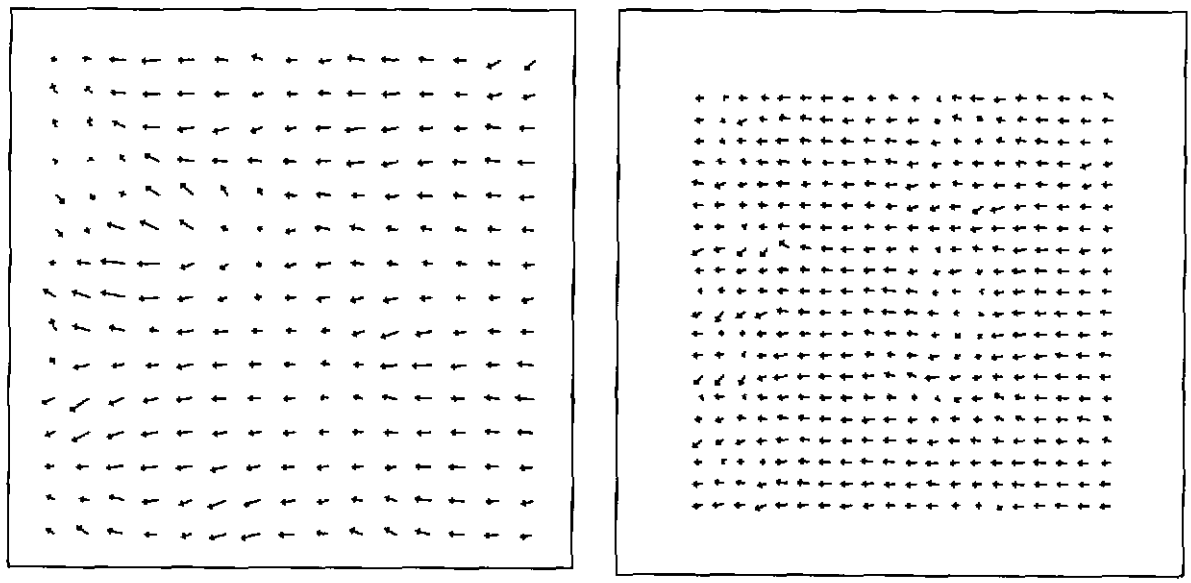
$$T(f; D) = e^{jfD}. \tag{4.56}$$

In our implementation, the window size parameter σ of the hypergeometric filters is set to 6.0, and the sampling of the frequency domain is $M_1 = N_1 = 0; M_2 = N_2 = 15$. For simplicity, we will denote the hypergeometric filter approach without foreshortening as HFS1, and the hypergeometric filter approach with foreshortening as HFS2.

First, we tested the hypergeometric filter approaches on the same synthetic image

	Uniformly Shifted Pair	Nonuniformly Shifted Pair	Tree Pair
KOA	0.008	0.035	0.016
MFS1	0.008	0.024	0.012
MFS2	0.006	0.020	0.010
HFS1	0.004	0.017	0.014
HFS2	0.005	0.006	0.010

Table 4.2: RMS Errors for Synthetic Images



Tree Pair Nonuniformly Shifted Pair

Figure 4.15: Foreshortening Estimated by HFS2

examples shown in Figure 3.15. Table 4.2 shows the RMS errors of the hypergeometric filter approaches. And Figure 4.15 shows the foreshortening estimated by HFS2. Overall, the performance of HFS1 and HFS2 is better than that of MFS1 and MFS2.

We also tested the hypergeometric filter approaches on real image pairs as shown in Figure 3.20 and Figure 3.23. Figure 4.16 and Figure 4.17 show the results from HFS1 and HFS2 on the cube image pair. Since the hypergeometric filter approaches compute disparity in every pixel location, the disparity values in the textureless background area are very noisy. The residual errors after plane fittings are shown in Table 4.3, together with the results from KOA, MFS1 and MFS2. Figure 4.18 and Figure 4.19 show the results from HFS1 and HFS2 on the house image pair. The residual errors after plane fittings are shown in Table 4.4, together with the results from KOA, MFS1

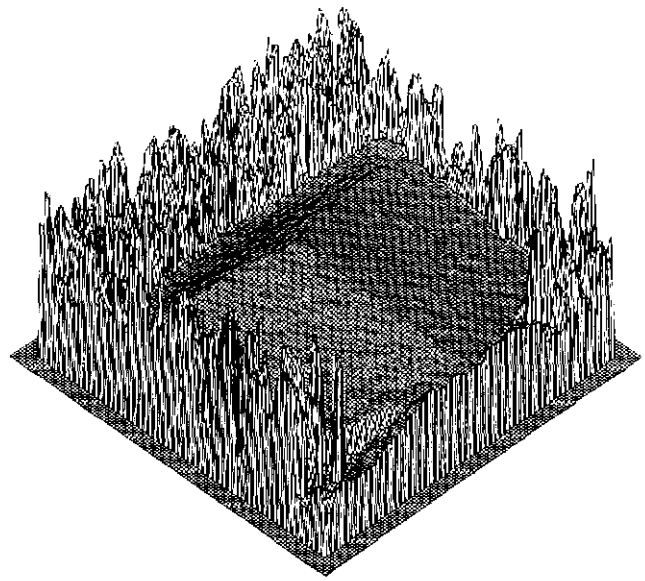


Figure 4.16: Disparity of the Cube Pair from HFS1

	Top Patch (6690 Pixels)	Left Patch (16025 Pixels)	Right Patch (19625 Pixels)
KOA	0.063	0.032	0.027
MFS1	0.039	0.024	0.020
MFS2	0.018	0.020	0.016
HFS1	0.040	0.030	0.025
HFS2	0.020	0.021	0.019

Table 4.3: RMS Errors of the Cube

and MFS2. In both pairs, the performances of HFS1 and HFS2 are slightly worse than that of MFS1 and MFS2,respectively, but better than that of KOA. The main reason for this is that the hypergeometric filter approaches used the low-frequency information ($0 \leq f \leq 0.77$), while the moment filter approaches used the middle frequency information ($0.31 \leq f \leq 1.04$). When the same amount of noise contaminated low-frequency and high-frequency information, the results derived from low-frequency information were damaged more severely. On the average, as we expected, the hypergeometric filter approaches and moment filter approaches are equally effective in modelling the window and foreshortening effects in stereo matching.

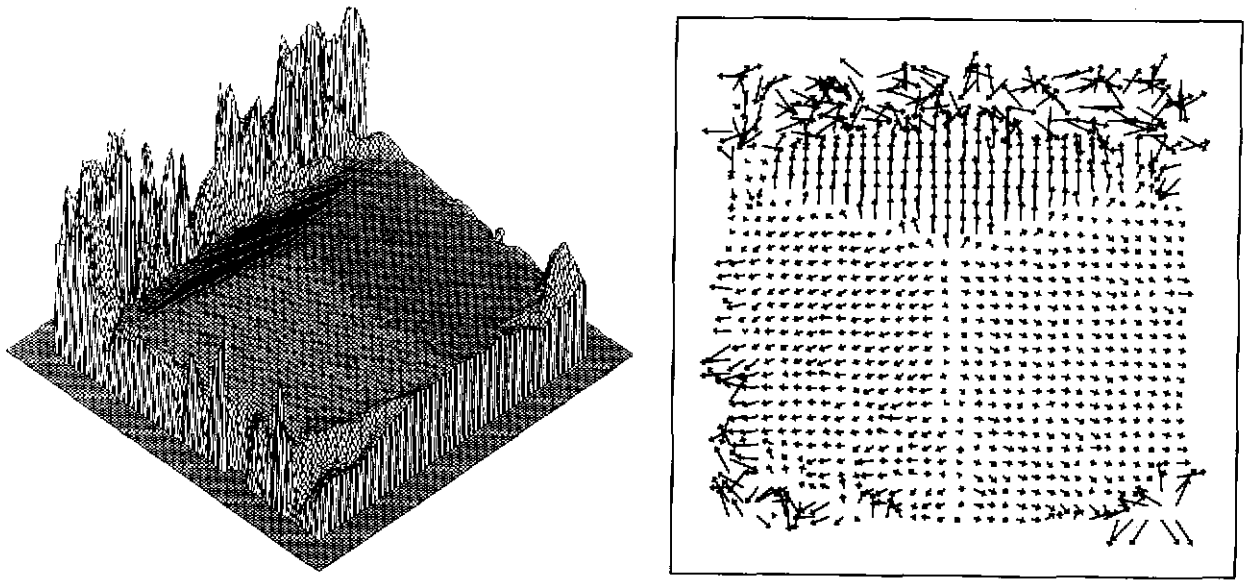


Figure 4.17: Disparity and Slope of the Cube Pair from HFS2

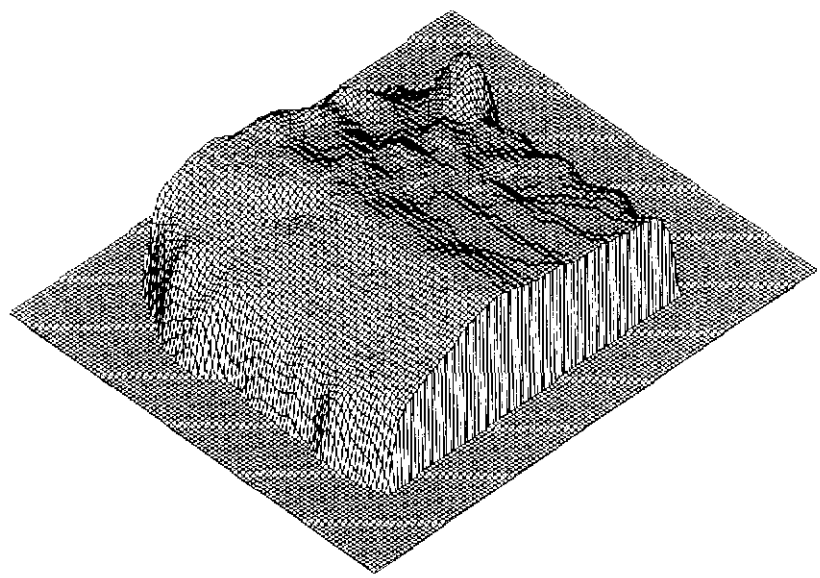


Figure 4.18: Disparity of the House Pair from HFS1

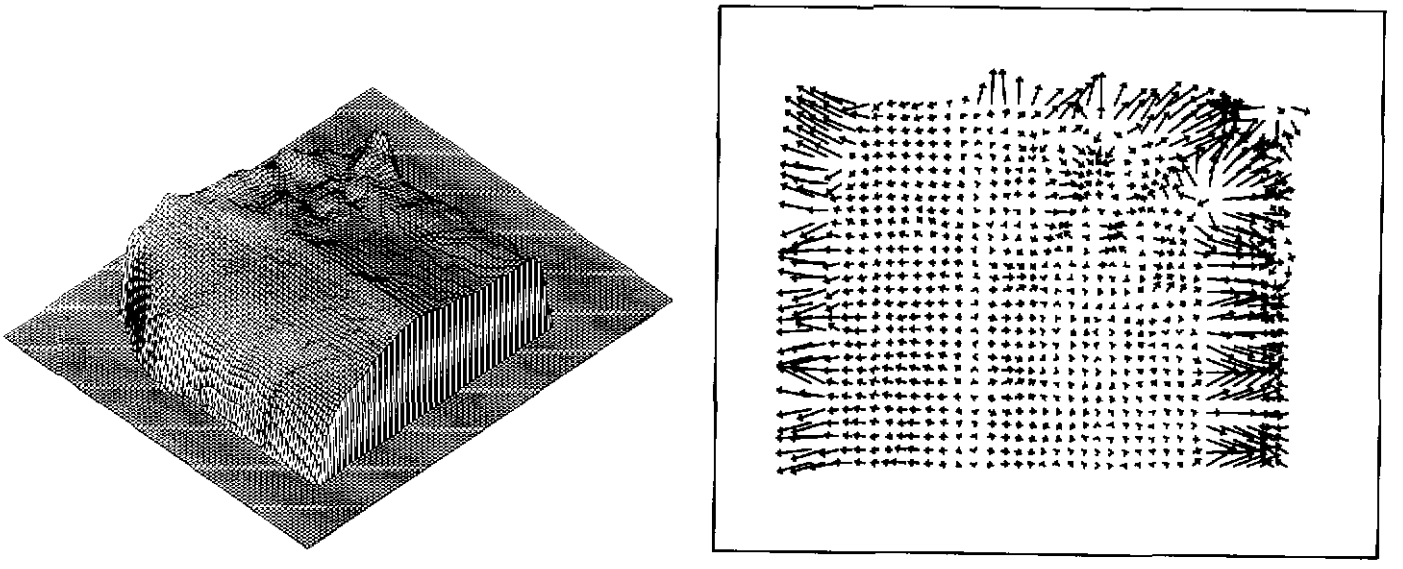


Figure 4.19: Disparity and Slope of the House Pair from HFS2

	Left Patch (11260 Pixels)	Right Patch (13204 Pixels)
KOA	0.032	0.031
MFS1	0.028	0.017
MFS2	0.019	0.017
HFS1	0.027	0.019
HFS2	0.024	0.017

Table 4.4: RMS Errors of the Toy House

4.7.4 Experiments on Optical Flow

There have been many algorithms to compute optical flow from two or more images. Among the published work, Barron *et al* [11] provided a quantitative measure for a few of the optical flow techniques. For comparison, we will demonstrate the capability of our new technique on the same test images. While most of the techniques presented in [11] use a sequence of images to reduce error, our technique achieves better results and uses only two adjacent frames. (In the next chapter, we will explore whole image sequences.) No pre-smoothing in the spatial domain or the temporal domain is done; therefore the quantization error and noise is much higher. The error measurement we use is the angular measure in [11]:

$$\psi_e = \arccos\left(\frac{uu_t + vv_t + 1}{\sqrt{u^2+v^2+1}\sqrt{u_t^2+v_t^2+1}}\right), \tag{4.57}$$

where (u, v) is the computed optical flow, (u_t, v_t) is the true optical flow, and ψ_e is the angle between $(u, v, 1)$, and $(u_t, v_t, 1)$. We do not use the standard deviation measure in [11] because it is hard to interpret. All angular errors are in degrees (°).

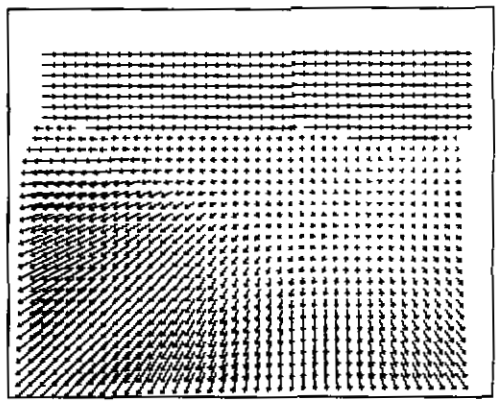
Without any post processing, the approach proposed here will compute a velocity at every pixel. Two well-known intrinsic problems have to be addressed, i.e., the aperture problem and the no-texture problem, both resulting from a lack of intensity variation around a pixel. If the error estimation procedure indeed works properly, the covariance matrix it computes will have one large eigenvalue in the case of the aperture problem, and two large eigenvalues in the case of no texture at all. The corresponding eigenvector should indicate the direction in which great uncertainty occurs. To compare our results with those from other techniques, we threshold the computed optical flow by the following formula:

$$\frac{E_{\max}}{u^2+v^2+1} < T_e, \tag{4.58}$$

in which $E_{\max}$ is the largest eigenvalue of the covariance matrix, and T_e is the threshold. By setting T_e to different values, we can obtain a curve of error versus density, as we will show later.

For the Yosemite sequence (Figure 4.20), we used the 9th and 10th frames to compute the optical flow since they were the two frames in the middle of the sequence. The size of the H filters was $\sigma = 7.0$, and the sampling limits in Figure 4.8 were: $M_1 = N_1 = 0, M_2 = N_2 = 10$, which corresponded to the area in the Fourier domain:

The Ninth Frame in the Yosemite Sequence



The Correct Flow

Figure 4.20: Yosemite Image Sequence

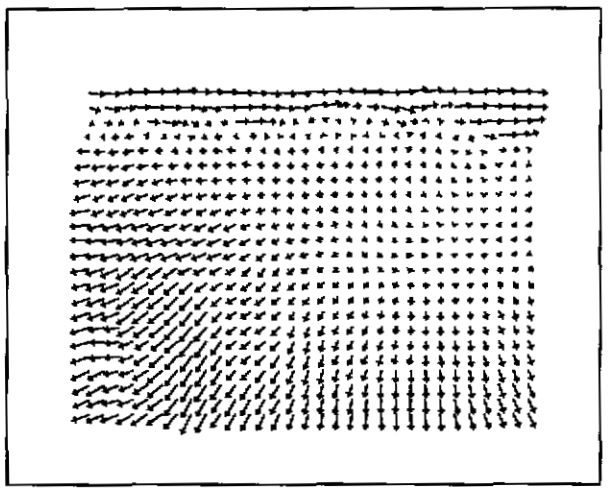


Figure 4.21: Computed Optical Flow of the Yosemite Scene

$(0.0 \leq f_x \leq 0.45, -0.45 \leq f_y \leq 0.45)$. In other words, we used only information whose frequency was lower than one-sixth of the Nyquist frequency. Figure 4.21 shows the raw (unthresholded) optical flow computed. The curve of the density versus average error is shown in Figure 4.22 along with the results reported in [11]. For further comparison, we list the average angular error and the corresponding density in Table 4.5. We can see that except for a couple of techniques which imposed smoothness constraints at 100% density, our approach has the best results for all densities, despite the fact that we used only low frequency information from only two frames.

The monotonic increase of the average error with respect to more restrictive eigen-

Average Error	Density
2.076670	1.67%
2.094967	3.03%
2.172882	5.94%
2.287890	9.25%
2.539107	16.01%
2.855300	23.05%
3.090271	28.95%
3.227844	33.87%
3.329092	37.90%
3.427297	40.97%
3.814695	48.93%
4.202163	55.16%
4.533337	60.26%
5.056063	68.17%
5.464669	72.73%
6.321297	79.80%
6.990403	84.22%
7.913178	89.40%
8.625618	92.77%
8.970320	94.59%
10.121454	100%

Table 4.5: Density vs. Average Error for the Yosemite Sequence

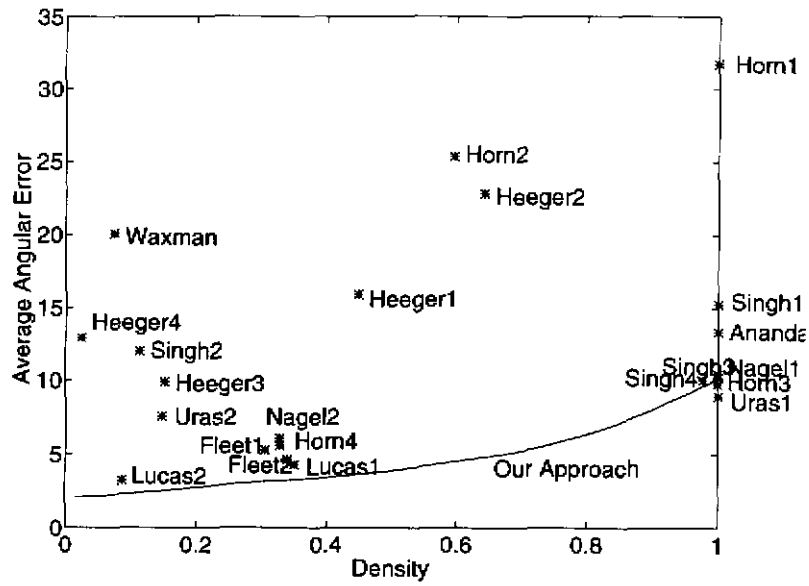


Figure 4.22: Average Error for the Yosemite Scene

value threshold indicates that the error estimation indeed represented the uncertainty properly. Furthermore, as a covariance matrix can be represented by an ellipse whose two principal directions are the two eigenvectors, and whose radii are the square roots of the corresponding eigenvalues, we superimpose the ellipses on top of the darkened image in Figure 4.23. From the size of the ellipses, we see that the top portion and lower-left portion have the largest errors. This situation is consistent with Figure 4.21 and Figure 4.20. We also observe that at locations where the aperture problem is obvious, e.g., near a single edge, the ellipses were elongated in the direction of the edges, which represented the lack of information in that direction.

The second synthetic sequence was the translating-tree sequence (Figure 4.24) from [11]. We used the 20th and 21st frames to compute the optical flow. The size of the H filters is $\sigma = 3.5$, and the sampling limits in Figure 4.8 were: $M_1 = N_1 = 0, M_2 = N_2 = 15$, which correspond to the area in the Fourier domain ($0.0 \leq f_x \leq 1.233, -1.233 \leq f_y \leq 1.233$). In other words, the highest frequency of the information we used was one-third of the Nyquist frequency. Figure 4.25 shows the unthresholded optical flow. Except for a few locations where there was no texture, the computed optical flow was fairly consistent with the ground truth. The curve of the density versus average error is shown in Figure 4.26. For further comparison, we list the density and corresponding average angular error in Table 4.6. Again we can see that our approach, while based upon only two adjacent frames and only low frequency

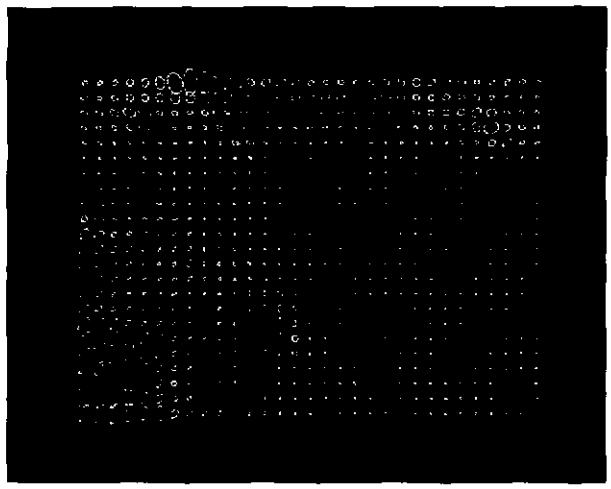


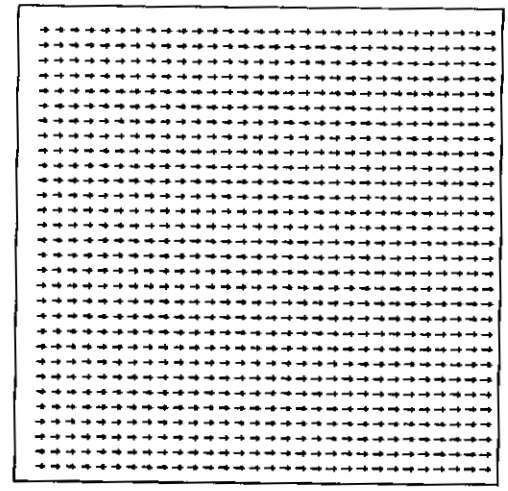
Figure 4.23: Uncertainty Estimation for the Yosemite Scene

information, nevertheless outperformed all the other algorithms at all densities.

Figure 4.27 shows the uncertainty ellipses superimposed on the darkened image. We can see that the uncertainty estimation not only detects the lack of information along certain direction, e.g., the aperture problem along trunks, but it also represents large errors caused by blank areas between trunks.

A trickier synthetic sequence is the diverging tree sequence from [11, 26] as illustrated in Figure 4.28. The significant difference between this sequence and previous sequences is that the underlying optical flow has a very large gradient, i.e., the image velocity changes rapidly as the location in the image changes. In fact, if we shift the correct optical flow by only two pixels horizontally, the average angle between the shifted flow and the original optical flow is 1.37° compared to 0.08° for the translating tree sequence and 0.65° for the Yosemite sequence. If the window size is larger than two pixel widths, we can expect that the average error of the computed optical flow will be much larger than 1.37°. In fact, as we will show in the next section, using the affine transformation model can decrease the average angular error by almost half. Therefore, the major cause of the error in this particular example is an incorrect transformation model, i.e., the transformation between two images should not be modelled as a translation. While some results reported in [11] were very close to, or even smaller than, 1.37°, we believe that the only reason those algorithms could reach such precision was the averaging in the temporal direction. Actually, as the authors pointed out in [11], the errors of the differential approaches increased substantially when only two images were used. And because the temporal averaging is crucial in

The 20th Frame in Translating Tree Sequence



The Correct Flow

Figure 4.24: Translating Tree Image Sequence

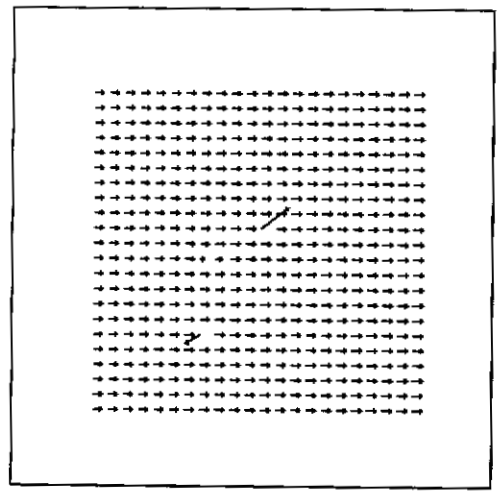


Figure 4.25: Computed Optical Flow of the Translating Tree Scene

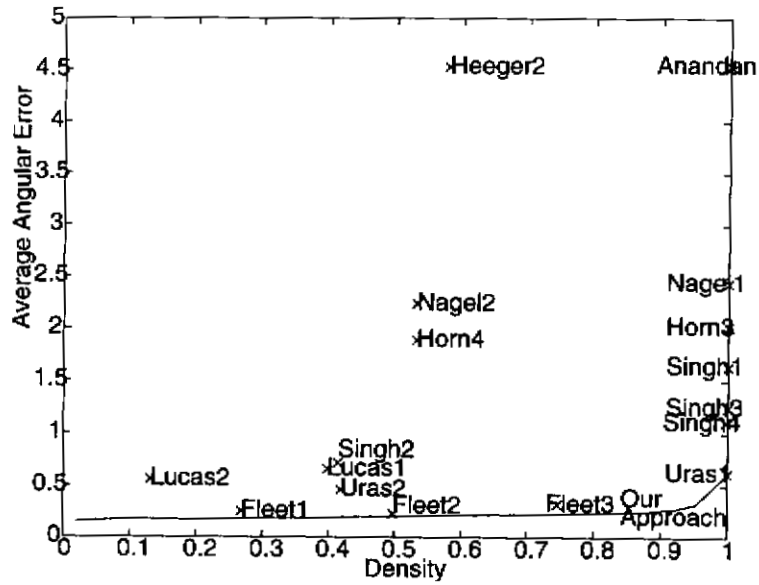


Figure 4.26: Average Error for the Translating Tree Scene

Average Angular Error(°)	Density
0.149777	1.99%
0.171012	10.05%
0.171912	20.89%
0.174841	26.22%
0.185415	36.59%
0.199455	45.09%
0.206545	52.08%
0.213830	57.48%
0.218359	61.58%
0.222904	64.73%
0.237176	78.72%
0.242210	84.52%
0.248934	86.27%
0.268584	90.39%
0.323182	95.25%
0.618968	100%

Table 4.6: Density vs. Average Error of the Translating Tree Scene

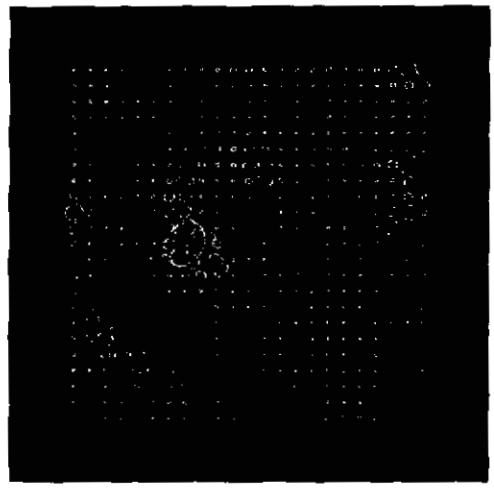
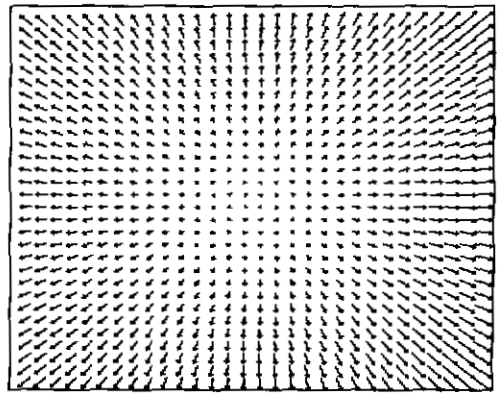


Figure 4.27: Uncertainty Estimation in Translating Tree Scene

The 20th Frame in Diverging Tree Sequence



The Correct Flow

Figure 4.28: The Diverging Tree Image Sequence

this sequence, we argue that the comparisons made in [11] (Table 5) is not fair for those algorithms with shorter temporal support.

Using the same parameters as those in the translating tree sequence, we computed the optical flows from adjacent frames. Table 4.7 shows the average errors when using only the 20th and 21st frame, and the average errors when we average fourteen computed optical flow fields from the 13th frame to the 27th frame. First of all, they did verify our observation that the average angular error close to 1.37° could be only achieved by averaging along the temporal direction. Secondly, the average error of our algorithm after averaging the fourteen optical flow fields was smaller than all

Density	Average Error from Two Frames	Average Error from Fifteen Frames
10%	2.191357	1.629412
20%	2.188124	1.595904
30%	2.292716	1.626959
40%	2.386003	1.647144
50%	2.483744	1.696696
60%	2.570150	1.724406
70%	2.686011	1.784787
80%	2.825161	1.863550
90%	3.038625	1.996772
100%	3.408894	2.195650

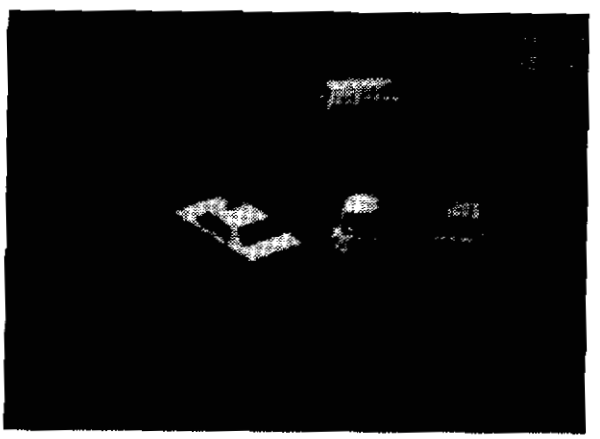
Table 4.7: Average Error of the Diverging Tree Sequence

the results[3] except those from Fleet and Jepson's algorithm which achieved errors less than 1.37°. We suspect two reasons for the outstanding performance of Fleet and Jepson's algorithm reported in [11]. One is that the extension of a 2D image sequence to a 3D spatiotemporal signal is a better way of temporal averaging than the simple averaging of the optical flows. The other reason is that the implementation of Fleet and Jepson's algorithm utilizes *only* high frequency information which is more robust against a large velocity gradient.

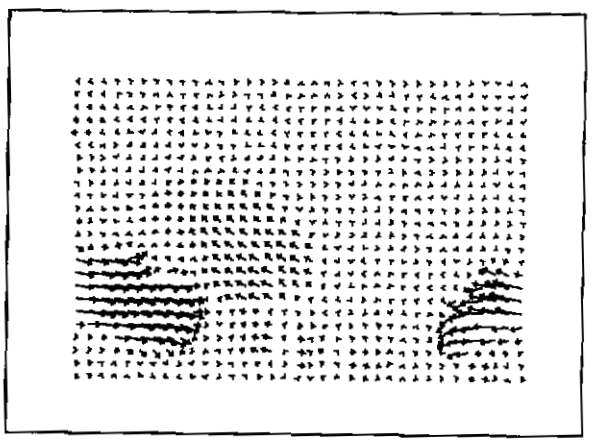
We also tested our algorithm on the four real image sequences in [11] as shown in Figure 4.29 through Figure 4.32. For all the real image sequences, we took two frames in the middle of each sequence to compute optical flows. The size of the H filters was $\sigma = 4.5$, and the sampling limit in Figure 4.8 was $M_1 = N_1 = 0, M_2 = N_2 = 15$, which corresponded to the area in the Fourier domain ($0.0 \leq f_x \leq 0.86, -0.86 \leq f_y \leq 0.86$). For each experiment, we show four images: the middle frame in the sequence, the unthresholded optical flow, thresholded optical flow and the uncertainty estimation.

Though there is no objective measure to compare our results with those reported in [11], we can still conclude that, on average, the performance of our algorithm is superior to those in [11] in both density and precision, despite the fact that we used only two frames in each sequence.

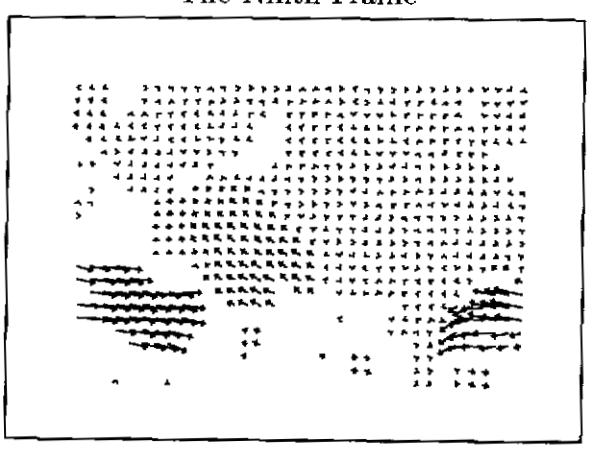
[3]All results reported in [11] used at least fifteen frames.



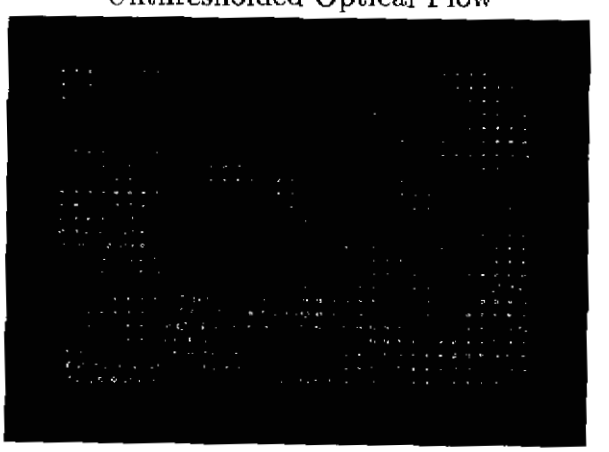
The Ninth Frame



Unthresholded Optical Flow

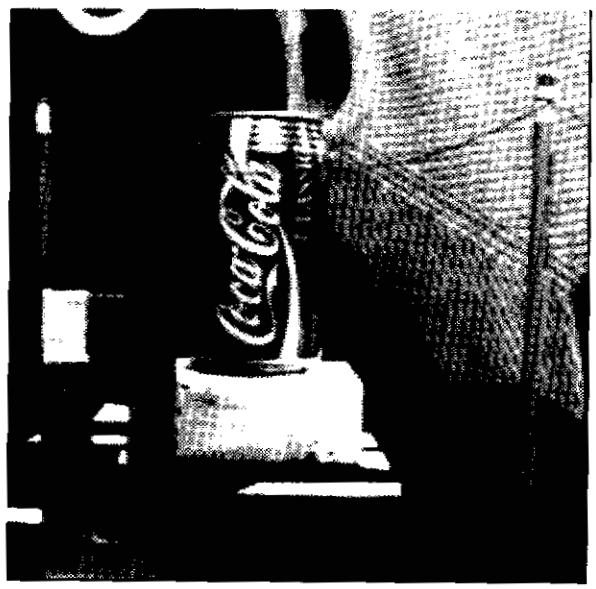


Thresholded Optical Flow $T_e = 0.005$

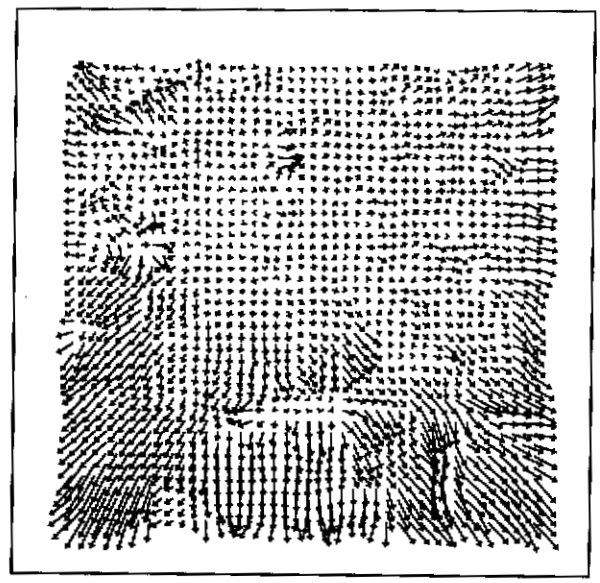


The Uncertainty Estimation

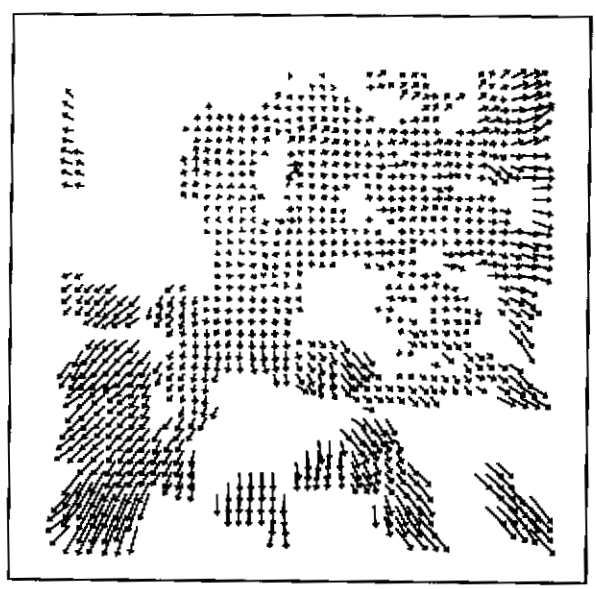
Figure 4.29: Optical Flow of the Taxi Sequence



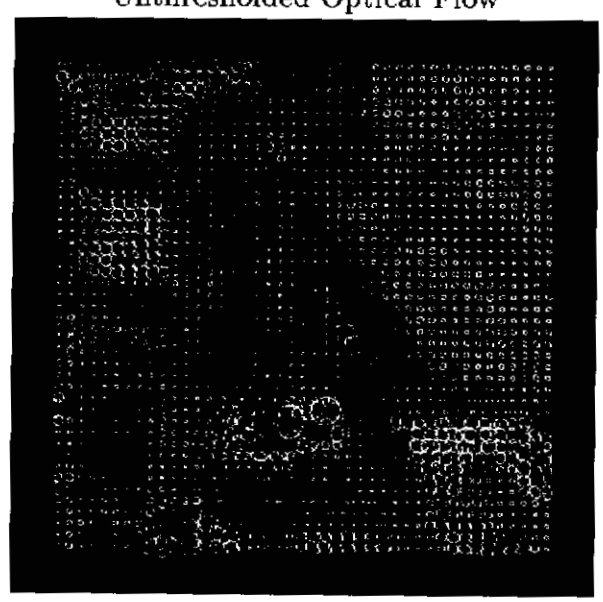
The Seventh Frame



Unthresholded Optical Flow

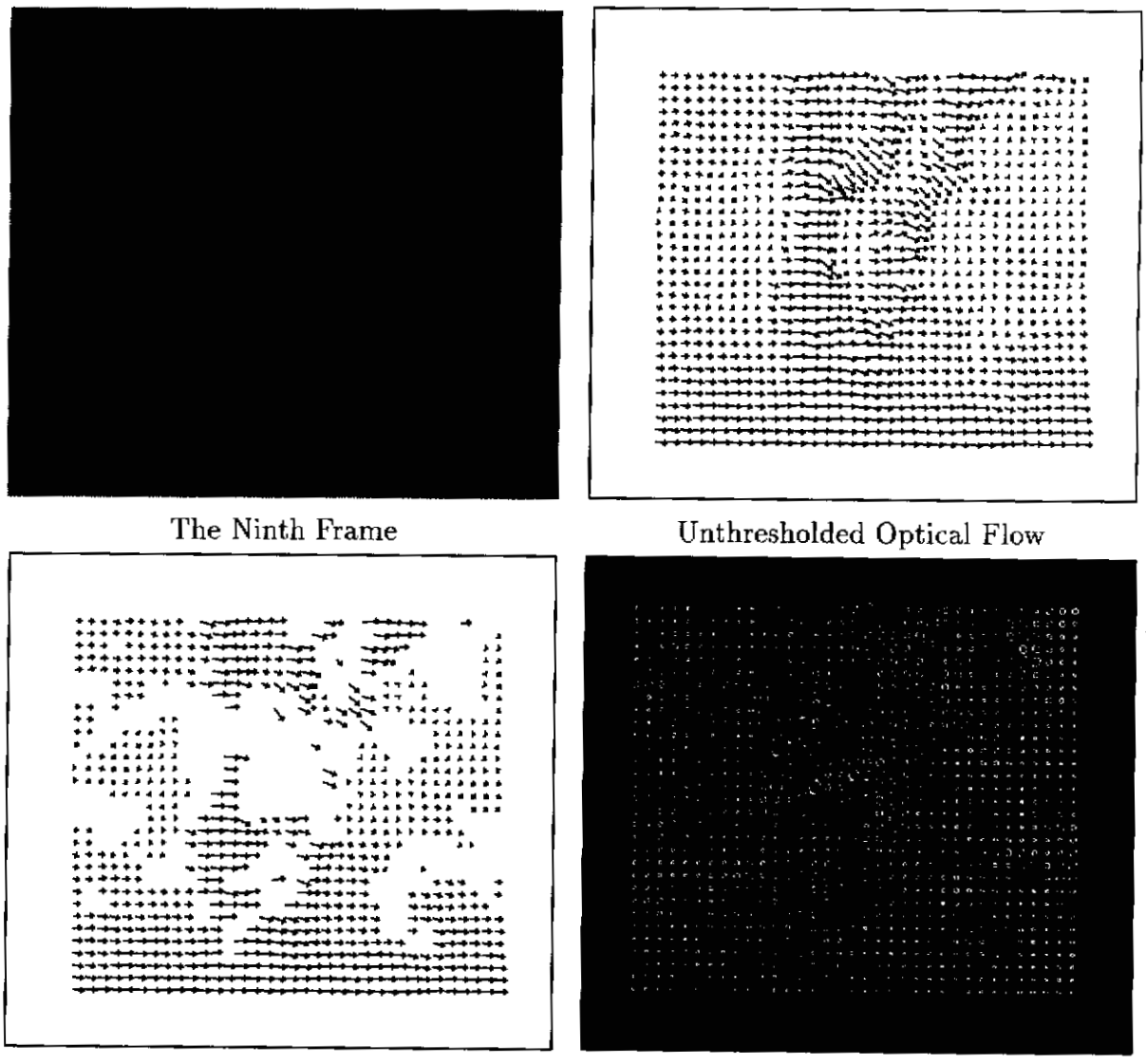


Thresholded Optical Flow $T_e = 0.001$



The Uncertainty Estimation

Figure 4.30: Optical Flow of the NASA Coke Sequence



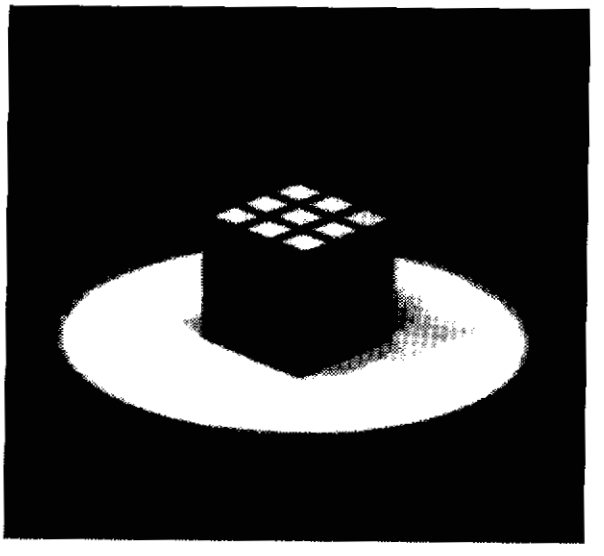
The Ninth Frame

Unthresholded Optical Flow

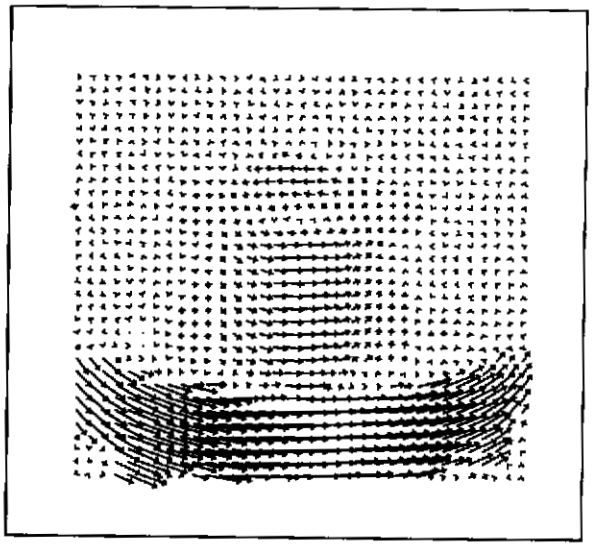
Thresholded Optical Flow $T_e = 0.001$

The Uncertainty Estimation

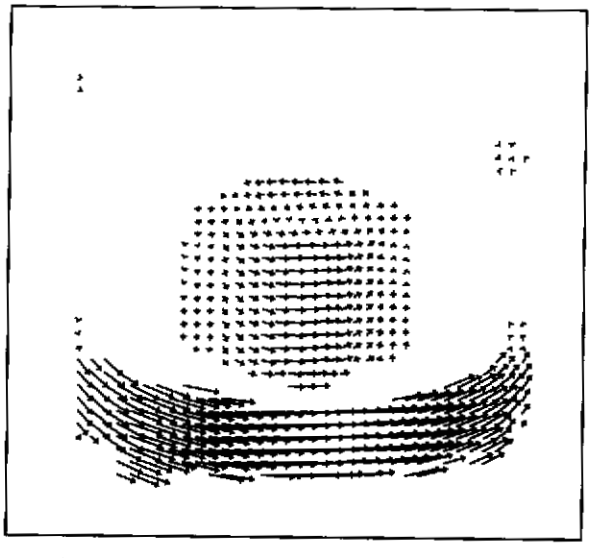
Figure 4.31: Optical Flow of the SRI Tree Sequence



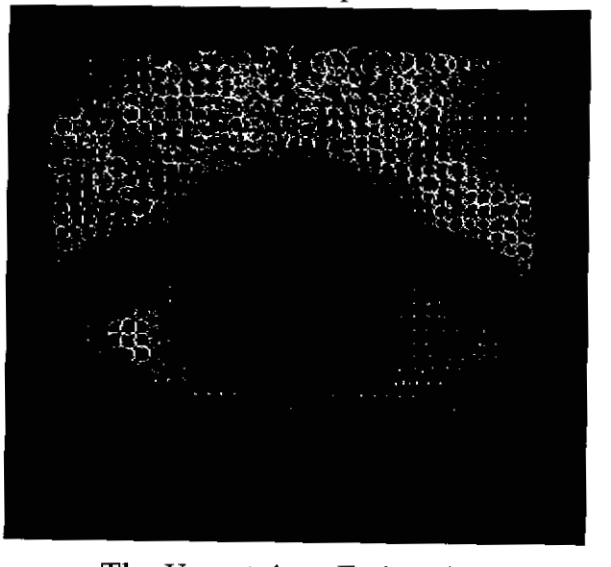
The Ninth Frame



Unthresholded Optical Flow



Thresholded Optical Flow $T_e = 0.001$



The Uncertainty Estimation

Figure 4.32: Optical Flow of the Rubic Sequence

4.7.5 Experiments on Affine Matching

We first demonstrate the ability of our approach to recover six affine parameters from a relatively small local patch. The affine transform of the original image (from [15]) in Figure 4.33(a) is computed by linear interpolation. Then we try to recover these parameters at the center of the image. Figure 4.33(b) shows an example of an affine transformed image. In this experiment, we set the window size parameter σ of H filters to 7.0 pixel-width, and the frequency sampling to $M_1 = N_1 = 0; M_2 = N_2 = 15$. In other words, we are using just the low-frequency information contained in a small area around the image center to compute the six affine parameters.

(a) Original Image (b) After Affine Transform

Figure 4.33: Image before and after Affine Transform

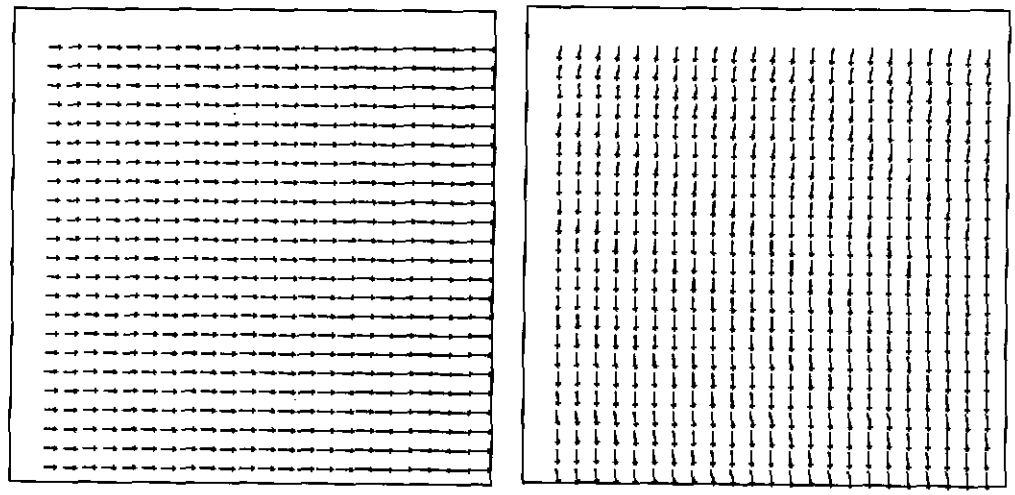
Table 4.8 shows the true and recovered affine parameters in four experiments. All numbers are in pixel width. The RMS error for the recovery of (u_0, v_0) is 0.051 pixel-width, and the RMS error for the recovery of (u_x, u_y, v_x, v_y) is 0.0057 pixel-width per pixel-width.

In the second and third experiments, we tried to find affine matching parameters for every pixel. We first used two frames from the diverging tree sequence in [11]. The optical flow of this sequence (Figure 4.28) contained large gradients which made the affine model necessary. Figure 4.34 illustrates the true gradients of (u, v), and Figure 4.28 shows the true (u, v).

The window constant of H filters was $\sigma = 7.0$, and the frequency sampling limits were $M_1 = N_1 = 0; M_2 = N_2 = 15$. Table 4.9 compares the precision of computed

	u_0	v_0	u_x	u_y	v_x	v_y
True	1.0000	1.0000	0.1000	0.0000	0.0000	0.1000
Recovered	1.0647	0.9505	0.0872	0.0013	0.0038	0.0920
True	0.5000	0.8000	0.0500	0.0300	0.0500	0.0300
Recovered	0.5409	0.8046	0.0481	0.0261	0.0538	0.0332
True	2.0000	0.5000	0.0100	0.0800	0.0700	0.0400
Recovered	2.0842	0.5372	0.0106	0.0710	0.0733	0.0328
True	0.0000	2.5000	0.0300	0.0200	0.0800	0.0700
Recovered	0.0600	2.5067	0.0253	0.0140	0.0787	0.0647

Table 4.8: Recovered Affine Parameters from a Small Patch



∇u ∇v

Figure 4.34: True Gradients of Optical Flow

	Translational Matching	Affine Matching
Angular Error of (u, v)	3.408894	1.900784

Table 4.9: Comparison under Translation and Affine Model

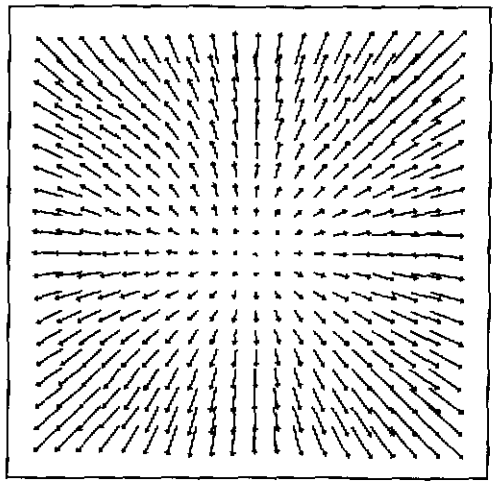


Figure 4.35: Recovered (u_0, v_0) from Diverging Tree Images

optical flow under the translation and affine model. Clearly the translation model does not fit well in this case. Figure 4.35 and Figure 4.36 show the computed six affine parameters at every pixel. We can see that the recovered six affine parameters were very precise except in areas with no texture. The RMS error of recovered (u_0, v_0) is 0.032 pixel-width, and the RMS error of recovered (u_x, u_y, v_x, v_y) is 0.0031 pixel-width per pixel-width.

We also applied the affine matching model to a real image pair as in Figure 4.37. The camera motion between the two images was a horizontal shift (0.5 inch) followed by a rotation ($\sim 2°$) about the vertical axis. Therefore, the ground truth of the translation parameters (u_0, v_0) was always horizontal, i.e $v_0 = 0$ and $\nabla v = (0, 0)$. The object was a textured cube located about 2.5 meters away from the camera. Figure 4.38 and Figure 4.39 illustrate the six affine parameters recovered from these two images.[4] Because the surfaces of the object were slanted with respect to the image plane, the direction of the affine deformation ∇u was the the 2D projection of the surface normal, and the magnitude of ∇u indicated the magnitude of slant. Again,

[4]In these graphs, The scale used for (u_0, v_0) is larger than that for ∇u and ∇v. But the scales for ∇u and ∇v are the same.

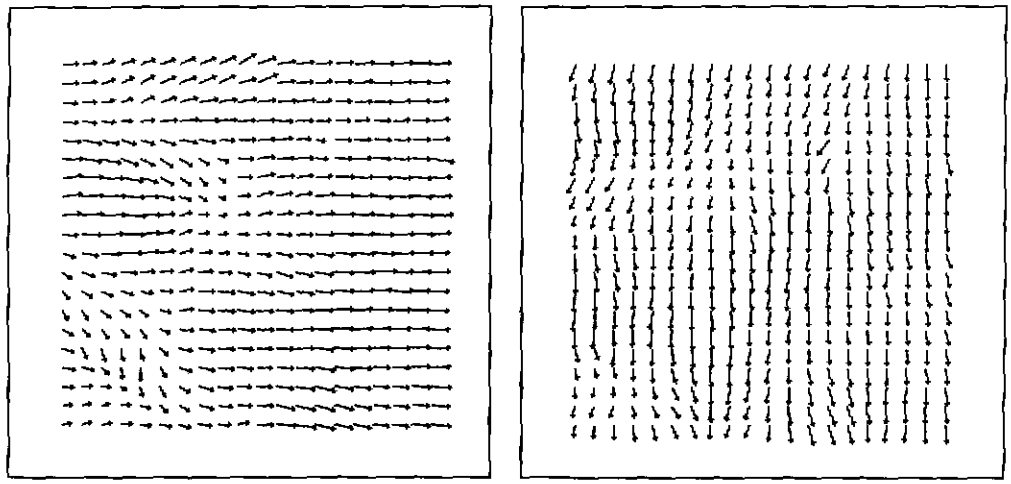


Figure 4.36: Recovered Gradient Parameters from Diverging Tree Images

using only local information, our algorithm robustly recovered six affine parameters for every pixel.

4.8 Summary

We have presented in this chapter a unified approach based on hypergeometric filters to the translational and affine matching problems as well as to other matching problems. Because of their recursiveness, the effects caused by the finite window size and non-zero gradients can be analytically expressed. Experimental results demonstrated high precision of this approach.

The original question we raised in Section 1.1 regarding high precision image matching is: *How to compute p precisely given only finite-duration signals as in Figure 1.1?* After going from Gabor filters to moment filters to hypergeometric filters, we can now conclude that, if the transformation model between two images is accurate and we use enough number of hypergeometric filters, the matching results are as precise as the image data allows because the only error source left is noise in images. On the other hand, this conclusion does not preclude further improvements on image matching since, in real images, it is rather difficult to have an accurate transformation model *a priori* as shown in the example of the diverging tree sequence.

Figure 4.37: Images of a Textured Cube

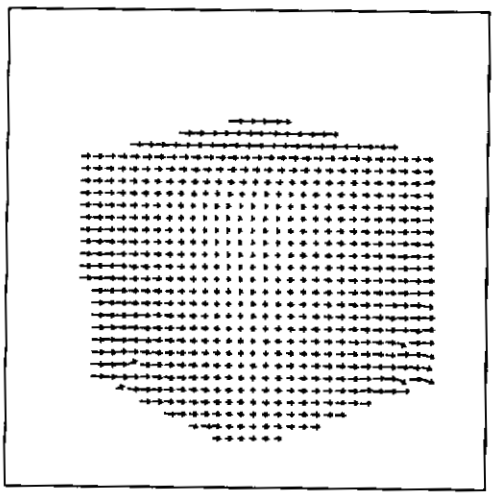
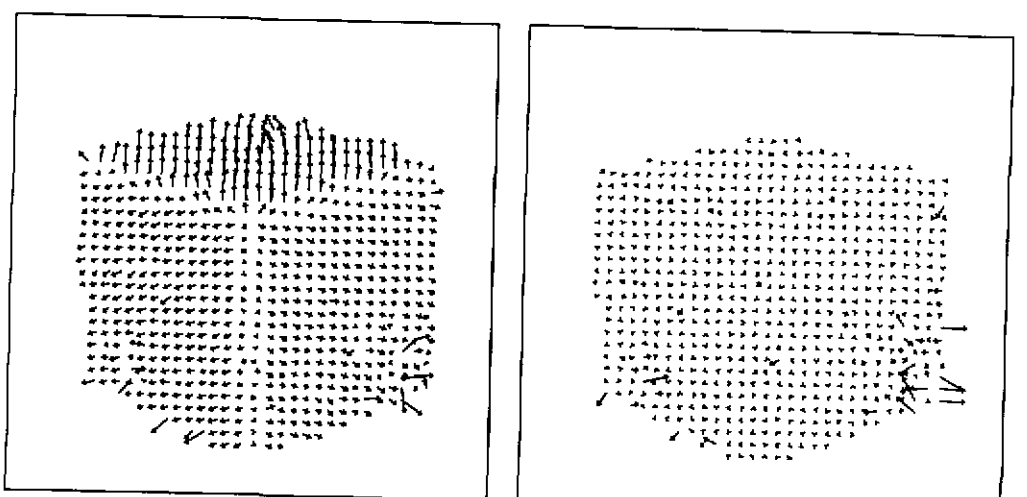


Figure 4.38: Recovered (u_0, v_0) from the Cube Images



∇u ∇v

Figure 4.39: Recovered Gradient Parameters from the Cube Images

Chapter 5

Dense Structure from Optical Flow Sequence

5.1 Introduction

Now that we have an algorithm to register images precisely, the next question is how we can use the results from the image matching. The most important motivation for high precision image matching is that such results can be converted into high precision 3D information. But in many cases, this conversion is not a straightforward mapping. In this chapter, we study one such vision problem: 3D structure from 2D image motion.

Structure from motion has been one of the most active areas in computer vision during the past decade. The idea is to recover structure or shape information from a sequence of images taken under unknown camera motions. Most approaches proposed in the literature can be classified according to whether they are based upon *features* or *optical flows*. Though there have been several successful feature-based SFM systems, the extension of those approaches to apply on optical flows is difficult in computation. And most of current flow-based approaches are unsuccessful in handling either multiple frames or noisy optical flows. In this paper, we show a flow-based approach which is capable of delivering accurate *dense* structure from multiple, real and dense optical flows.

Feature-based methods compute the structural information at feature locations by analyzing their 2D motion in images. Examples of such systems are reported by Tsai and Huang [89], Tomasi and Kanade [88], Broida *et al* [16] and Azarbayejani and

Pentland [7]. Because the whole analysis is limited to features which usually number no more than hundreds, the results from those systems yield very sparse shape information. While stripping a full-resolution image to a handful of features may greatly simplify the algorithm and the computation, most of the information contained in the image is lost. In many applications such as model acquisition, inspection and navigation, dense structural information is more desirable.

Traditional flow-based methods, such as reported by Bruss and Horn [17], Weng *et al* [93], Heeger and Jepson [35], Adiv [3], have concentrated on either solving the problem of recovering motion and structure from a single optical flow field or using very low resolution optical flows. As far as we know, little has been done to achieve a dense structure-from-motion system except Heel's work in [36]. Unfortunately, as pointed out by Tomasi in [88], whether Heel's algorithm converges is still an open question. Overall, the difficulties of such a system arise from two main factors:

- *Computation*: While a feature-based method can easily afford an $O(N^2)$ or $O(N^3)$ algorithm where N is the number of features, a flow-based method cannot even afford an $O(N^2)$ algorithm where N is the number of pixels.

- *Accumulation*: While a feature-based method can accumulate structural information for features because they are tracked across many frames, optical flows usually cannot be used to track pixels because their measurements are uncertain. In other words, a feature-based approach quantifies the image information as either totally unreliable or reliable, but a flow-based approach has to use information of a spectrum of reliability. Therefore, it is impossible to accumulate structural information by tracking all pixels across many frames in a flow-based method.

This chapter shows our approach to overcome these difficulties. We demonstrate a system which *incrementally* accumulates *dense* structural information from a sequence of optical flows (as computed in the previous chapter). The system, which is also described in [102, 103], has the following features:

- The system is based on EKF (extended Kalman filtering) as proposed in [16]. We will show in our experiment that the nonlinearity problem is not a very serious problem even when the initial data are very crude.

- We represent the uncertainty information at every pixel location as a scalar which represents the independent uncertainty, and a vector which represents the

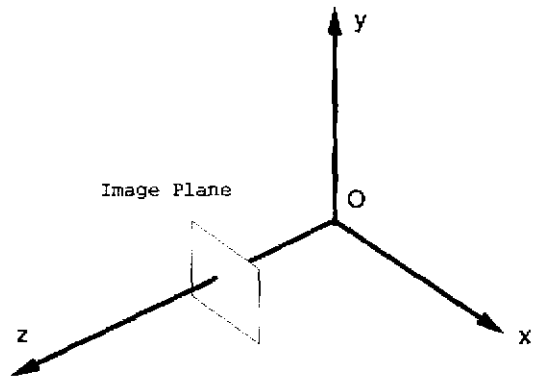


Figure 5.1: The Camera Coordinate

correlated uncertainty. By employing the representation and other mathematical techniques such as the Sherman-Morrison-Woodbury inversion and principal component analysis, we can achieve an $O(N)$ numerical algorithm for every frame in Kalman filtering.

- The underlying motion of the camera can be discontinuous. Unlike many EKF-based approaches, ours computes the initial motion of every frame independently.
- We propose the concept of "dynamic motion parameterization" which means that a different parameterization of the six motion parameters is used at every frame. Such a dynamic parameterization ensures that the optical flow is equally sensitive to each parameter and, therefore, stabilizes numerical computations.

5.2 System Overview

5.2.1 Coordinates and Motion

The system is based on the camera coordinate system $OXYZ$ shown in Figure 5.1, in which the origin O is the center of projection, the Z axis coincides with the optical axis, and the image plane is located at $Z = 1$.

If the relative motion of the camera with respect to the scene is composed of a translation velocity (U, V, W) and a rotation velocity (A, B, C), we have the following relation between the flow velocity (u, v) and the depth Z of pixel location (x, y) from

[50]:

$$u = \frac{-U + xW}{Z} + Axy - B(x^2 + 1) + Cy, \tag{5.1}$$

$$v = \frac{-V + yW}{Z} - Bxy + A(y^2 + 1) - Cx. \tag{5.2}$$

If we designate the camera motion parameterization as $\vec{M}_0 = (U, V, W, A, B, C)^T$ and the flow velocity as $\vec{v} = (u, v)^T$, the above equation can be expressed as

$$\vec{v} = \vec{v}(x, y, Z, \vec{M}_0), \tag{5.3}$$

or its inverse

$$Z = Z(\vec{v}, x, y, \vec{M}_0). \tag{5.4}$$

5.2.2 Block Diagram of the System

Functionally, the system is decomposed into three major blocks in addition to optical flow computation as in Figure 5.2. For simplicity, we will refer to an optical flow and its uncertainty together as the optical flow information, the structure and its uncertainty together as the structural information, and the motion and its uncertainty together as the motion information. We also designate the camera coordinate system before current motion as *a priori* coordinate system and the camera coordinate system after current motion as *a posteriori* coordinate system. The computations within each block are as follows.

- *Initial Motion Estimate*: This block uses the current optical flow information and the predicted structural information to compute an initial estimate of motion information for the current frame. Since the motion can be discontinuous, the current motion is independent of the previous ones. Once the motion information is estimated, we can re-parameterize the motion parameters such that the optical flow is equally sensitive to all of the motion parameters.

- *EKF-based Update*: This block uses the current flow information, the predicted structure information and the initial motion information to compute *a posteriori* structural and motion information. The structure is represented with respect to the *a priori* coordinate system.

- *Interpolation and Transformation*: This block converts the structural information from the *a priori* coordinate system into the *a posteriori* coordinate system by interpolation, spatial rotations and translations.

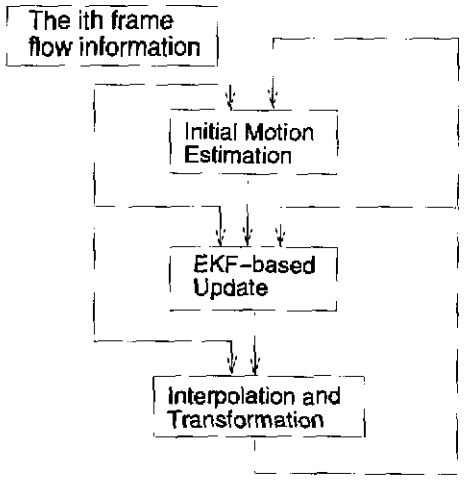


Figure 5.2: Block Diagram of the System

5.3 Initial Motion Estimation

Unlike many other approaches such as [7], we do not assume continuous motion. In other words, we assume that the motion at the current frame is totally unrelated to the motion in the previous frame because we believe that the continuous motion assumption is unrealistic in many cases such as navigating on real roads, hand-held video recording, and so on. Thus, in order to apply the EKF framework to our problem, we need to estimate an initial motion and its *a priori* covariance for each frame. Theoretically we do not need *a priori* covariance because it should be infinitely large. But in practice we need a finite one for numerical stability and re-parameterization.

First of all, given *a priori* structure Z_i and flow velocity $\vec{v}_i$ (i = 1, 2, ..., N), we can estimate the initial motion $\vec{M}_0$ from Eq. 5.3 by a linear least squares fitting, e.g., minimizing

$$\sum_{i=1}^{N} \left(\vec{v}_i - \vec{v}(x_i, y_i, Z_i, \vec{M}_0)\right)^T \mathbf{r}_i^{-1} \left(\vec{v}_i - \vec{v}(x_i, y_i, Z_i, \vec{M}_0)\right), \tag{5.5}$$

where $\mathbf{r}_i$ is the covariance of the flow velocity $\vec{v}_i$.

But we cannot use the above minimization to estimate the covariance of $\vec{M}_0$ because we did not consider the uncertainties of structure Z_i. Designating the *a priori* depth

as

$$\vec{Z}_a = \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_N \end{pmatrix}, \tag{5.6}$$

and the depth computed from current motion $\vec{M}_0$ and $\vec{v}$ by Eq. 5.4 as $\vec{Z}_c$, we have the following objective function to minimize:

$$\sum_{i=1}^{N} \left(\vec{v}_i - \vec{v}(x_i, y_i, Z_i, \vec{M}_0)\right)^T \mathbf{r}_i^{-1} \left(\vec{v}_i - \vec{v}(x_i, y_i, Z_i, \vec{M}_0)\right) + (\vec{Z}_a - \vec{Z}_c)^T (\mathbf{P}_s + \mathbf{C}_d)^{-1} (\vec{Z}_a - \vec{Z}_c), \tag{5.7}$$

where $\vec{Z}_a$ and $\vec{Z}_c$ are $N \times 1$ vectors, $\mathbf{P}_s$ is an $N \times N$ covariance matrix representing the uncertainty of $\vec{Z}_a$, and $\mathbf{C}_d$ is the depth uncertainty caused by the current flow uncertainty given $\vec{M}_0$. Because the flow uncertainties of different pixels are independent, $\mathbf{C}_d$ is an $N \times N$ diagonal matrix. Inverting and multiplying $(\mathbf{P}_s + \mathbf{C}_d)$ using $O(N)$ computation and memory will be explained in the next section.

If we ignore the dependence of $\mathbf{C}_d$ on $\vec{M}_0$, the minimization of Eq. 5.7 can be achieved using the Levenberg-Marquardt method [72]. In fact, this simplification is justifiable because $\mathbf{C}_d$ is usually insensitive to $\vec{M}_0$. Once the objective function is minimized, the curvature of the objective function at the minima can be used to estimate the covariance of $\vec{M}_0$.

5.3.1 Dynamic Motion Parameterization

Motion is traditionally parameterized using three translation parameters and three rotation parameters as in Eq. 5.1 and Eq. 5.2. As pointed out in [7], if the camera has a long focal length, the optical flow is much more sensitive to translations in the XY plane than to translations in the Z direction. Ideally we want the optical flow to be equally sensitive to all six motion parameters because otherwise the the covariance of motion $\mathbf{C}_m$ in Eq. 5.21 could be numerically near singular and therefore ruin the numerical computation of EKF.

We introduce the concept of "dynamic motion parameterization" to equalize sensitivities of motion parameters. There are two sources of sensitivity difference:

1. *Static Sensitivity Difference* is caused by the camera configuration. For example,

if the camera has a narrow field of view, the optical flow is usually much more sensitive to rotation than to translation.

2. *Dynamic Sensitivity Difference* is caused by the current flow or depth estimate instead of the camera. For example, if the optical flow has uncertainty much greater in one direction than in others, the optical flow is less sensitive to the motion which caused the optical flow in that direction.

If we designate the covariance of $\vec{M}_0$ computed from minimizing the objective function of Eq. 5.7 as $\mathbf{C}_{mt}$, we can normalize sensitivities by using a new set of motion parameters

$$\vec{M} = \mathbf{T}^{-1}\vec{M}_0, \tag{5.8}$$

where

$$\mathbf{C}_{mt} = \mathbf{T}\mathbf{T}^T. \tag{5.9}$$

It can be easily verified that the covariance of the new motion vector $\vec{M}$ is the unit matrix $\mathbf{I}$.

Note that the *a posteriori* uncertainty covariance of motion parameters $\vec{M}$ is the unit matrix because we have already used the optical flow information of the current frame. The only reason to compute motion covariance $\mathbf{C}_{mt}$ is to re-parameterize motion parameters. We will not use the covariance in the following EKF because EKF requires *a priori* motion covariance. Theoretically the *a priori* motion covariance should be infinitely large due to the fact that motions are uncorrelated. But very large covariance may cause numerical problems. Ideally, we want the *a priori* covariance to be small enough to avoid numerical problems, and yet large enough to not contain any information from the current frame. In practice, we use $1000\mathbf{I}$, where $\mathbf{I}$ is the unit matrix, as *a priori* motion covariance because it avoids the numerical problem of an infinitely large covariance and is also large enough (compared to *a posteriori* covariance $\mathbf{I}$) to be uninformative.

5.4 EKF-based Uncertainty Update

The extended Kalman filtering approach has been applied successfully in many fields to combine uncertainty information. In this section, we will briefly go over the general EKF framework as in [29] (Chapter 6), and then apply this framework to our nonlinear problem.

The measurements $\vec{z}$ and the state vector $\vec{x}$, which is what we need to estimate, are related according to:

$$\vec{z} = h(\vec{x}) + \vec{n}, \tag{5.10}$$

where $\vec{n}$ is a zero-mean measurement error whose covariance is $\mathbf{R}$. If the *a priori* estimates of the state vector and its covariance are $\vec{x}_-$ and $\mathbf{P}_-$ respectively, the *a posteriori* estimates of the state vector and its covariance after combining the new measurements are $\vec{x}_+$ and $\mathbf{P}_+$:

$$\vec{x}_+ = \vec{x}_- + \mathbf{K}(\vec{z} - h(\vec{x}_-)), \tag{5.11}$$

$$\mathbf{P}_+ = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}_-, \tag{5.12}$$

where $\mathbf{K}$ is the Kalman gain

$$\mathbf{K} = \mathbf{P}_-\mathbf{H}^T(\mathbf{H}\mathbf{P}_-\mathbf{H}^T + \mathbf{R})^{-1}, \tag{5.13}$$

and $\mathbf{H}$ is the Jacobian matrix of the measurement equations, i.e.,

$$\mathbf{H} = \frac{\partial h(\vec{x})}{\partial \vec{x}}|_{\vec{x}=\vec{x}_-}. \tag{5.14}$$

In the problem of dense structure from optical flows, the state vector is an $(N + 6)$ vector composed of six motion parameters $\vec{M}$ and depth at every pixel $Z_i, i = 1, 2, ..., N$, where N is the number of pixels. We express the state vector as

$$\vec{x} = (\vec{M}^T, Z_1, Z_2, ..., Z_N)^T. \tag{5.15}$$

Note that the motion parameters $\vec{M}$ have a linear relation with the original motion parameters $\vec{M}_0$ in Eq. 5.3 as shown in Eq. 5.8, i.e.,

$$\vec{M}_0 = \mathbf{T}\vec{M}, \tag{5.16}$$

where $\mathbf{T}$ is a 6×6 non-singular matrix.

The measurements are stored in a $2N$ vector which represents flow velocities at every pixel, i.e.,

$$\vec{z} = (\vec{v}_1^T, \vec{v}_2^T, ..., \vec{v}_N^T)^T, \tag{5.17}$$

where $\vec{v}_i$ $(i = 1, 2, ..., N)$ is the measured flow velocity (u_i, v_i) at each pixel. And the error covariance of the measurement vector $\vec{z}$ is a $2N \times 2N$ diagonal block matrix

$$\mathbf{R} = \begin{pmatrix} \mathbf{r}_1 & & & \\ & \mathbf{r}_2 & & \\ & & \ddots & \\ & & & \mathbf{r}_N \end{pmatrix}, \tag{5.18}$$

in which $\mathbf{r}_i$ $(i = 1, 2, ..., N)$ is the 2×2 error covariance matrix of the flow velocity at each pixel.

Using Eq. 5.3 and Eq. 5.16, the measurement equations can be expressed as

$$\vec{z} = \begin{pmatrix} \vec{v}(x_1, y_1, Z_1, \mathbf{T}\vec{M}) \\ \vec{v}(x_2, y_2, Z_2, \mathbf{T}\vec{M}) \\ \vdots \\ \vec{v}(x_N, y_N, Z_N, \mathbf{T}\vec{M}) \end{pmatrix} = \begin{pmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_N \end{pmatrix}, \tag{5.19}$$

in which x_i, y_i $(i = 1, 2, ..., N)$ and $\mathbf{T}$ are known. The Jacobian matrix of the measurement equations is

$$\mathbf{H} = \begin{pmatrix} \frac{\partial \vec{v}_1}{\partial \vec{M}} & \frac{\partial \vec{v}_1}{\partial Z_1} & & \\ \frac{\partial \vec{v}_2}{\partial \vec{M}} & & \frac{\partial \vec{v}_2}{\partial Z_2} & \\ \vdots & & & \ddots \\ \frac{\partial \vec{v}_N}{\partial \vec{M}} & & & \frac{\partial \vec{v}_N}{\partial Z_N} \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{S} \end{pmatrix}, \tag{5.20}$$

in which $\mathbf{A}$ is a $2N \times 6$ matrix and $\mathbf{S}$ is an $N \times N$ diagonal block matrix with each block a 2×1 matrix.

Now that we have formulated the problem of recovering dense structure from optical flows in the EKF framework, it seems that all we need to do is to plug those formulas into Eq. 5.11 and Eq. 5.12 so that the dense structure information can be recursively estimated. And that is exactly what people did in feature-based methods such as [16] and [7]. Unfortunately, if we apply this scheme directly to the dense structure recovery problem, the computation and memory requirements are insurmountable. As pointed out in [86], the uncertainties of the depth values $Z_i, i = 1, 2, ..., N$ are correlated due to uncertain motion. Thus the covariance matrix $\mathbf{P}$ is a full $N \times N$ matrix. And the computation of the Kalman gain in Eq. 5.13 which contains an inverse of a full $2N \times 2N$ matrix requires at least $O(N^2)$ computation and memory. Considering an ordinary 256×256 image, storing a symmetric $2N \times 2N$ (here $N = 256 \times 256 = 65,536$) matrix in single precision will require more than 30 gigabytes of memory. Even if we could represent such a matrix, it is impractical to invert it on ordinary workstations.

5.4.1 Decomposition of Independent and Correlated Uncertainty

Fortunately, we can take advantage of this specific problem to overcome these difficulties. As we mentioned before, the uncertainties of the depth values are correlated due to uncertain motion. But since there are only six motion parameters, the correlated uncertainty of the depth values caused by a single uncertain motion is an $N \times N$ matrix with rank of only six.

Because the rank of the correlated uncertainty is much smaller than N, the covariance matrix $\mathbf{P}$ can be decomposed into the following format:

$$\mathbf{P} = \begin{pmatrix} \mathbf{C}_m & \mathbf{C}_p^T \\ \mathbf{C}_p & \mathbf{P}_s \end{pmatrix} = \begin{pmatrix} \mathbf{C}_m & \mathbf{C}_p^T \\ \mathbf{C}_p & (\mathbf{C}_s + \mathbf{U}\mathbf{V}^T) \end{pmatrix}, \tag{5.21}$$

where $\mathbf{C}_m$ is a 6×6 matrix representing the covariance of the motion parameters, $\mathbf{C}_p$ is an $N \times 6$ matrix representing the correlation between the motion and the structure, $\mathbf{P}_s$ is an $N \times N$ matrix representing the correlation among depth values, which is, in turn, decomposed as a sum of two terms: $\mathbf{C}_s$ is an $N \times N$ diagonal matrix representing the *independent* uncertainty of the depth value of each pixel, and $\mathbf{U}$ and $\mathbf{V}$ are both $N \times k$ matrices whose outer product is a rank k matrix representing the *correlated* uncertainty of the depth values. Therefore, storing the matrix $\mathbf{P}$ sparsely will only require $O(N)$ memory if k is a constant.

Now that we can represent the covariance matrix $\mathbf{P}$, we will show that in the EKF framework, once the *a priori* uncertainty $\mathbf{P}_-$ can be represented in the format of Eq. 5.21, the computations involved in the EKF formulae (Eq. 5.11 and Eq. 5.12) is $O(N)$, and the *a posteriori* uncertainty $\mathbf{P}_+$ can also be represented in the same format.

The EKF-based uncertainty update explained above involves two types of matrix manipulations: inverse and multiplication. Our goal here is to achieve those matrix manipulations with $O(N)$ computation and memory. There are two mathematical tools that we can use to achieve that goal:

1. **Sherman-Morrison-Woodbury Inversion:** The inverse of a full rank matrix $\mathbf{C}$ perturbed by a rank k matrix $\mathbf{U}\mathbf{V}^T$ is sum of the inverse of $\mathbf{C}$ and another rank k matrix $\mathbf{U}_1\mathbf{V}_1^T$, i.e.,

$$(\mathbf{C} + \mathbf{U}\mathbf{V})^{-1} = \mathbf{C}^{-1} + \mathbf{U}_1\mathbf{V}_1^T, \tag{5.22}$$

where

$$\mathbf{U}_1 = -\mathbf{C}^{-1}\mathbf{U}(\mathbf{I}_k + \mathbf{V}^T\mathbf{C}^{-1}\mathbf{U})^{-1}, \quad (5.23)$$

$$\mathbf{V}_1 = \mathbf{C}^{-T}\mathbf{V}. \quad (5.24)$$

In our case, since $\mathbf{C}$ is a diagonal matrix, the above matrix inverse will need only $O(kN)$ computation and memory.

2. **Concatenation in multiplication:** The multiplication of two matrices, which are both in form of $\mathbf{P}_s$ in Eq. 5.21, can still be represented in the same form:

$$(\mathbf{C}_1 + \mathbf{U}_1\mathbf{V}_1^T)(\mathbf{C}_2 + \mathbf{U}_2\mathbf{V}_2^T) = \mathbf{C}_3 + \mathbf{U}_3\mathbf{V}_3^T, \quad (5.25)$$

where $\mathbf{C}_1$, $\mathbf{C}_2$ and $\mathbf{C}_3$ are $N \times N$ diagonal matrices, $\mathbf{U}_1$ and $\mathbf{V}_1$ are $N \times k$ matrices, $\mathbf{U}_2$ and $\mathbf{V}_2$ are $N \times k$ matrices, and $\mathbf{U}_3$ and $\mathbf{V}_3$ are $N \times 2k$ matrices:

$$\mathbf{C}_3 = \mathbf{C}_1\mathbf{C}_2 \quad (5.26)$$

$$\mathbf{U}_3 = \begin{pmatrix} \mathbf{C}_1\mathbf{U}_2 & \mathbf{U}_1 \end{pmatrix} \quad (5.27)$$

$$\mathbf{V}_3 = \begin{pmatrix} \mathbf{V}_2 & (\mathbf{C}_2^T + \mathbf{V}_2\mathbf{U}_2^T)\mathbf{V}_1 \end{pmatrix}. \quad (5.28)$$

The computation and memory requirement in the above matrix multiplication is $O(kN)$.

As shown in Appendix H, if the *a priori* covariance matrix $\mathbf{P}$ could be decomposed as Eq. 5.21, the *a posteriori* covariance matrix using Kalman filtering as in Eq. 5.12 is

$$\mathbf{P}_+ = \begin{pmatrix} \mathbf{C}_{mp} & \mathbf{C}_{pp}^T \\ \mathbf{C}_{pp} & (\mathbf{C}_4 + \mathbf{U}_4\mathbf{V}_4^T) \end{pmatrix}, \quad (5.29)$$

where $\mathbf{C}_4$ is an $N \times N$ diagonal matrix, and $\mathbf{U}_4$ and $\mathbf{V}_4$ are both $N \times (3k+6)$ matrices.

In other words, if we formulate the structure-from-motion problem as an recursive EKF estimation, the covariance matrix $\mathbf{P}$ can always be represented sparsely as in Eq. 5.21 throughout the processing of the whole optical flow sequence because $\mathbf{P}$, $\mathbf{R}$ and $\mathbf{H}$ are all special matrices. We never need to explicitly represent $\mathbf{P}$ as an $(N+6) \times (N+6)$ matrix.

If we are careful about the ordering of matrix multiplications, we then have an algorithm which updates the state vector and its covariance using $O(kN)$ computation and memory for every frame. Unfortunately, k increases linearly after each frame, which makes the above algorithm $O(MN)$ for the Mth frame. Though M is usually much smaller than the number of pixels N, it is still impractical for long image sequences. In the next section, we introduce weighted principal component analysis to keep k constant, and therefore achieve an $O(N)$ algorithm.

5.4.2 Weighted Principal Component Analysis

First of all, let us consider the eigenvalues and eigenvectors of the correlated uncertainty matrix $\mathbf{U}_4\mathbf{V}_4^T$. In general, there are $l = 3k + 6$ non-zero eigenvalues and corresponding eigenvectors, which can be computed easily as in Appendix G. Because the outer product represents a covariance which must be symmetric, it can be expressed as

$$\mathbf{U}_4\mathbf{V}_4^T = \begin{pmatrix} & & & \\ \vec{e}_1 & \vec{e}_2 & \cdots & \vec{e}_l \\ & & & \end{pmatrix} \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_l \end{pmatrix} \begin{pmatrix} & \vec{e}_1^T & \\ & \vec{e}_2^T & \\ & \vdots & \\ & \vec{e}_l^T & \end{pmatrix}, \tag{5.30}$$

where $\vec{e}_i, i = 1, 2, \ldots, l$ are $N \times 1$ eigenvectors, and $\lambda_i, i = 1, 2, \ldots, l$ are the corresponding eigenvalues ordered by magnitude such that λ_1 is the largest eigenvalue.

Every eigenvector is an $N \times 1$ vector, which represents an eigenimage. This eigenimage illustrates the *pattern* of the depth uncertainty, and the corresponding eigenvalue represents the *magnitude* of this depth uncertainty. For example, if the eigenimage is an image with same value at every pixel, the depth uncertainty represented by this eigenimage is that depth values of all pixels have to change simultaneously by the same amount. In other words, changes of depth values allowed by this eigenimage have to be in the pattern specified by the eigenimage:

$$\delta \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_N \end{pmatrix} = c\vec{e}, \tag{5.31}$$

where c is a scalar constant and $\vec{e}$ is the eigenimage. The meaning of the eigenvalue is similar to that of σ in a Gaussian distribution, which represents the magnitude of the uncertainty.

Since the eigenvalues in Eq. 5.30 are in descending order, the naive approach to keep k constant is to truncate the eigenvalues after first k largest ones, i.e.,

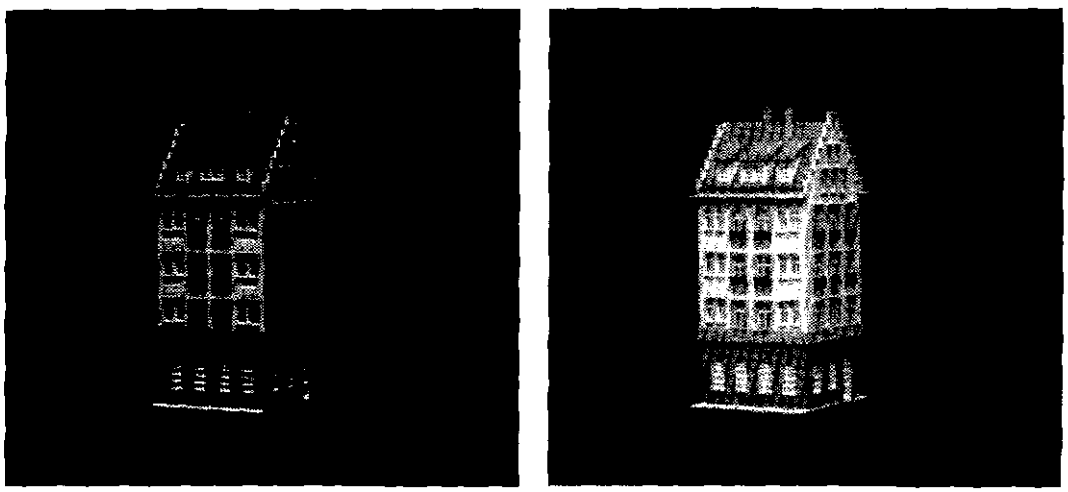
$$\mathbf{U}_4\mathbf{V}_4^T \approx \begin{pmatrix} & & & \\ \vec{e}_1 & \vec{e}_2 & \cdots & \vec{e}_k \\ & & & \end{pmatrix} \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_k \end{pmatrix} \begin{pmatrix} \vec{e}_1^T & & \\ \vec{e}_2^T & & \\ \vdots & & \\ \vec{e}_k^T & & \end{pmatrix}. \tag{5.32}$$

In other words, we throw away those insignificant dimensions. Thus $\mathbf{U}_4$ and $\mathbf{V}_4$ can both be reduced to $N \times k$ matrices. The iterations of EKF updating illustrated in the previous section can be carried out in $O(N)$ for every frame no matter how long the sequence is.

The underlying assumption of truncating small eigenvalues in Eq. 5.32 is that the uncertainty information implied by those eigenvalues/eigenimages is negligible compared to the independent uncertainty $\mathbf{C}_4$. And the reason for keeping large eigenvalues is that we assume the uncertainties implied by these large eigenvalues and their corresponding eigenvectors are at least comparable to the independent uncertainty $\mathbf{C}_4$. But since the independent uncertainties of pixels are not uniform, truncating by the magnitudes of eigenvalues may not make much sense at all because even though a relatively large eigenvalue may imply a large uncertainty in a certain area in the eigenimage, if the independent uncertainty happens to be even larger in the same area, this eigenvalue/eigenimage becomes less significant.

Based on the above speculation, we propose a weighted principal component analysis, i.e., the correlated uncertainty is weighted by independent uncertainty before decomposition as in Eq. 5.30. Since the independent uncertainty $\mathbf{C}_4$ is a diagonal matrix, and its diagonal elements have to be positive, we can decompose it as

$$\mathbf{C}_4 = \begin{pmatrix} c_1 & & & \\ & c_2 & & \\ & & \ddots & \\ & & & c_N \end{pmatrix}$$



Original Image Gamma = 4.0

Figure 5.3: An Image Sequence of a Toy House

$$= \begin{pmatrix} \sqrt{c_1} & & & \\ & \sqrt{c_2} & & \\ & & \ddots & \\ & & & \sqrt{c_N} \end{pmatrix} \begin{pmatrix} \sqrt{c_1} & & & \\ & \sqrt{c_2} & & \\ & & \ddots & \\ & & & \sqrt{c_N} \end{pmatrix} = \mathbf{Q}\mathbf{Q}^T. \quad (5.33)$$

Therefore, the overall uncertainty can be represented as

$$\mathbf{C}_4 + \mathbf{U}_4\mathbf{V}_4^T = \mathbf{Q}\left(\mathbf{I} + \mathbf{Q}^{-1}\mathbf{U}_4(\mathbf{Q}^{-1}\mathbf{V}_4)^T\right)\mathbf{Q}^T. \quad (5.34)$$

In other words, we have weighted the correlated uncertainty $\mathbf{U}_4$ and $\mathbf{V}_4$ by the independent uncertainty $\mathbf{Q}^{-1}$. We then truncate small eigenvalues of $\mathbf{Q}^{-1}\mathbf{U}_4(\mathbf{Q}^{-1}\mathbf{V}_4)^T$.

Figure 5.3 (Original Image) shows one frame of a toy house sequence. The uncertainties of flow velocity computed from two adjacent frames are very large in the dark background area compared to those in the house area. There are small intensity variations in the background, which are visible after Gamma correction as in Figure 5.3. Figure 5.4 shows the six most significant eigenvalues/eigenimages of the correlated structure uncertainty using the direct decomposition method from Eq. 5.32 after processing the first frame of optical flow. The eigenimages are shown by linearly quantizing 0 to grey-level 125, -0.025 or smaller values to grey-level 0 and 0.025 or larger values to grey-level 255. As indicated by either high or low grey-levels, the uncertainty information captured in the 2nd, 3rd, 4th, 5th and 6th eigenimages/eigenvalues is mainly in the background area. On the other hand, Figure 5.5

shows the first six weighted principal components. Obviously, the weighted principal components carry much more *useful* structural uncertainty information than the naive approach.

Since eigenimages represent orthogonal patterns of possible change or deformation of the depth map in Eq. 5.31, they also lend themselves for intuitive interpretations. For example, across the house in the second eigenimage in Figure 5.5, there is one skew line of grey-level 125, whose left side is bright and right side is dark. Referring to Eq. 5.31, we can see that this pattern represents a possible rotation around the skew line in the depth map, i.e., the depth values on the left side increase while the ones on the right side decrease or vice versa. Other eigenimages can be similarly interpreted although the patterns may be more complicated. In the context of structure from motion, we believe that the intrinsic ambiguities [4] such as camera translation versus rotation are represented and carried through recursive estimations by those uncertainty patterns so that they could be resolved when new information becomes available.

5.5 Interpolation and Forward Transformation

We represent the 3D shape by a depth map in the current camera coordinate system as in Figure 5.1. In the prediction phase of EKF, we need to transform the current depth map into next camera coordinate system and resample the depth map according to next sensor grid. This transformation and resample present two new problems:

- The interpolation of correlated uncertainty.
- The fact that the 3D translation and rotation between the current and the next coordinate systems are correlated to the 3D depth maps makes it difficult to transform the uncertainty information into the new coordinate system. Most existing recursive structure-from-motion systems ignore the fact that motion and structure are actually correlated as C_{pp} in Eq. 5.29 when they rotate or translate the structure according to the motion. Although the exact effects of this simplification are still unknown, our system will incorporate the information about the correlation between motion and structure in the prediction phase in EKF.

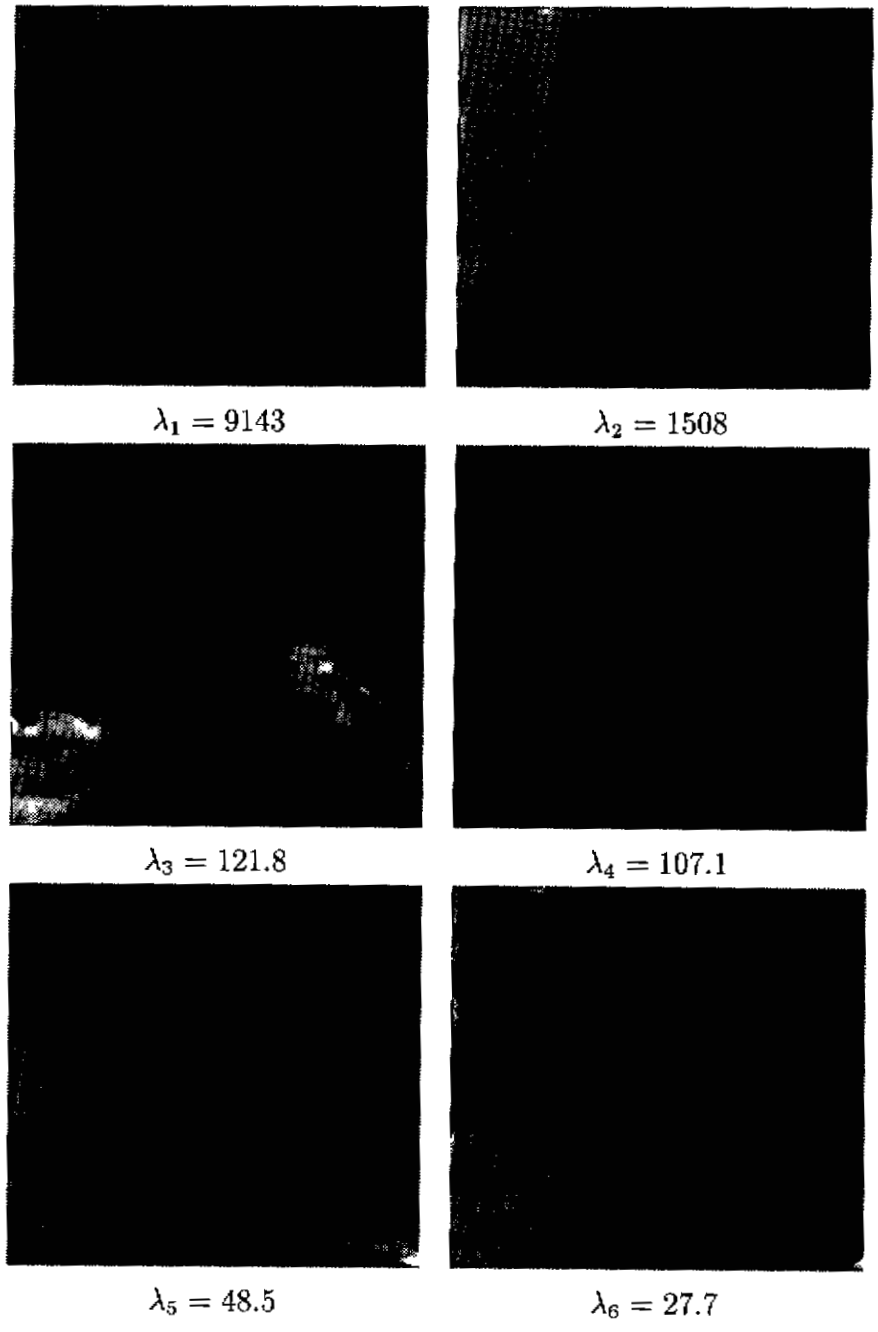
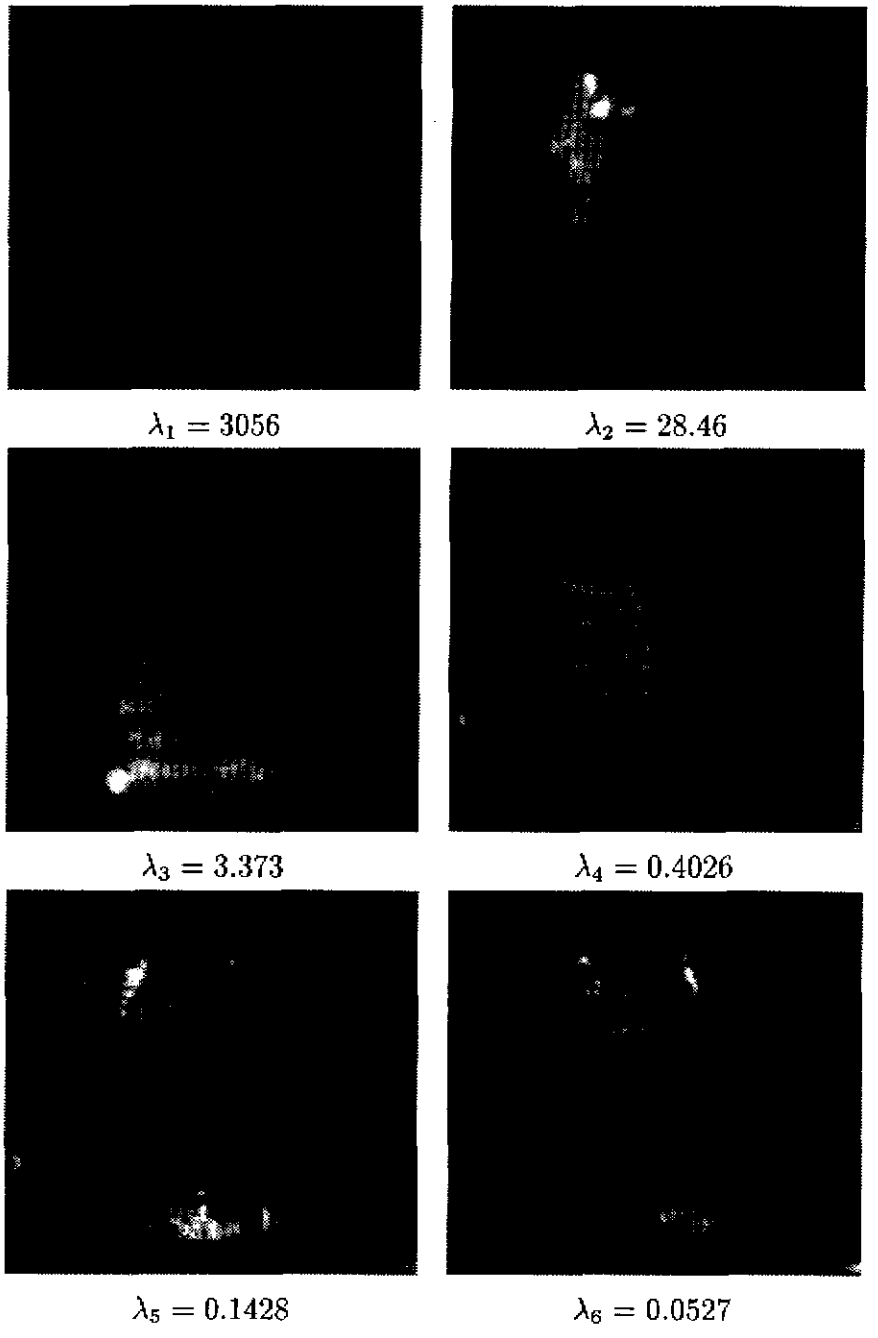


Figure 5.4: Six Non-Weighted Principal Components



$\lambda_1 = 3056$ $\lambda_2 = 28.46$

$\lambda_3 = 3.373$ $\lambda_4 = 0.4026$

$\lambda_5 = 0.1428$ $\lambda_6 = 0.0527$

Figure 5.5: Six Weighted Principal Components

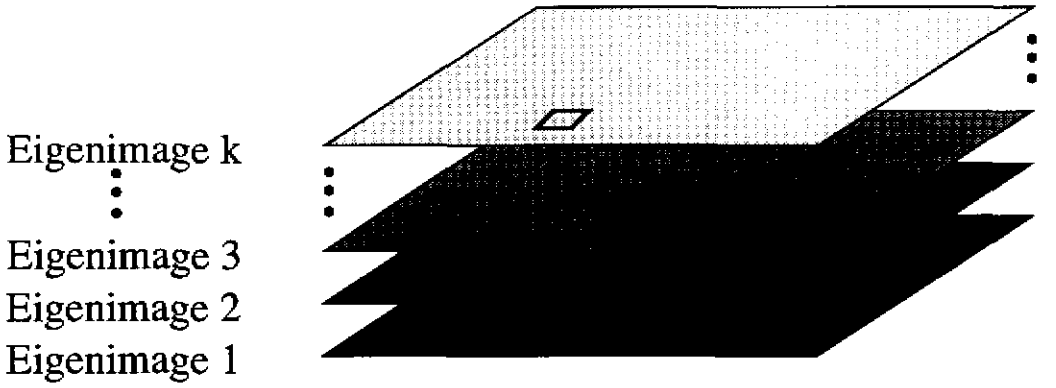


Figure 5.6: Vector Field as a Correlated Uncertainty

As explained in Appendix G, since a correlated uncertainty is always a positive definite symmetric outer product, it can be represented as

$$\mathbf{U}\mathbf{V}^T = \mathbf{B}\mathbf{B}^T, \tag{5.35}$$

where $\mathbf{B}$ is an $N \times k$ matrix just like $\mathbf{U}$. In other words, every row of $\mathbf{B}$ is a vector of length k that can be regarded as an attribute of the corresponding pixel. Therefore we can represent the correlated uncertainty as a vector field as in Figure 5.6, where the value at every pixel location is a vector, i.e., a stack of scalars. Furthermore, the correlated uncertainty between any two locations is the inner product of the vectors at the two locations. Interpolating the correlated uncertainty is done by interpolating this vector field.

Since the optical flow establishes the correspondence between two adjacent frames, we can interpolate, resample and transform the depth map represented in the current camera coordinate such that we have the depth and uncertainty information for grid positions in the next frame. We designate this process as the "forward transform". For every pixel position in the next frame, suppose its correspondence in the current frame is at location (x_i, y_i) in the image plane, and has depth Z_i in the current camera coordinate; we can compute the depth Z_i^+ in the next camera coordinate using the motion parameters, e.g.,

$$Z_i^+ = f(x_i, y_i, Z_i, \vec{M}_0) = f(x_i, y_i, Z_i, \mathbf{T}\vec{M}), \tag{5.36}$$

where f represents 3D rotation and translation function. The 3D transformation matrix can be found in [18] (page 52).

To compute the structural uncertainty in the new camera coordinate, we take the derivative of Eq. 5.36

$$\begin{aligned} dZ_i^+ &= \frac{\partial f}{\partial Z_i} dZ_i + \frac{\partial f}{\partial \vec{M}} \cdot d\vec{M} \\ &= a_i dZ_i + \vec{b}_i^T d\vec{M}, \end{aligned} \tag{5.37}$$

where $\vec{b}_i$ is a vector of length six. Thus we have the covariance between two arbitrary points as

$$\begin{aligned} E[dZ_i^+ dZ_j^+] &= a_i a_j E[dZ_i dZ_j] + a_i \vec{b}_j^T E[dZ_i d\vec{M}] + \\ & \quad a_j \vec{b}_i^T E[dZ_j d\vec{M}] + \vec{b}_i^T E[d\vec{M} d\vec{M}^T] \vec{b}_j. \end{aligned} \tag{5.38}$$

If we rewrite the matrices in Eq. 5.29 as

$$\mathbf{C}_{mp} = E[d\vec{M} d\vec{M}^T], \tag{5.39}$$

$$\mathbf{C}_{pp} = \begin{pmatrix} E[dZ_1 d\vec{M}]^T \\ E[dZ_2 d\vec{M}]^T \\ \vdots \\ E[dZ_N d\vec{M}]^T \end{pmatrix} = \begin{pmatrix} \vec{p}_1^T \\ \vec{p}_2^T \\ \vdots \\ \vec{p}_N^T \end{pmatrix}, \tag{5.40}$$

$$\mathbf{U}_4 = \begin{pmatrix} \vec{U}_1^T \\ \vec{U}_2^T \\ \vdots \\ \vec{U}_N^T \end{pmatrix}, \tag{5.41}$$

$$\mathbf{V}_4 = \begin{pmatrix} \vec{V}_1^T \\ \vec{V}_2^T \\ \vdots \\ \vec{V}_N^T \end{pmatrix}, \tag{5.42}$$

where $\mathbf{C}_{mp}$ is a 6×6 matrix, and $\vec{p}_i$'s, $\vec{U}_i$'s and $\vec{V}_i$'s are vectors, we have the structural uncertainty after the forward transform as

$$E\left[\begin{pmatrix} dZ_1^+ \\ dZ_2^+ \\ \vdots \\ dZ_N^+ \end{pmatrix} \begin{pmatrix} dZ_1^+ \\ dZ_2^+ \\ \vdots \\ dZ_N^+ \end{pmatrix}^T \right] = \mathbf{C}^+ + \mathbf{U}^+ (\mathbf{V}^+)^T, \tag{5.43}$$

where

$$\mathbf{C}^+ = \begin{pmatrix} a_1^2 & & & \\ & a_2^2 & & \\ & & \cdots & \\ & & & a_N^2 \end{pmatrix} \mathbf{C}_4, \tag{5.44}$$

$$\mathbf{U}^+ = \begin{pmatrix} a_1\vec{U}_1^T & a_1\vec{p}_1^T & \vec{b}_1^T & \mathbf{C}_{mp}\vec{b}_1^T \\ a_2\vec{U}_2^T & a_2\vec{p}_2^T & \vec{b}_2^T & \mathbf{C}_{mp}\vec{b}_2^T \\ \vdots & \vdots & \vdots & \vdots \\ a_N\vec{U}_N^T & a_N\vec{p}_N^T & \vec{b}_N^T & \mathbf{C}_{mp}\vec{b}_N^T \end{pmatrix}, \tag{5.45}$$

$$\mathbf{V}^+ = \begin{pmatrix} a_1\vec{V}_1^T & \vec{b}_1^T & a_1\vec{p}_1^T & \vec{b}_1^T \\ a_2\vec{V}_2^T & \vec{b}_2^T & a_2\vec{p}_2^T & \vec{b}_2^T \\ \vdots & \vdots & \vdots & \vdots \\ a_N\vec{V}_N^T & \vec{b}_N^T & a_N\vec{p}_N^T & \vec{b}_N^T \end{pmatrix}. \tag{5.46}$$

Since $\mathbf{U}^+$ and $\mathbf{V}^+$ are now $N \times (k+18)$ matrices if $\mathbf{U}_4$ and $\mathbf{V}_4$ were $N \times k$ matrices, they may also be reduced to $N \times k$ by weighted principal component analysis.

5.6 Implementation Issues and Experiments

5.6.1 Implementation Issues

We implemented our system using single precision matrices. As always in manipulating large matrices, the numerical stability has to be carefully watched while carrying out those computations. The following are potential sources of unstable computations:

1. *Matrix Multiplications*: When computing the inner product of two large matrices $\mathbf{V}^T\mathbf{U}$, where both $\mathbf{U}$ and $\mathbf{V}$ are $N \times k$ ($k << N$), we have to carry those additions in double precision due to large N. For example, if the image is 256×256, we can easily lose four significant digits during multiplications if our intermediate computations are carried out in single precision.

2. *Ill-conditioned Matrices*: There is always a danger when one of the matrices in the computation is singular or near singular. In the worst case, we may lose all significant digits. Thus it is extremely helpful to avoid any ill-conditioned matrix. In our system, we pay special attention to the following matrices:

- Flow Uncertainty: The estimated optical flow uncertainty $\mathbf{r}_i$'s in Eq. 5.18.
- Motion Uncertainty: We used dynamic motion parameterization to prevent the motion covariance matrix $\mathbf{C}_m$ in Eq. 5.21 from being ill-conditioned.
- Kalman Gain: In computing the Kalman gain as in Eq. 5.13, it is numerically devastating if $\mathbf{HP_H}^T + \mathbf{R}$ is ill-conditioned. $\mathbf{HP_H}^T$ represents the projection of the uncertainty of motion and structure into the uncertainty of optical flow. In order for $\mathbf{HP_H}^T + \mathbf{R}$ to be well conditioned, we need to make sure that the projected uncertainty of optical flow is not significantly larger ($> 10^5$) than the estimated uncertainty $\mathbf{R}$. That is the reason we choose $1000\mathbf{I}$ as the *a priori* motion uncertainty.

3. *Sherman-Morrison-Woodbury Inversion:* Although inverting a structural uncertainty matrix according to the Sherman-Morrison-Woodbury formula significantly reduces the amount of computation and memory requirements compared to the traditional Gaussian elimination ([72]), the inversion has the disadvantage of being more fragile numerically ([33, 9], Appendix F).

Another common problem for a structure from motion system is the handling of the disappearance and reappearance of parts of the scene due to relative movement between the camera and the scene. Our system had no problem dealing with new parts, which were simply assigned a preset large independent uncertainty and a zero correlated uncertainty. But the depth information of disappearing parts was discarded. In the future, we would like to maintain a global shape module such that the structure information of disappeared parts could be stored and retrieved.

5.6.2 Experiments

We tested our system on real image sequences taken by a Sony XC-75 video camera. The relative camera movements in all the sequences involved both rotations and translations. We digitized images in two ways. One was to digitize by Matrox board while shooting the sequence. In order to digitize while taking images, we mounted the camera on a computer-controlled 6 DOF platform in Calibrated Imaging Lab, and stopped for every frame. Another way was to record the sequence on Umatic SP video tape, and digitized the tape frame by frame. Unfortunately, the digitizing device we had could digitize only one of two fields in every frame, and the videotape also introduced additional noise in the images. We will demonstrate the performance

of our system on image sequences digitized both ways. All images in our experiments are 480×512. The CPU time of processing every optical flow frame is about two minutes on Sparc 10.

Ambiguities

It is well known that there are intrinsic ambiguities in recovering structure from motion. There are three kinds of ambiguities:

1. *Scale Ambiguity*: The scale of the object or the absolute depth of the object can never be recovered. In motion estimation, the magnitude of translation cannot be determined.

2. *Rotation/Translation Ambiguity*: If the camera has a small field of view, the optical flow caused by a small camera rotation is very similar to that caused by a small camera translation.

3. *Other Motion Ambiguity*: Since the optical flow is always uncertain, we will always have uncertainty in estimating other motion parameters such as rotation and translation around z axis although they are usually less significant.

For every motion ambiguity, there is a corresponding structure ambiguity, since motion and structure are coupled by Eq. 5.3. In other words, all three kinds of ambiguities exist in both structure and motion estimations. We also note that there is no fundamental difference in terms of origination between the second and the third kinds of ambiguities other than the fact that, for an ordinary camera, the magnitude of the second kind of ambiguities is usually much larger than that of the third kind. Historically the second kind of ambiguity was frequently singled out in the literature.

In our system, we assign an initial depth and uncertainty to the first frame. Practically we assign a flat depth map and uniform independent uncertainty as *a priori* depth information. It serves two purposes, which are: disambiguation of the scale ambiguity by providing absolute depth; and allowing deformation of the *a priori* depth map to the true depth map by providing large independent uncertainty.

Our system keeps six principal eigenimages to represent the correlated uncertainty as shown in Figure 5.5. Among these eigenimages, the first one usually represents the scale ambiguity. The second and the third ones usually represent the rotation/translation ambiguity in two orthogonal directions. And the remaining eigenimages represent other minor ambiguities.

Conceptually, the independent uncertainty represents a *chaotic* uncertainty pattern, while the correlated uncertainty represents an *organized* uncertainty pattern. For example, if the eigenimage of a correlated uncertainty is uniformly bright, it denotes that the corresponding depth map can move back and forth. In other words, the depth values of all pixels have to change uniformly while the shape does not change at all. Since we set *a priori* depth uncertainty as totally chaotic, we will expect that, as more optical flow information is incorporated, the uncertainty will become less and less chaotic, more and more organized. In our framework, that means that the magnitude of the independent uncertainty will decrease while the magnitude of the correlated uncertainty can increase or decrease.

Optical flow information alone does not provide any information which we could use to decrease the scale ambiguity. Therefore, we expect that the eigenimage representing the scale ambiguity in correlated uncertainty will have larger and larger eigenvalue. Note that the eigenvalue is that of the *weighted* correlated uncertainty. The absolute magnitude of the scale ambiguity is constant, but since the magnitude of independent uncertainty monotonically decreases, the eigenvalue representing the scale ambiguity increases monotonically. On the other hand, the second and third kinds of ambiguities are strong in some optical flows while weak in other ones. Thus the eigenimages representing these ambiguities can have fluctuating eigenvalues, depending on the specific optical flow.

Figure 5.7 shows one frame in a fifty-frame sequence. The motions of the camera with respect to the straw hat involve translations in X, Y, and Z directions and rotations around X and Y axes from the 1st frame to the 30th frame. From the 30th frame to the 40th frame, the motions are translations in the Y direction and small rotations around the X axis[1]. From the 40th frame to the 50th frame, the motions are translations in the X direction and small rotations around the Y axis. Figure 5.8 shows the first three eigenimages, and Figure 5.9 shows the evolutions of average independent uncertainty and the eigenvalues corresponding to the three eigenimages. Note that the eigenimages change from frame to frame. In the examples shown here, the eigenimages did not change dramatically over the whole sequence, thus providing a good example to demonstrate the evolution of correlated uncertainty.

First of all, the fact that the average independent uncertainty decreases monotonically and the first eigenvalue which represents the scale ambiguity increases monoton-

[1]X is the column direction, and Y is the row direction

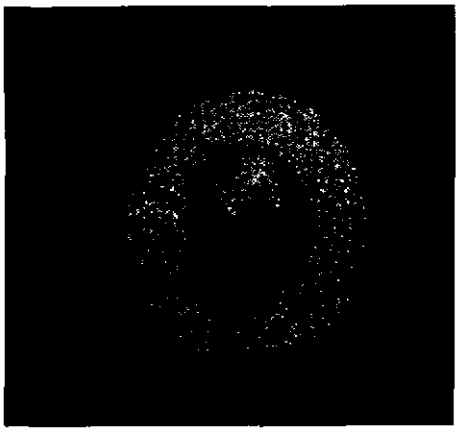


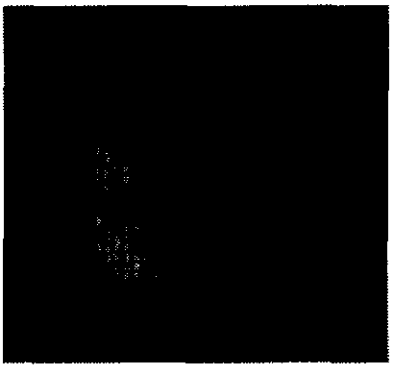
Figure 5.7: One Frame in a Straw Hat Sequence

ically indicates a steady improvement from a chaotic pattern to an organized pattern. Secondly, in the interval between the 30th and the 40th frame, there is an accumulating ambiguity of translation in Y direction versus rotation around X axis. This motion ambiguity maps into structural uncertainty as generally increasing third eigenvalues. And because there is no ambiguity of translation in the X direction versus rotation around the Y axis, the second eigenvalue decreases in the same time. For the same reason, in the interval between the 40th and the 50th frame, the second eigenvalue increases, at the same time the third eigenvalue decreases for the exactly opposite reason.

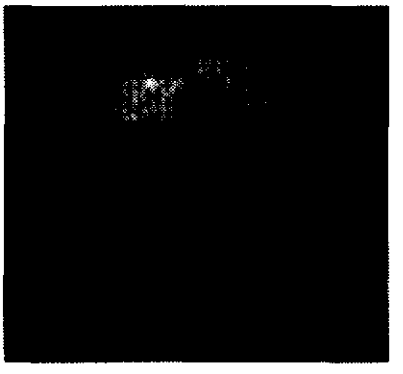
If the underlying camera motions or the optical flows of the whole sequence tend to be rather homogeneous, the system may never be able to resolve one or more ambiguities intrinsic to this type of optical flows. In this situation, the correlated uncertainty eigenimages representing the second kinds of ambiguity may have an ever increasing eigenvalues, which represent lack of information to disambiguate. If we have active control over the camera, the eigenimages could then be used to plan the camera motion in order to resolve the ambiguities.

Figure 5.10 (1) shows one frame of a road sequence which was shot using a camera fixed on a moving vehicle. The motions of the camera with respect to the scene were very homogeneous. In fact all the optical flows are similar to the one shown in Figure 5.10 (2). Figure 5.11 shows the first three eigenimages representing the correlated uncertainty. Figure 5.12 shows the average independent uncertainty and the eigenvalues corresponding to the three eigenimages. In this example, we can see that the ambiguity represented by the second eigenimage was never resolved. Comparing

First



Second



Third

Figure 5.8: Three Eigenimages for the Straw Hat Sequence

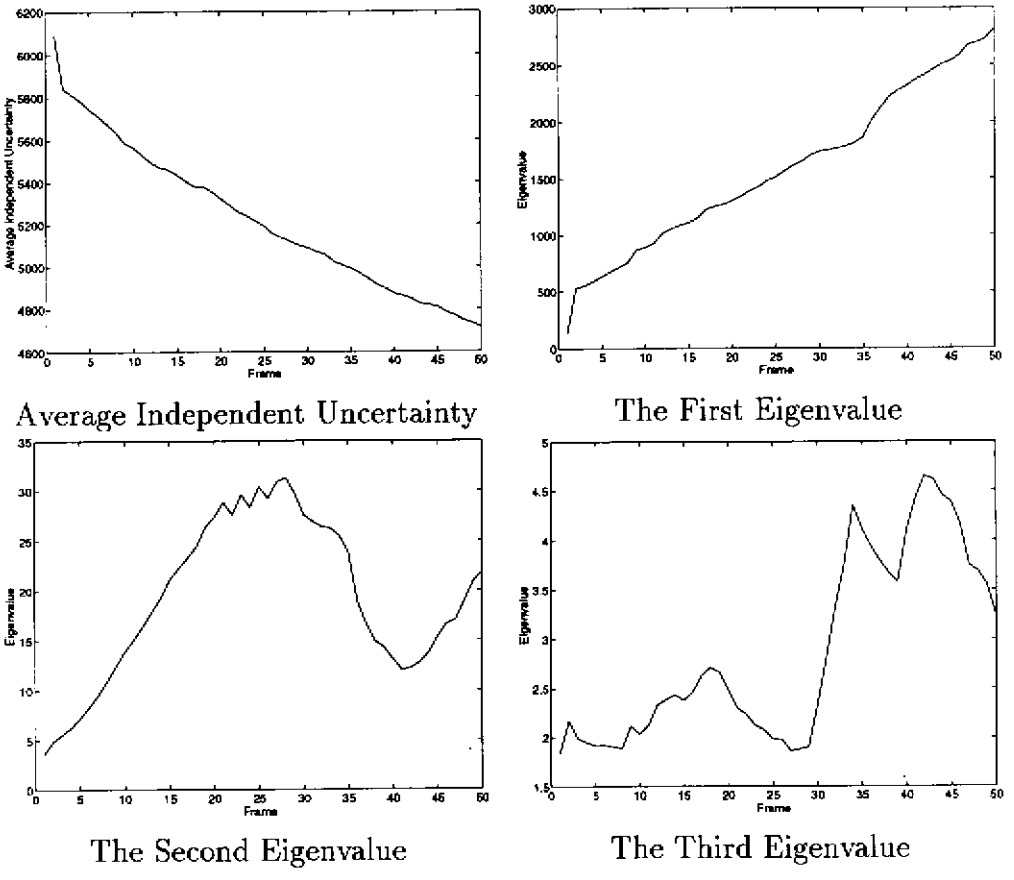
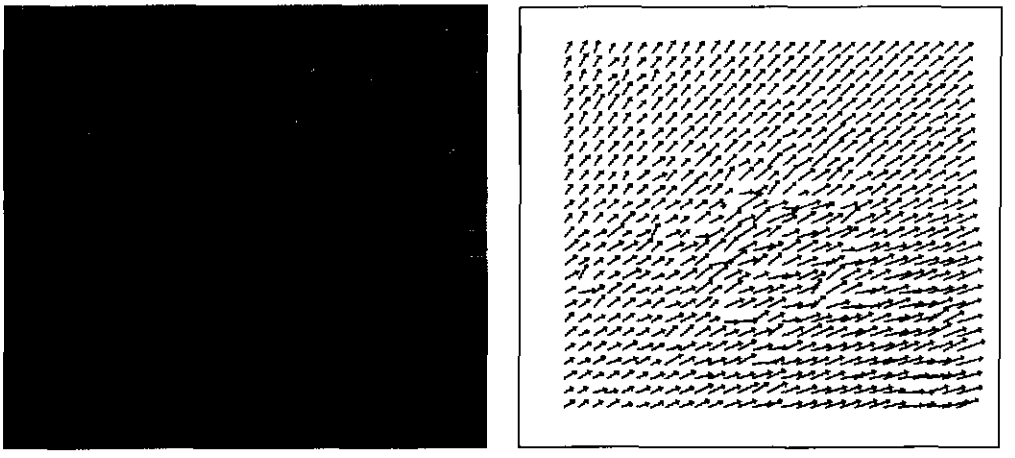


Figure 5.9: Evolutions of Straw Hat Uncertainties



(1) One Image Frame (2) One Optical Flow

Figure 5.10: A Road Sequence

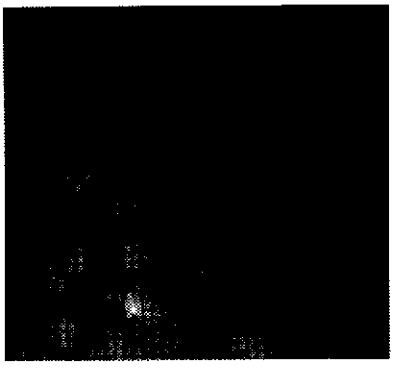
the optical flow in Figure 5.10, it is obvious that these optical flows provided little information to resolve the second kind of ambiguity in the flow direction, i.e., the direction from bottom-left to top-right, while they did provide enough information to resolve the second kind of ambiguity in the perpendicular direction. Secondly, unlike the previous example, the magnitude of the average independent uncertainty and the first eigenvalue were approaching constants. In other words, after about 20 frames, the improvement from the chaotic uncertainty pattern to an organized uncertainty pattern seems to have stopped. The reason is that, in this example, there is continuous appearing of new parts which were assigned chaotic uncertainties and disappearing of old parts whose uncertainties were organized. Therefore, after a certain number of frames, the improvement from chaotic to organized uncertainty pattern obtained in each new frame was totally cancelled by the introduction of new parts and by the loss of old parts.

Experiments on Real Sequences

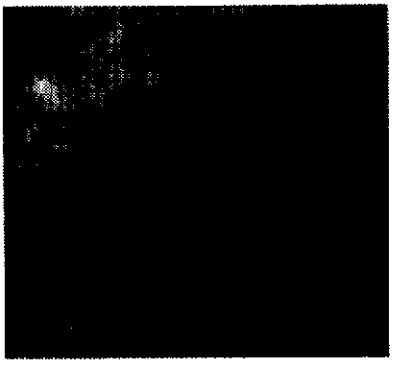
We tested our system on many image sequences with different signal noise ratios and different fields of view. No pre-processing was done on image sequences. Once we obtained a depth map sequence as the output from our system, we masked out background areas since the depth information in these areas is arbitrary. The separation of foreground and background was done by a simple thresholding and hole-filling.

The first sequence includes fifty-one-frame images of a straw hat as we showed in

First

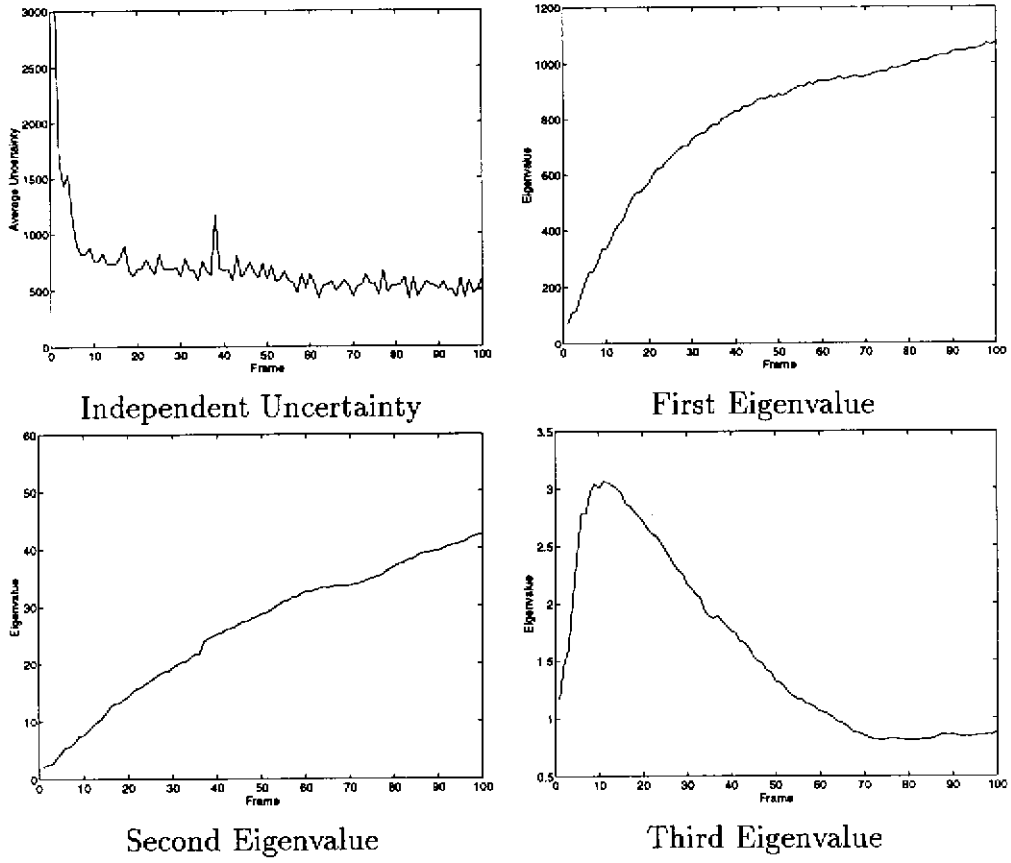


Second



Third

Figure 5.11: The First Three Eigenimages of the Road Sequence



Independent Uncertainty First Eigenvalue

Second Eigenvalue Third Eigenvalue

Figure 5.12: Evolutions of Road Uncertainties

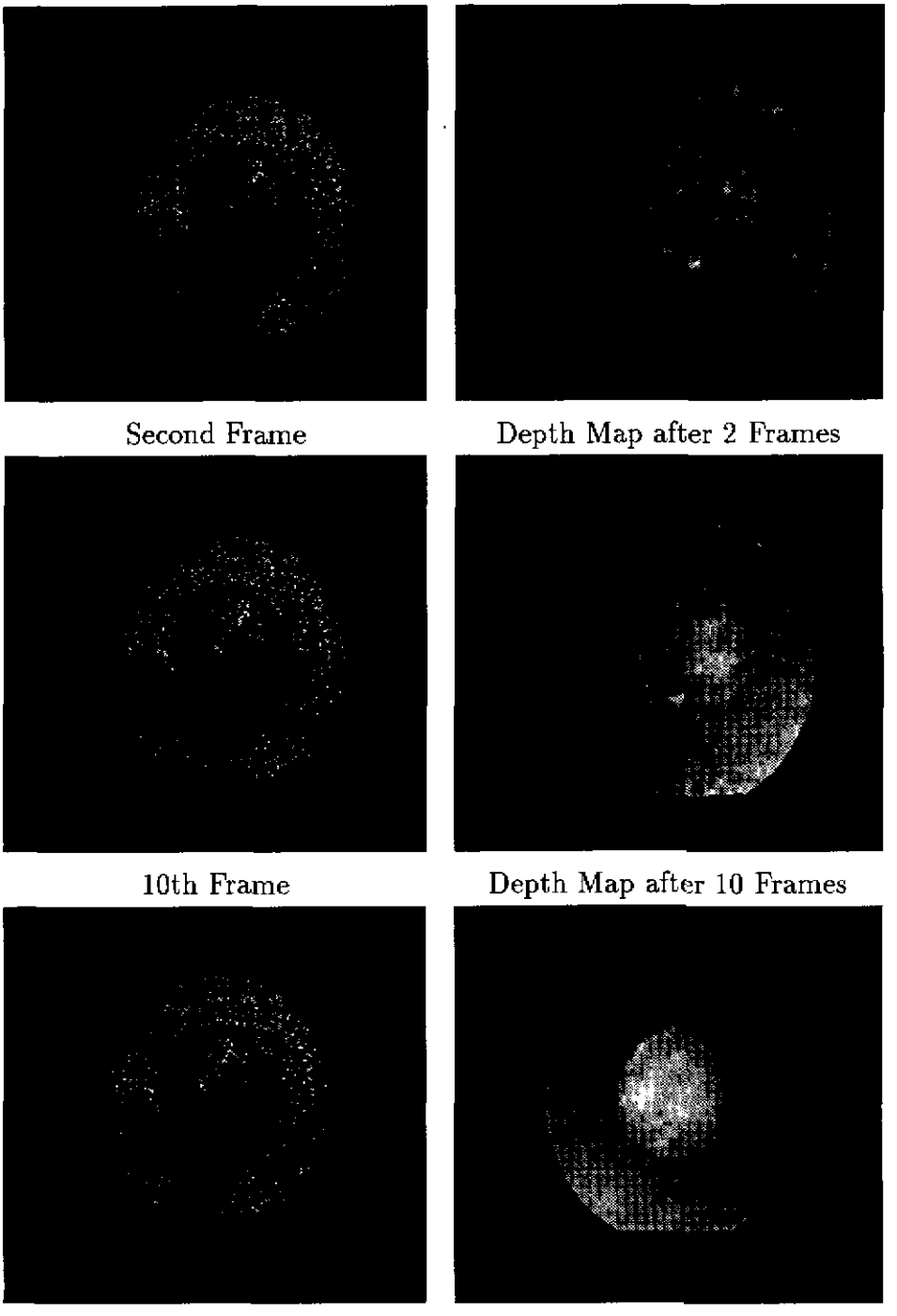
the previous section The images were digitized by a Matrox board. The rotations and translations of the camera with respect to the straw hat were discontinuous. The camera had about an 11° field of view. The optical flow and its uncertainty were computed using hyper-geometric filters [100, 104]. Figure 5.13 shows the intensity images and depth maps computed after the corresponding frames. It clearly shows the converging shape resulting from recursively combining information from multiple frames.

In order to measure how good the recovered shape is, we manually measured the shape of the straw hat. The profile of the straw hat is sketched in Figure 5.14. Figure 5.15 shows the evolution of the estimated shape of the straw hat over the whole sequence. Obviously, over the sequence, the noises in estimated shape are reduced as more images are used. We attribute this elegant convergence to the EKF framework which recursively improves the quality of estimations. Another observation is that after the second and tenth frames, the shapes are somehow skewed compared to the true shape. Since the rotations involved in the camera motions were never large enough to make the shape so obviously skewed with respect to the camera coordinate, we believe that the shape distortions were caused by the ambiguity between rotation and translation. In other words, due to lack of information and presence of noise in optical flows, the system incorrectly attributed part of image motion to be generated by rotation when in fact was caused by a translation. Since the translation/rotation ambiguities were carried through eigenimages/eigenvalues, those ambiguities were gradually resolved as more images were used.

Figure 5.16 shows the recovered shape after fifty frames, together with the true shape sketch. Although it is impossible to characterize the quality of results with absolute certainty because we do not have the exact ground truth, it is still safe to say that, for this particular example, the relative depth error is below 1/200. On the other hand, we found from experiments that this traditional way of measuring quality by comparing the ground truth and the estimations pixel by pixel not fair to those sequences which did not provide enough information to resolve all global ambiguities. In those situations, the errors at different pixels are actually correlated with each other. And this kind of correlations represents the *global* shape ambiguity. One possible way to get rid of this ambiguity is to provide a better initial shape. Further work needs to be done to provide a measure of quality of the recovered shape.

We also tested our system on many other real image sequences including:

- *The Chair Sequence*: The camera had approximately a 22° field of view. The



Second Frame | Depth Map after 2 Frames

10th Frame | Depth Map after 10 Frames

50th Frame | Depth Map after 50 Frames

Figure 5.13: The Straw Hat Sequence

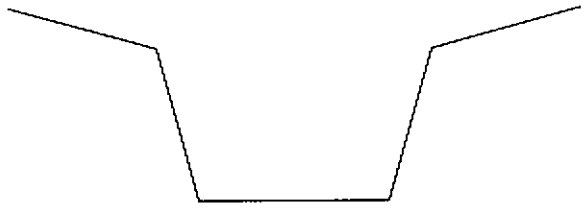
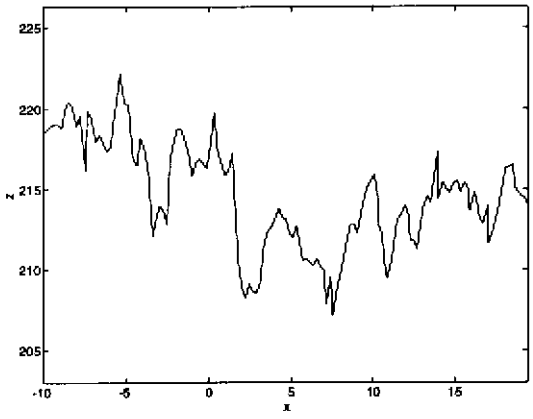
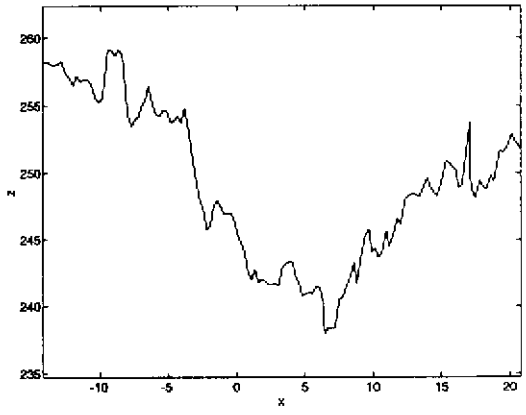


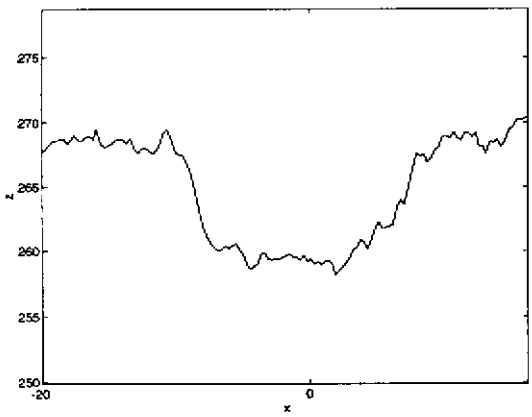
Figure 5.14: Sketch of True Shape



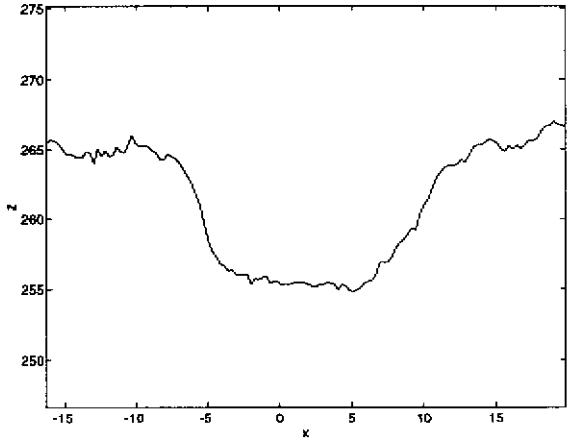
After 2 Frames



After 10 Frames



After 30 Frames



After 50 Frames

Figure 5.15: Evolution of the Shape of Straw Hat

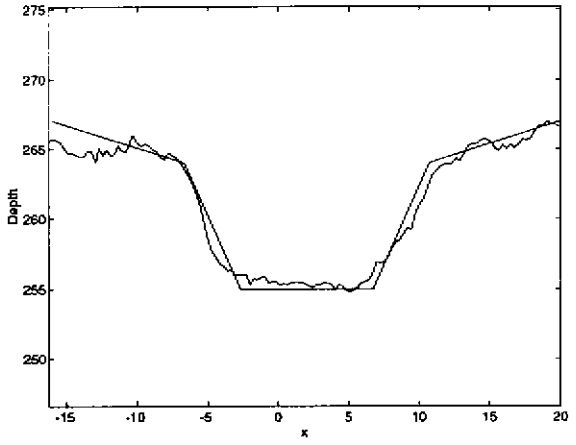


Figure 5.16: Measuring Quality of Results

object was a real chair we used in our lab. The chair was rotating in front of the static camera as in Figure 5.17. Digitization was done by a Matrox board. The optical flow computation sometimes returned wrong results at some locations, which we believe were caused by texture aliasing. We can see that, even in the tenth frame, the two buttons are very clear in the depth map. Also noticeable is that part of the chair in the left side is moving out and part of the chair in the right side is moving in. The move-in part is at first rather noisy, but then gradually becomes smoother and smoother.

- *The Cube Sequence*: The camera had approximately a 22° field of view. The sequence in Figure 5.18 was taken by a hand-held video camera connected to a Umatic recorder. It was recorded on a Umatic SP videotape and then digitized by a BVU digitizer, which could capture only one field in a frame. The digitized images have significantly higher noise levels than those digitized by the Matrox board. We can see that the system still performs fairly well on those noisy images.

- *The Basket Sequence*: The camera had a 22° field of view. The target was a basket which was moving and rotating in front of the camera. The digitization was done the same way as the cube sequence was performed.

- *The Sphere and Dog Sequence*: The camera had an 11° field of view. The target was a toy dog on top of a ball. The difficulties of this sequence were that (1)

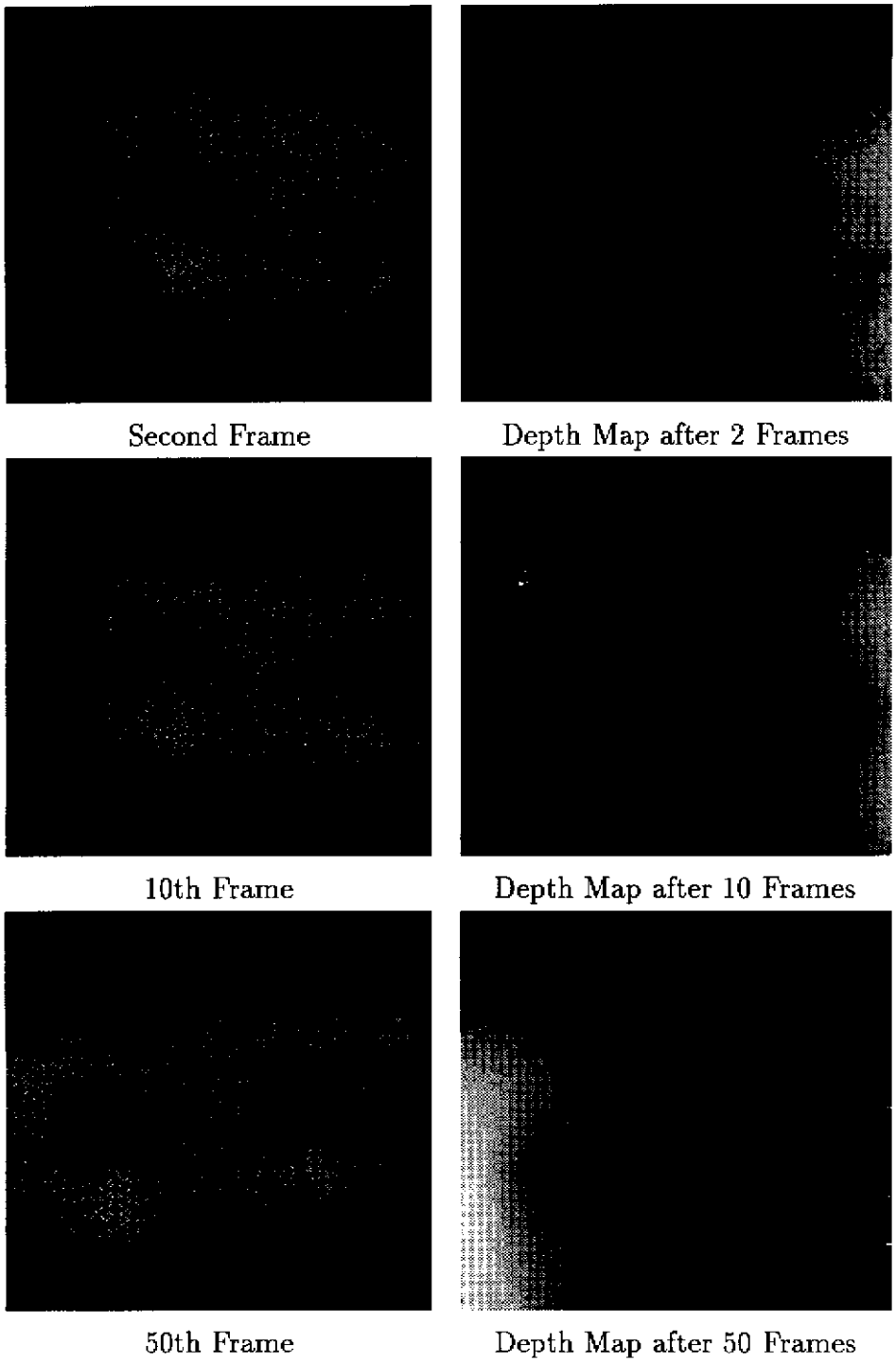


Figure 5.17: The Chair Sequence

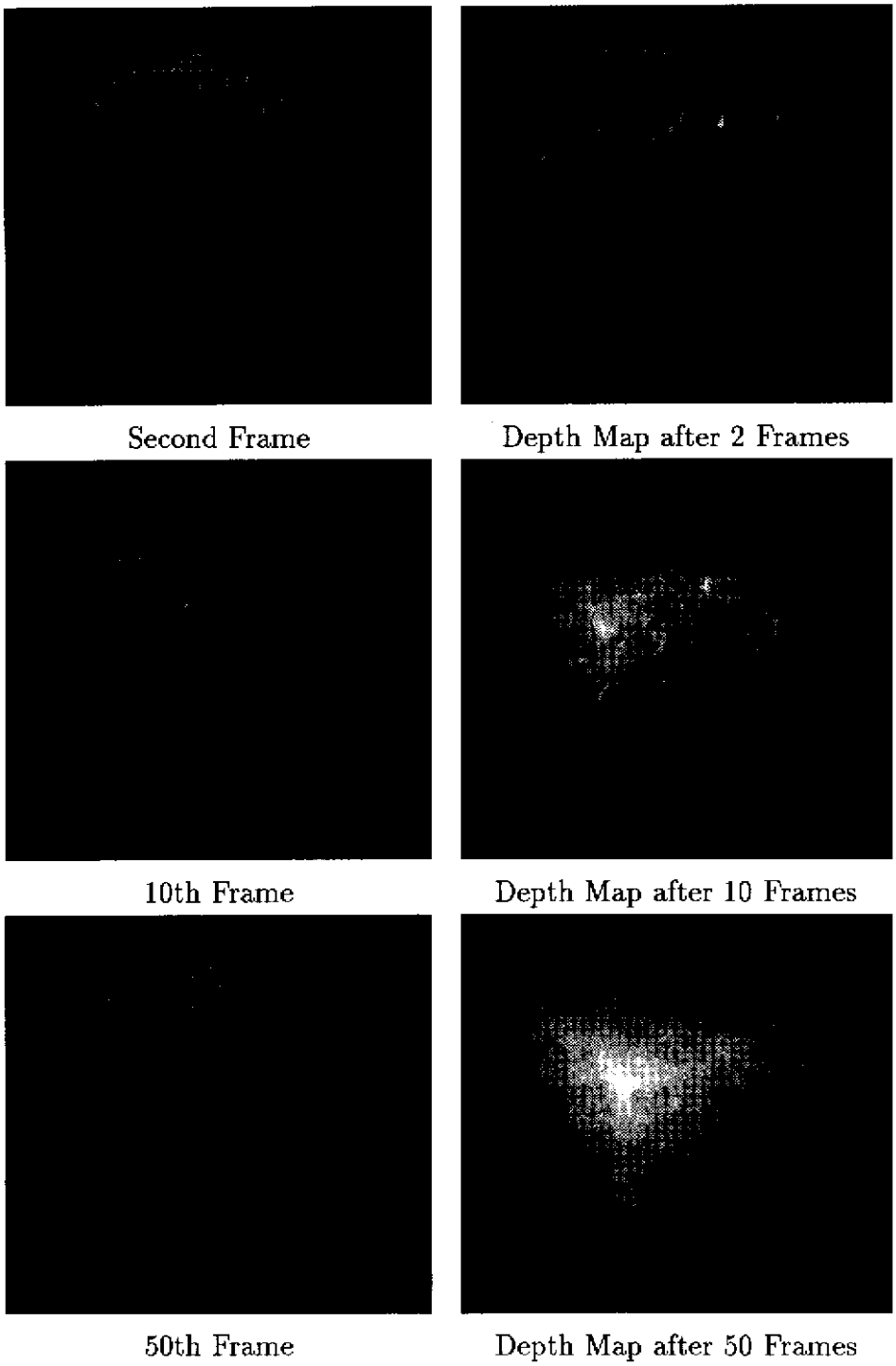
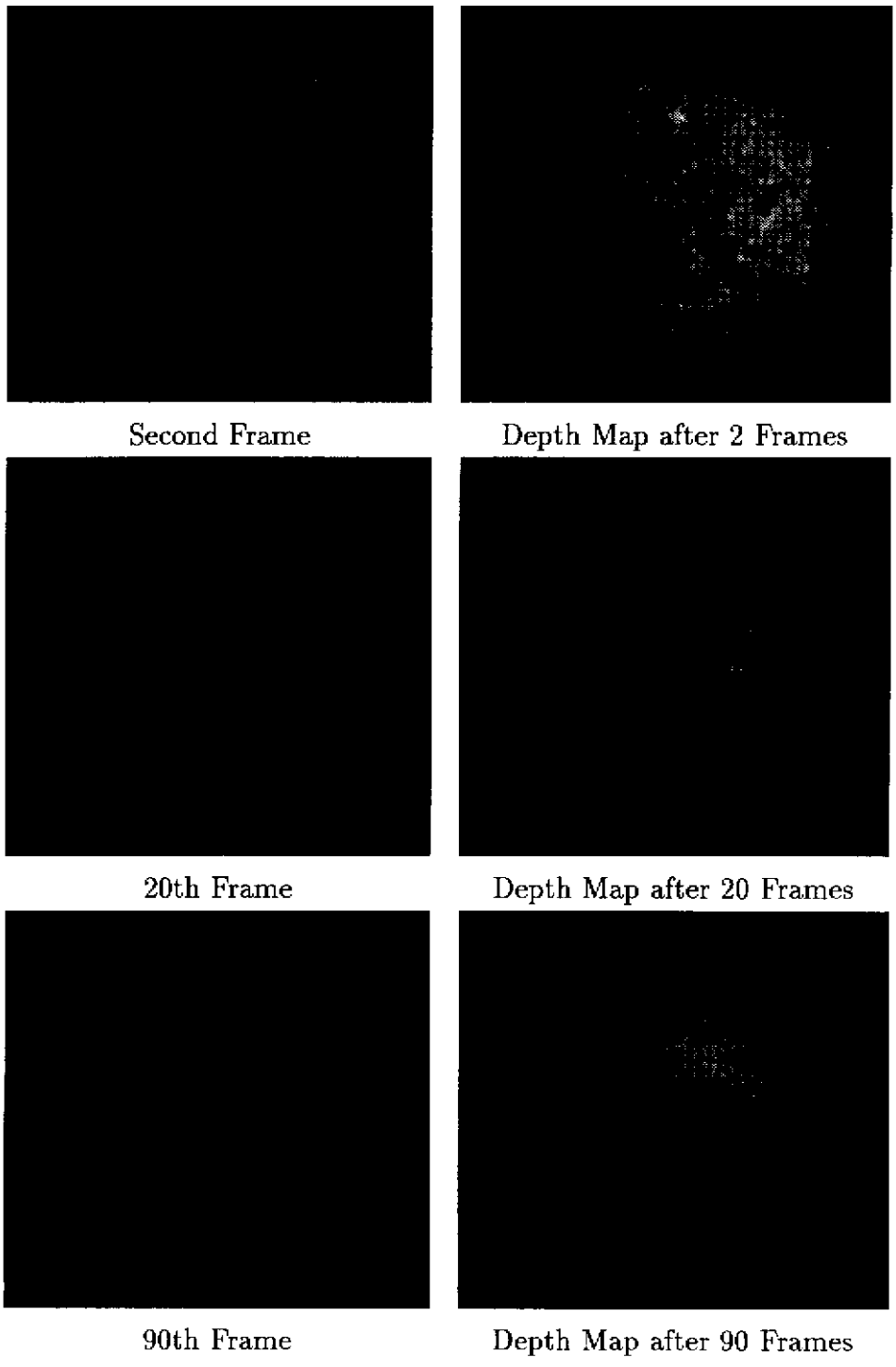


Figure 5.18: The Cube Sequence



Second Frame | Depth Map after 2 Frames

20th Frame | Depth Map after 20 Frames

90th Frame | Depth Map after 90 Frames

Figure 5.19: The Basket Sequence

the ball had a very low contrast near its boundaries; (2) it had obvious specular reflections which confused the optical flow algorithm, and (3) the toy dog had a sparse texture. Despite these difficulties, our system still performed reasonably well as in Figure 5.20. The different depth values of the sphere and dog are clearly visible, and even the depth of the tail tip of the dog is correctly shown.

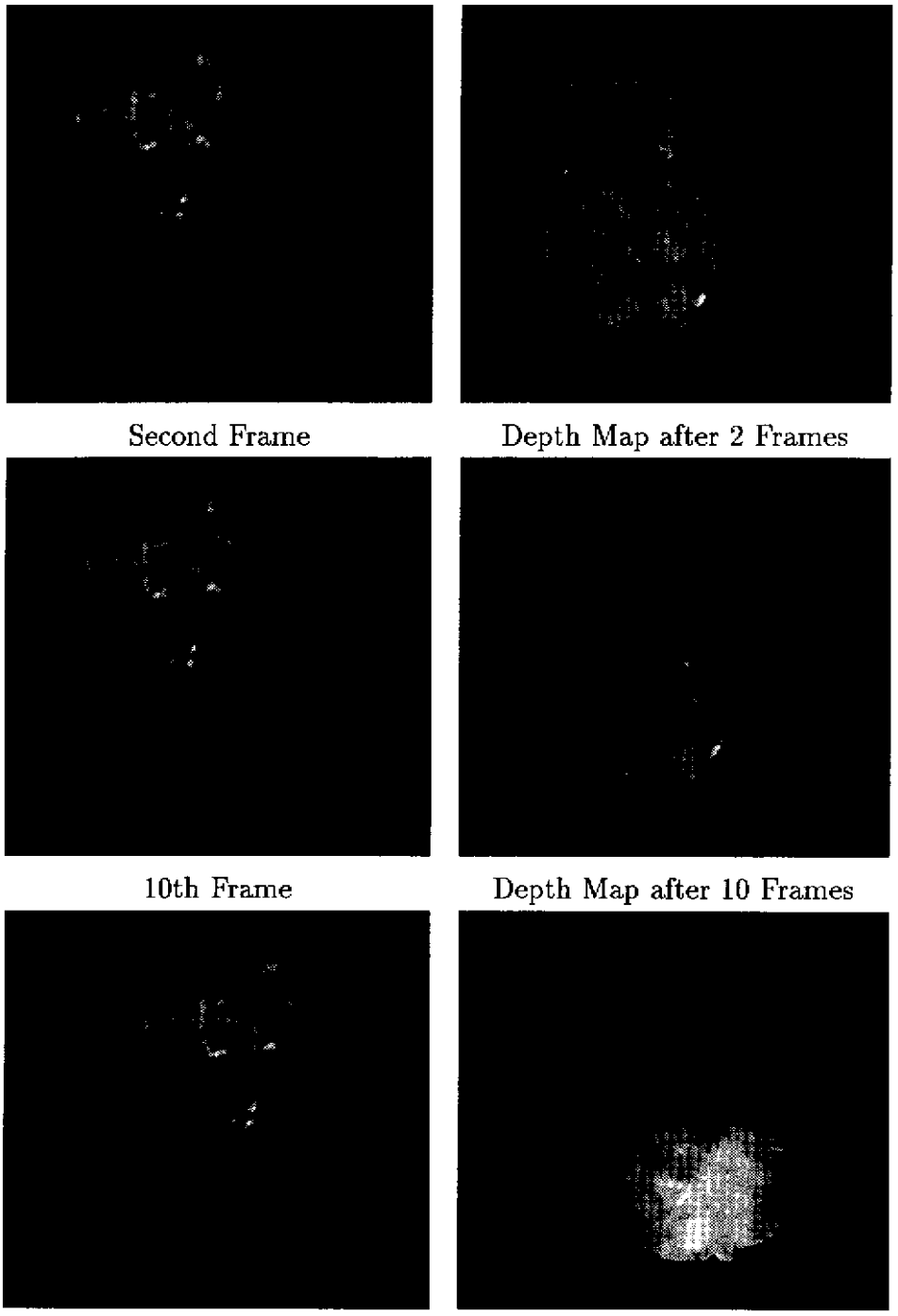
In all the experiments, the initial structure information was set as a flat surface parallel to the image plane at depth 200 with independent uncertainty $\sigma = 100$ at every pixel. Despite such a crude initial estimation, there was no obvious error caused by nonlinearity in our experiments.

From all these experiments, we conclude that our system of recursively recovering dense structure from a dense optical flow sequence can converge to the true 3D shape of the scene quickly and accurately. We have demonstrated its robust performance under adverse conditions such as noisy images, specularity, and texture aliasing.

5.7 Summary

In summary, we presented an EKF-based system which recursively combines dense structural information from a sequence of optical flows. At the current stage, our system is able to deliver an evolving sequence of depth maps using optical flows. We also showed that the system was very robust when the optical flows were noisy or contain outliers caused by texture aliasing or specularity.

Current representation of 3D dense information by depth maps and their uncertainty is very limiting in that complicated objects can not be represented. In the future, we would like to expand our system to deliver a final 3D model of the scene based on the image sequence. In other words, we would like to maintain an independent module to store, retrieve and update 3D structural information. Therefore we could extract *a priori* depth information from the module for every optical flow frame, and merge *a posteriori* depth information into the module. The problem of representing 3D dense structure and its uncertainty (independent and correlated) still remains very challenging.



Second Frame | Depth Map after 2 Frames

10th Frame | Depth Map after 10 Frames

50th Frame | Depth Map after 50 Frames

Figure 5.20: The Sphere and Dog Sequence

Chapter 6

Summary, Major Contributions, and Future Directions

6.1 Summary

The goal of this thesis is to solve the precision problem in image matching and shape recovery after the image matching. We began our research into high precision matching by studying the traditional Gabor filter based approaches. We identified the window effects as one of the major reasons for inaccuracy in image matching. By measuring the similarity between a filter output and a infinitely long sinusoid, we proposed to use the generalized stability criterion to threshold information obtained from Gabor filters. We applied this technique to the focus matching and optical flow problem and observed progress in both precision and density.

Even though the stability criterion could identify the serious cases when the filter outputs were overwhelmingly contaminated by window effects, it did not provide any way to correct minor errors caused by window effects. In Chapter 3, we proposed novel moment filters. The moment filters allow the window effects to be represented as an infinitely long Taylor series, while the traditional Gabor filter approach used in Chapter 2 is just a special case of truncating the series after the first term. By using higher order terms in the series, we could drastically reduce the matching errors. Furthermore, because of the recursiveness of moment filters in the spatial domain, the shift variance problem in image matching could also be corrected and the spatial derivative of the matching parameters could also be estimated simultaneously. We applied the moment filter approach to focus and stereo matching, and it achieved

much higher precision than did previous state-of-the-art techniques.

Though the moment filters were able to eliminate window and shift variance effects, they were awkward in handling frequency sampling problem. Based on moment filters, we proposed another set of filters – hypergeometric filters – which were also recursive in both the spatial and frequency domains and therefore were able to analytically represent window and shift variance effects. In addition, the set of hypergeometric filters is orthogonal and complete, and thus we do not need to worry about the sampling of the frequency domain. We demonstrated the effectiveness of the hypergeometric filters on optical flow, affine matching and other matching problems.

Using the high precision optical flows, we proceeded to compute 3D structure from uncalibrated motions. Here the major difficulty lies in the high computational complexity of recovering dense 3D structure and handling of complicated uncertainty information. We proposed to decompose structural uncertainty information into independent and correlated parts, and adopted the weighted principal component analysis to compress correlated uncertainty information. We decreased the computation and memory requirements from $O(N^2)$ to $O(N)$, which made it possible to recover *dense* structural information. A dense structure from motion system was demonstrated.

6.2 Major Contributions

The major contributions of the thesis are:

- *A Unified Approach to Image Matching*: The approaches proposed in the thesis are for the general image matching problem. We demonstrated that the same approaches could be applied to a wide range of problems such as focus matching, stereo matching, optical flow matching, and affine matching.

  - *Moment Filters*: We developed the novel "moment" filters based on the traditional Gabor filters. The window effects and shift variance effects could be represented analytically by the filter outputs due to the recursiveness of the moment filters in the spatial and frequency domains.

  - *Hypergeometric Filters*: We developed the novel "hypergeometric" filters. Not only could the window effects and shift variance effects be represented analytically by the filter outputs due to the recursiveness of the hypergeometric filters in the spatial and frequency domains, but also, unlike the

moment filters, the set of hypergeometric filters is non-redundant and complete.

- *High Precision in Image Matching*: In addition to the generality of the approaches, we also demonstrated that our approaches achieved much higher precision that those state-of-the-art techniques designed specifically for those problems. The reason for the high precision of our approaches is that the window effects and shift variance effects are explicitly considered.

  - *Depth from Focus*: Not only did our approach to the focus matching problem produce results of higher quality as shown in the experiments, but it was also general in the sense that it is applicable to any blurring model.
  - *Depth from Stereo*: A simple implementation of the moment filter approach on the stereo matching problem was superior to the most sophisticated stereo matching algorithm in precision. We also showed that our approach is able to deal with the foreshortening problem without warping images.
  - *High Precision Optical Flow*: We implemented the hypergeometric filter approach on the problem of optical flow matching. Compared to a number of other state-of-the-art techniques, our results were the best in both precision and density even though we used only two frames in a sequence and we did not do any smoothing.
  - *Affine Matching without Warping*: We implemented the hypergeometric filter approach in the problem of affine matching. Because the affine transformation in the spatial domain could be represented as linear combination in the filter space, our approach did not need image warping and achieved very high precision in computing six affine parameters simultaneously.

- *Dense Structure from Motion System*: We demonstrated a system which is capable of delivering dense structure from an optical flow sequence. Unlike traditional systems which require $O(N^2)$ or larger an amount of computation and memory, making it impossible for them to process dense structure, our system requires only $O(N)$ computation and memory.

  - *Separation of Independent and Correlated Uncertainty*: In the structure from motion system, we proposed the separation of independent and correlated

uncertainty. To the problem of structure from motion, since the rank of correlated uncertainty is of low rank, the separation enabled us to store and manipulate an $N \times N$ matrix using only $O(N)$ computation and memory.

- *Weighted Principal Component Analysis*: We represented the correlated uncertainty by using weighted principal components. By doing so, we could compress the information by truncating small eigenvalues. Additionally, the eigenimage/eigenvalue representation of the correlated uncertainty made it easy to interpolate the correlated uncertainty.

6.3 Future Directions

There were many directions which we hope to pursue in the future. In a sense, these directions complement and/or advance the research reported in the thesis. Here we list a couple of topics which, we think, are important and could be pursued based on our work:

- *Discontinuity in Matching*: Even though we mentioned in Chapter 3 that the discontinuity problem could be regarded as an extreme of the shift variance problem, e.g., the discontinuity in stereo is when the foreshortening is infinitely large or, at least, very large. And our experiments showed that indeed depth discontinuity generated very large values in slope estimation in both focus and stereo matching. It seems promising that the discontinuity problem could be approached in this way.

- *3D Modelling from Motion*: As mentioned in Chapter 5, the depth map representation of dense structure was extremely limiting in that complicated structures could not be represented properly. Therefore, in order to build complicated 3D models out of image sequences, we need to use a more advanced and flexible 3D representation to replace the depth map representation we used in our system.

- *Uncertainty-Driven Sensor Planning*: In the context of active vision, it is commonly believed that the sensor should be controlled to actively *seek* information instead of waiting for the light to fall on it. One key question is how to represent the lack of information. Whaite and Ferrie [96] used uncertainty of parameters of superellipses to represent the uncertainty of shape. But such a representation

has a very limited applicability. Our representation of uncertain shape by eigen-images/eigenvalues is probably better for this task since it is not limited to any specific type of shape.

Appendix A

Stability Criteria for an Arbitrary Signal

Let us assume an arbitrary signal $i(x)$, and its Fourier transform

$$I(f) = \int_{-\infty}^{+\infty} i(x) e^{-jfx} dx. \tag{A.1}$$

Its Gabor transform in any location x_0 and any peak frequency f_0 can be represented as

$$\begin{aligned} G(x_0, f_0) &= \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{+\infty} i(x - x_0) e^{-\frac{x^2}{2\sigma^2}} e^{-jf_0 x} dx \\ &= \int_{-\infty}^{+\infty} I(f) e^{-jfx_0} e^{-(f_0 - f)^2 \sigma^2 / 2} df \\ &= a(x_0, f_0) e^{j\phi(x_0, f_0)} \\ &= R_0(x_0, f_0) + j I_0(x_0, f_0). \end{aligned}$$

Then, we have

$$\frac{\partial \phi}{\partial x_0} = \frac{R_0 \frac{\partial I_0}{\partial x_0} - I_0 \frac{\partial R_0}{\partial x_0}}{R_0^2 + I_0^2}, \tag{A.2}$$

$$\frac{1}{a} \frac{\partial a}{\partial x_0} = \frac{R_0 \frac{\partial R_0}{\partial x_0} + I_0 \frac{\partial I_0}{\partial x_0}}{R_0^2 + I_0^2}. \tag{A.3}$$

Similarly

$$f_0 \frac{\partial \phi}{\partial f_0} = f_0 \frac{R_0 \frac{\partial I_0}{\partial f_0} - I_0 \frac{\partial R_0}{\partial f_0}}{R_0^2 + I_0^2}, \tag{A.4}$$

$$\frac{f_0}{a} \frac{\partial a}{\partial f_0} = f_0 \frac{R_0 \frac{\partial R_0}{\partial f_0} + I_0 \frac{\partial I_0}{\partial f_0}}{R_0^2 + I_0^2}. \tag{A.5}$$

Let us define $G_1(x_0, f_0)$ as

$$\begin{aligned} G_1(x_0, f_0) &= \frac{j}{\sqrt{2\pi}\sigma^3}\int_{-\infty}^{+\infty} i(x-x_0)xe^{-\frac{x^2}{2\sigma^2}}e^{-jfx}dx & \text{(A.6)}\\ &= \int_{-\infty}^{+\infty} I(f)e^{-jfx_0}(f_0-f)e^{-(f_0-f)^2\sigma^2/2}df & \text{(A.7)}\\ &= R_1 + jI_1, \end{aligned}$$

where the equivalence between Eq. A.7 and Eq. A.6 can be easily verified by using the following property of Fourier transform:

$$\mathcal{F}[gh] = \mathcal{F}[g] \otimes \mathcal{F}[h], \tag{A.8}$$

where $\mathcal{F}$ denotes the Fourier transform operator.

Now, we have

$$\begin{aligned} \frac{\partial G}{\partial x_0} &= \frac{\partial R_0}{\partial x_0} + j\frac{\partial I_0}{\partial x_0}\\ &= -\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{+\infty} i'(x-x_0)e^{-\frac{x^2}{2\sigma^2}}e^{-if_0x}dx & \text{(A.9)}\\ &= -\frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{+\infty} i(x-x_0)(\frac{x}{\sigma^2}+jf_0)e^{-\frac{x^2}{2\sigma^2}}e^{-jf_0x}dx & \text{(A.10)}\\ &= j(G_1(x_0,f_0) - f_0G(x_0,f_0)), & \text{(A.11)} \end{aligned}$$

where from step A.9 to step A.10 we used the partial integration method:

$$\int u'(x)v(x)dx = u(x)v(x) - \int u(x)v'(x)dx.$$

Using the same technique, we have

$$\begin{aligned} \frac{\partial G}{\partial f_0} &= \frac{\partial R_0}{\partial f_0} + j\frac{\partial I_0}{\partial f_0}\\ &= -\int_{-\infty}^{+\infty} I(f)e^{-jfx_0}\sigma^2(f_0-f)e^{-(f_0-f)^2\sigma^2/2}df\\ &= -\sigma^2 G_1(x_0,f_0). & \text{(A.12)} \end{aligned}$$

Replacing Eq. A.11 and Eq. A.12 into Eq. A.2 A.3 A.4 A.5, we obtain

$$T_1 = \frac{\partial\phi}{\partial x_0} - f_0 = \frac{R_0R_1 + I_0I_1}{R_0^2 + I_0^2}, \tag{A.13}$$

$$T_2 = f_0\frac{\partial\phi}{\partial f_0} = \frac{\sigma^2 f_0(I_0R_1 - R_0I_1)}{R_0^2 + I_0^2}, \tag{A.14}$$

$$T_3 = \frac{1}{a}\frac{\partial a}{\partial x} = \frac{I_0R_1 - R_0I_1}{R_0^2 + I_0^2}, \tag{A.15}$$

$$T_4 = \frac{f_0}{a}\frac{\partial a}{\partial f_0} = -\frac{\sigma^2 f_0(R_0R_1 + I_0I_1)}{R_0^2 + I_0^2}. \tag{A.16}$$

Therefore, we have

$$T_1^2 + T_3^2 = \frac{1}{(\sigma^2 f_0)^2}(T_2^2 + T_4^2) = \frac{R_1^2 + I_1^2}{R_0^2 + I_0^2}. \tag{A.17}$$

Appendix B

Orthogonality and Completeness of Moment Filters

First, let us rephrase the theorems from [67] (page 23, page 29, page 59) and list them as following:

Theorem B.1 *The generalized Laguerre polynomial $L(n, \alpha, x)$ is the solution of the following differential equation:*

$$xy'' + (1 + \alpha - x)y' + ny = 0. \tag{B.1}$$

Theorem B.2 *If*

$$x^{k+1+\alpha}e^{-x}|_{x=a,b} = 0, (k = 0, 1, ...), \tag{B.2}$$

at the endpoints of an interval (a, b), then the polynomial $L(n, \alpha, x)$ corresponding to different n's are orthogonal on (a, b), i.e.

$$\int_a^b L(n, \alpha, x)L(m, \alpha, x)x^{\alpha}e^{-x}dx = 0, when(m \neq n). \tag{B.3}$$

Theorem B.3 *Let $f(x)$ be continuous on $a < x < b$ and have a piecewise continuous derivative in this interval, If the integrals*

$$\int_a^b f^2(x)x^{\alpha}e^{-x}dx, \quad \int_a^b [f'(x)]^2 x^{1+\alpha}e^{-x}dx \tag{B.4}$$

converge, then the following expansion is valid:

$$f(x) = \sum_{n=0}^{\infty} c_n L(n, \alpha, x), \tag{B.5}$$

where

$$c_n = \frac{\int_a^b f(x)L(n,\alpha,x)x^\alpha e^{-x}dx}{\int_a^b L(n,\alpha,x)L(n,\alpha,x)x^\alpha e^{-x}dx}. \tag{B.6}$$

Let us set $f_0 = 0$ in Eq. 3.8, we will prove that in this case the set of $k_i(x)$ is orthogonal and complete. The case when $f_0 \neq 0$ can be similarly proved.

Because k_{2n} is an even function, and k_{2m+1} is an odd function, we have

$$< k_{2n}(x), k_{2m+1}(x) >=< k_{2m+1}(x), k_{2n}(x) >= 0. \tag{B.7}$$

And

$$\begin{aligned} & < k_{2n}(x), k_{2m}(x) > \\ = & \frac{2^{m+n}m!n!}{\sqrt{2\pi}\sigma^{2(m+n)+1}} \int_{-\infty}^{+\infty} L(n, -\frac{1}{2}, \frac{x^2}{2\sigma^2})L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2})e^{-\frac{x^2}{2\sigma^2}}dx \\ = & \frac{2^{m+n}m!n!}{\sqrt{2\pi}\sigma^{2(m+n)+1}} \frac{\sigma}{\sqrt{2}} \int_0^{+\infty} L(n, -\frac{1}{2}, t)L(m, -\frac{1}{2}, t)t^{-\frac{1}{2}}e^{-t}dt, \end{aligned} \tag{B.8}$$

where we replace $\frac{x^2}{2\sigma^2}$ by t. Set the interval (a, b) to $(0, +\infty)$, and apply Theorem A.2 to the above equation; we have

$$< k_{2n}(x), k_{2m}(x) >= 0, \text{when}(m \neq n). \tag{B.9}$$

Similarly

$$\begin{aligned} & < k_{2n+1}(x), k_{2m+1}(x) > \\ = & -\frac{2^{m+n}m!n!}{\sqrt{2\pi}\sigma^{2(m+n)+5}} \int_{-\infty}^{+\infty} x^2 L(n, \frac{1}{2}, \frac{x^2}{2\sigma^2})L(m, \frac{1}{2}, \frac{x^2}{2\sigma^2})e^{-\frac{x^2}{2\sigma^2}}dx \\ = & -\frac{2^{m+n}m!n!}{\sqrt{2\pi}\sigma^{2(m+n)+2}}\sqrt{2} \int_0^{+\infty} L(n, \frac{1}{2}, t)L(m, \frac{1}{2}, t)t^{\frac{1}{2}}e^{-t}dt \\ = & 0, \text{when}(m \neq n). \end{aligned} \tag{B.10}$$

From Eq. B.7, Eq. B.9 and Eq. B.10, we prove the orthogonality of the moment filters, i.e.

$$< k_i(x), k_j(x) >= 0, \text{when}(i \neq j). \tag{B.11}$$

For any function $\xi(x)$, it can always be decomposed into the sum of an odd function $\xi_o(x)$ and an even function $\xi_e(x)$ as

$$\begin{aligned} \xi_o(x) &= \frac{1}{2}(\xi(x) - \xi(-x)), \\ \xi_e(x) &= \frac{1}{2}(\xi(x) + \xi(-x)), \\ \xi(x) &= \xi_e(x) + \xi_o(x). \end{aligned} \tag{B.12}$$

From Theorem A.3, in the interval $(0, +\infty)$, for any function $g(t)$ which satisfies Eq. B.4, we have

$$t^{-1/2}g(t) = \sum_{n=0}^{\infty} c_n L(n, \frac{1}{2}, t), \tag{B.13}$$

$$g(t) = \sum_{m=0}^{\infty} d_m L(m, -\frac{1}{2}, t). \tag{B.14}$$

Therefore, let $t = \frac{x^2}{2\sigma^2}$, we have

$$g(\frac{x^2}{2\sigma^2}) = \sqrt{2}\sigma \sum_{n=0}^{\infty} c_n x L(n, \frac{1}{2}, \frac{x^2}{2\sigma^2}), \tag{B.15}$$

$$g(\frac{x^2}{2\sigma^2}) = \sum_{m=0}^{\infty} d_m L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}). \tag{B.16}$$

Because $g(t)$ can be any function, we can let $g(\frac{x^2}{2\sigma^2}) = \xi_o(x)$ in Eq. B.15, and $g(\frac{x^2}{2\sigma^2}) = \xi_e(x)$ in Eq. B.16. Replacing them into Eq. B.12, in $(0, \infty)$ we have

$$\begin{aligned}\xi(x) &= \sqrt{2}\sigma \sum_{n=0}^{+\infty} c_n x L(n, \frac{1}{2}, \frac{x^2}{2\sigma^2}) + \sum_{m=0}^{+\infty} d_m L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) \\ &= \sum_{i=0}^{+\infty} \frac{w_i}{< k_i(x), k_i(x) >} k_i(x),\end{aligned} \tag{B.17}$$

where, when $i = 2m$,

$$\begin{aligned}w_{2m} &= \frac{\sigma^{2m}}{2^m m!} < k_{2m}(x), k_{2m} > d_m \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \frac{2^m m!}{\sigma^{2m}} L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}} dx \\ &\quad \frac{\int_0^{+\infty} \xi_e(x) L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2})(\frac{x^2}{2\sigma^2})^{-1/2} e^{-\frac{x^2}{2\sigma^2}} d(\frac{x^2}{2\sigma^2})}{\int_0^{+\infty} L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2})(\frac{x^2}{2\sigma^2})^{-1/2} e^{-\frac{x^2}{2\sigma^2}} d(\frac{x^2}{2\sigma^2})} \\ &= \frac{2^m m!}{\sigma^{2m}} \frac{\sqrt{2}}{\sqrt{\pi}\sigma} \int_0^{+\infty} \xi_e(x) L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}} dx \\ &= \frac{2^m m!}{\sigma^{2m}} \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{+\infty} \xi_e(x) L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}} dx \\ &= \frac{2^m m!}{\sigma^{2m}} \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{+\infty} \xi(x) L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}} dx \\ &= < \xi(x), k_{2m}(x) > .\end{aligned} \tag{B.18}$$

Similarly when $i = 2n + 1$,

$$w_{2n+1} = < \xi(x), k_{2n+1}(x) > . \tag{B.19}$$

Similarly in interval $(-\infty, 0)$, we also have

$$w_i = < \xi(x), k_i(x) > . \tag{B.20}$$

Appendix C

Properties of Moment Filters

C.1 Recursive Properties of Moment Filters

From Formula 22.3.9 in [1], we have the the explicit expression for the generalized Laguerre polynomial as

$$L(n,\alpha,x) = \sum_{m=0}^{n} \frac{\Gamma(n+\alpha+1)(-1)^m}{\Gamma(m+\alpha+1)m!(n-m)!}x^m. \tag{C.1}$$

Therefore, we have

$$\begin{aligned}
& L(n,1/2,x) - L(n-1,1/2,x) \\
= & \sum_{m=0}^{n} \frac{\Gamma(n+3/2)(-1)^m}{\Gamma(m+3/2)m!(n-m)!}x^m + \sum_{m=0}^{n-1} \frac{\Gamma(n+1/2)(-1)^m}{\Gamma(m+3/2)m!(n-1-m)!}x^m \\
= & \sum_{m=0}^{n-1} \frac{\Gamma(n+1/2)(-1)^m}{\Gamma(m+3/2)m!(n-m)!}((n+1/2)-(n-m))x^m + \frac{(-1)^n}{n!}x^n \\
= & \sum_{m=0}^{n} \frac{\Gamma(n-1/2+1)(-1)^m}{\Gamma(m-1/2+1)m!(n-m)!}x^m \\
= & L(n,-1/2,x).
\end{aligned} \tag{C.2}$$

Similarly,

$$(n+1/2)L(n,-1/2,x) - (n+1)L(n+1,-1/2,x) = xL(n,1/2,x). \tag{C.3}$$

Applying Eq. C.2 and Eq. C.3 to $p_i(x)$ in Eq. 3.2, we have

$$xp_{2n}(x) = \frac{2^n n!}{\sigma^{2n}} xL(n,-\frac{1}{2},\frac{x^2}{2\sigma^2})$$

$$
\begin{aligned}
&= \frac{2^n n!}{\sigma^{2n}}\left(xL(n,\frac{1}{2},\frac{x^2}{2\sigma^2}) - xL(n,\frac{1}{2},\frac{x^2}{2\sigma^2})\right) \\
&= -j(\sigma^2 p_{2n+1}(x) - 2nP_{2n-1}(x)) \qquad \text{(C.4)} \\
xp_{2n+1}(x) &= j\frac{2^n n!}{\sigma^{2n+2}}x^2 L(n,\frac{1}{2},\frac{x^2}{2\sigma^2}) \\
&= j\frac{2^n n!}{\sigma^{2n+2}}2\sigma^2\left((n+1/2)L(n,-\frac{1}{2},\frac{x^2}{2\sigma^2}) - (n+1)L(n+1,-\frac{1}{2},\frac{x^2}{2\sigma^2})\right) \\
&= j((2n+1)p_{2n}(x) - \sigma^2 p_{2n+2}(x)). \qquad \text{(C.5)}
\end{aligned}
$$

Actually Eq. C.4 and Eq. C.5 can be merged into a single formula:

$$xp_i(x) = j(ip_{i-1}(x) - \sigma^2 p_{i+1}(x)). \qquad \text{(C.6)}$$

Replacing Eq. C.6 into Eq. 3.1, we then obtain the recursiveness of the moment filters in the spatial domain,

$$xm_i(x) = j(im_{i-1}(x) - \sigma^2 m_{i+1}(x)). \qquad \text{(C.7)}$$

Another recursive property of the moment filters is with respect to differentiation, i.e.

$$
\begin{aligned}
m_i'(x) &= j^{-i}\frac{d}{dx}\left(e^{-jf_0 x}\frac{d^i}{dx^i}g(x)\right) \\
&= j(m_{i+1}(x) - f_0 m_i(x)). \qquad \text{(C.8)}
\end{aligned}
$$

The following recursive property of the moment filters in the frequency domain is obviously true from Eq. 3.4:

$$(f - f_0)M_i(f) = M_{i+1}(f). \qquad \text{(C.9)}$$

C.2 Relations to Instantaneous Frequency and Stability Criterion

When we consider the filtering of the whole signal instead of only a specific spatial location, the weights w_i in Eq. 3.13 become a function of the location, i.e.

$$w_i(x_0) = \int_{-\infty}^{+\infty} \xi(x)m_i(x - x_0)dx. \qquad \text{(C.10)}$$

We know that $w_0(x_0)$ is the output of the Gabor filter; let

$$w_0(x_0) = R_0(x_0) + jI_0(x_0),$$

where $R_0(x_0)$ and $I_0(x_0)$ are the real and imaginary parts of $w_0(x_0)$ respectively. Then the instantaneous frequency is defined as the spatial derivative of phase value, i.e.

$$\begin{aligned} \frac{d}{dx_0}\phi(x_0) &= \frac{R_0 I_0' - R_0' I_0}{R_0^2 + I_0^2} \\ &= \frac{\mathrm{Im}(w_0^* w_0')}{\| w_0 \|^2}, \end{aligned} \tag{C.11}$$

where Im means the imaginary part.

From Eq. C.10 and Eq. 3.17, we can compute the spatial derivative of w_0 as

$$\begin{aligned} w_0' &= -\int_{-\infty}^{+\infty} \xi(x) m_0'(x - x_0) dx \\ &= -j \int_{-\infty}^{+\infty} \xi(x)(m_1(x - x_0) - f_0 m_0(x - x_0)) dx \\ &= j(f_0 w_0 - w_1). \end{aligned} \tag{C.12}$$

Replacing Eq. C.12 into Eq. C.11, we then have

$$\frac{d}{dx_0}\phi(x_0) = f_0 - \frac{\mathrm{Re}(w_1 w_0^*)}{\| w_0 \|^2} = f_0 - \mathrm{Re}\left(\frac{w_1}{w_0}\right), \tag{C.13}$$

where Re means the real part.

The stability criterion of the Gabor filter output is usually formulated as a threshold ([24, 101], Chapter2):

$$T^2 = \left(\frac{1}{a}\frac{da}{dx_0}\right)^2 + \left(\frac{d\phi}{dx_0} - f_0\right)^2, \tag{C.14}$$

where a is the amplitude.

Because

$$\begin{aligned} \frac{1}{a}\frac{da}{dx_0} &= \frac{R_0 R_0' + I_0 I_0'}{R_0^2 + I_0^2} \\ &= \frac{\mathrm{Re}(w_0^* w_0')}{\| w_0 \|^2} \\ &= -\frac{\mathrm{Im}(w_1 w_0^*)}{\| w_0 \|^2} \\ &= -\mathrm{Im}\left(\frac{w_1}{w_0}\right), \end{aligned} \tag{C.15}$$

we then have

$$T^2 = \frac{\| w_1 \|^2}{\| w_0 \|^2}. \tag{C.16}$$

Appendix D

Error Estimations in Moment Filter Approach

D.1 Focus

Suppose the energy of the white noise in the image is N_w^2; the noise energy in U_i or V_i will be:

$$W_i^2 = E_i^2 N_w^2, \tag{D.1}$$

where E_i^2 is the energy of the filter in Eq. 3.35.

Let

$$\begin{aligned} C_0(u) &= U_0 + f_0 u U_1 - \frac{1}{2}u(1 - f_0^2 u)U_2, \\ C_1(u) &= U_1 + f_0 u U_2 - \frac{1}{2}u(1 - f_0^2 u)U_3. \end{aligned}$$

Replacing them back to Eq. 3.43 and Eq. 3.45, and taking the two error sources, i.e., white noise $W()$ and truncation error $T()$, into consideration, we have:

$$u = \frac{2}{f_0^2}\left(\ln(C_0(u) + W(C_0) + T(C_0)) - \ln(V_0 + vP_0 + W(V_0))\right), \tag{D.2}$$

$$\frac{V_1 + vP_1 + W(V_1)}{V_0 + vP_0 + W(V_0)}v = \frac{C_1(u) + W(C_1) + T(C_1)}{C_0(u) + W(C_0) + T(C_0)}. \tag{D.3}$$

While the magnitude of the white noise error $W(C_0), W(C_1), W(V_0)$ and $W(V_1)$ can be estimated from Eq. D.1, the magnitude of the truncation error (assuming the

third order is truncated) can be estimated as

$$\begin{aligned} \| T(C_0) \|^2 &= \| e^{f_0^2 s^2/2}(V_0 + vP_0) - C_0 \|^2 + \frac{E_3^2}{E_0^2} \| f_0 s^4 (3 - f_0^2 s^2) U_0 \|^2, \\ \| T(C_1) \|^2 &= \| e^{f_0^2 s^2/2}(V_1 + vP_1) - C_1 \|^2 + \frac{E_4^2}{E_0^2} \| f_0 s^4 (3 - f_0^2 s^2) U_0 \|^2, \end{aligned}$$

where the first term is the residual in Eq. 3.43, and the second the estimated energy of the truncated part.

From Eq. D.2 and Eq. D.3, by assuming all the error sources are independent, we have, after some manipulations,

$$\begin{bmatrix} du \\ dv \end{bmatrix} = \begin{bmatrix} \frac{f_0^2}{2} - \frac{C_0'(u)}{C_0(u)} & \frac{P_0}{V_0 + vP_0} \\ (V_0 + vP_0)C_1'(u) - (V_1 + vP_1)C_0'(u) & C_1P_0 - C_0P_1 \end{bmatrix}^{-1} \cdot$$

$$\begin{bmatrix} \frac{1}{C_0(u)} & 0 & -\frac{1}{V_0 + vP_0} & 0 & \frac{1}{C_0(u)} & 0 \\ V_1 + vP_1 & -(V_0 + vP_0) & -C_1(u) & C_0(u) & V_1 + vP_1 & -(V_0 + vP_0) \end{bmatrix} \begin{bmatrix} W(C_0) \\ W(C_1) \\ W(V_0) \\ W(V_1) \\ T(C_0) \\ T(C_1) \end{bmatrix}.$$

Rewrite the above equation in a more concise form; we have

$$\begin{bmatrix} du \\ dv \end{bmatrix} = \mathbf{AW}, \tag{D.4}$$

where **A** is a 2x6 complex matrix, and **W** is the 6x1 error vector.

D.2 Stereo

Suppose the energy of the white noise in the image is N_w^2; then the noise energy in U_i or V_i will be:

$$W_i^2 = E_i^2 N_w^2, \tag{D.5}$$

where E_i^2 is the energy of the ith moment filter.

Let

$$C_0(D) = U_0 - jDU_1 - \frac{1}{2}D^2U_2, \tag{D.6}$$

$$C_1(D) = U_1 - jDU_2 - \frac{1}{2}D^2U_3, \tag{D.7}$$

and $C_0'(D)$ and $C_1'(D)$ be their derivatives respectively.

The magnitude of the truncation error (assuming the third order is truncated) can be estimated as

$$\| T(C_0) \|^2 = \| e^{-jf_0 D}(V_0 + \mu P_0) - C_0 \|^2 + \frac{E_3^2}{E_0^2} \| \frac{D^3}{6} U_0 \|^2, \quad \text{(D.8)}$$

$$\| T(C_1) \|^2 = \| e^{-jf_0 D}(V_1 + \mu P_1) - C_1 \|^2 + \frac{E_4^2}{E_0^2} \| \frac{D^3}{6} U_0 \|^2, \quad \text{(D.9)}$$

where the first term is the residual in Eq. 3.79, and the second the estimated energy of the truncated part.

We then have the error of D and μ as linear functions of the two error sources, i.e., white noise $W()$ and the truncation error $T()$, as follows:

$$\begin{bmatrix} dD \\ d\mu \end{bmatrix} = \begin{bmatrix} -jf_0 - \frac{C_0'(D)}{C_0(D)} & \frac{P_0}{V_0+\mu P_0} \\ (V_0+\mu P_0)C_1'(D) - (V_1+\mu P_1)C_0'(D) & C_1 P_0 - C_0 P_1 \end{bmatrix}^{-1} \cdot$$

$$\begin{bmatrix} \frac{1}{C_0(D)} & 0 & -\frac{1}{V_0+\mu P_0} & 0 & \frac{1}{C_0(D)} & 0 \\ V_1+\mu P_1 & -(V_0+\mu P_0) & -C_1(D) & C_0(D) & V_1+\mu P_1 & -(V_0+\mu P_0) \end{bmatrix} \begin{bmatrix} W(C_0) \\ W(C_1) \\ W(V_0) \\ W(V_1) \\ T(C_0) \\ T(C_1) \end{bmatrix}.$$

Rewrite the above equation into a more concise form; we have

$$\begin{bmatrix} dD \\ d\mu \end{bmatrix} = \mathbf{AW}, \quad \text{(D.10)}$$

where **A** is a 2x6 complex matrix, and **W** is the 6x1 error vector.

Appendix E

Properties of Hypergeometric Filters

In this appendix, we will prove some H filter properties which are fundamental to their applications. The properties include the relation between H filters and moment filters, the recursiveness in the frequency and spatial domains, and the recursiveness with respect to differential operation. First of all, let us list the of the confluent hypergeometric function $\Phi(a,c,x)$ properties, which are useful to us. All of the properties listed below are from Chapter 13 in [1]:

$$\Phi(a,c,x) = e^x \Phi(c-a,c,-x), \tag{E.1}$$

$$(1+a-c)\Phi(a,c,x) - a\Phi(a+1,c,x) + (c-1)\Phi(a,c-1,x) = 0, \tag{E.2}$$

$$c\Phi(a,c,x) - c\Phi(a-1,c,x) - x\Phi(a,c+1,x) = 0, \tag{E.3}$$

$$\frac{n!}{(\alpha+1)_n} L(n,\alpha,x) = \Phi(-n,\alpha+1,x). \tag{E.4}$$

From Eq. 4.8 and Eq. 4.10, using the relation between the Laguerre polynomial and the confluent hypergeometric function in Eq. E.4, we have

$$\begin{aligned} h_{2i}(x) + h_{-2i}(x) &= 2a_m \Phi(-i, \frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}} \\ &= 2a_m \frac{i!}{(1/2)_n} L(i, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}}, \end{aligned} \tag{E.5}$$

$$\begin{aligned} h_{2i+1}(x) - h_{-(2i+1)}(x) &= j2b_m x \Phi(-i, \frac{3}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}} \\ &= j2b_m \frac{i!}{(3/2)_n} x L(i, \frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}}, \end{aligned} \tag{E.6}$$

in which the right hand sides are the moment filters ($f_0 = 0$) multiplied by constants.

From the definition of the H filters in Eq. 4.1, Eq. 4.2, and Eq. 4.3, we have the recursive property of the H filters in the frequency domain,

$$\begin{aligned} fH_m(f) &= \frac{c_m}{c_{m+1}} H_{m+1}(f), \\ fH_0(f) &= \frac{c_0}{c_1}(H_1(f) - H_{-1}(f)), \\ fH_{-m}(f) &= -\frac{c_m}{c_{m+1}} H_{-(m+1)}(f). \end{aligned}$$

To prove the recursive property in the spatial domain, let us first simplify our notations by the following replacements:

$$t = \frac{x}{\sqrt{2}\sigma}, \tag{E.7}$$

$$\begin{aligned} h_m(t) &= \sqrt{\frac{2^m}{\pi\Gamma(m+1/2)\sigma}} e^{-t^2} \\ &\quad \left(\frac{1}{\sqrt{2}}\Gamma(\frac{m+1}{2})\Phi(-\frac{m}{2}, \frac{1}{2}, t^2) + j\sqrt{2}\Gamma(1+\frac{m}{2})\Phi(\frac{1-m}{2}, \frac{3}{2}, t^2)t \right) \\ &= \sqrt{\frac{2^m}{\pi\Gamma(m+1/2)\sigma}} \varphi_m(t), \end{aligned} \tag{E.8}$$

where

$$\begin{aligned} \varphi_m(t) &= e^{-t^2} \left(\frac{1}{\sqrt{2}}\Gamma(\frac{m+1}{2})\Phi(-\frac{m}{2}, \frac{1}{2}, t^2) + j\sqrt{2}\Gamma(1+\frac{m}{2})\Phi(\frac{1-m}{2}, \frac{3}{2}, t^2)t \right) \\ &= \frac{1}{\sqrt{2}}\Gamma(\frac{m+l}{2})\Phi(\frac{1+m}{2}, \frac{1}{2}, -t^2) + j\sqrt{2}\Gamma(1+\frac{m}{2})\Phi(1+\frac{m}{2}, \frac{3}{2}, -t^2)t. \end{aligned} \tag{E.9}$$

Then by using Eq. E.2 and Eq. E.3 and the property of the Gamma function

$$\Gamma(x+1) = x\Gamma(x), \tag{E.10}$$

we have

$$\begin{aligned} jt\varphi_m(t) &= j\frac{1}{\sqrt{2}}\Gamma(\frac{1+m}{2})t\Phi(\frac{1+m}{2}, \frac{1}{2}, -t^2) - \sqrt{2}\Gamma(\frac{2+m}{2})t^2\Phi(\frac{2+m}{2}, \frac{3}{2}, -t^2) \\ &= -j\sqrt{2}\Gamma(\frac{1+m}{2})t \left(\frac{m}{2}\Phi(\frac{1+m}{2}, \frac{3}{2}, -t^2) - \frac{1+m}{2}\Phi(\frac{3+m}{2}, \frac{3}{2}, -t^2) \right) \\ &\quad + \frac{1}{\sqrt{2}}\Gamma(\frac{2+m}{2}) \left(\Phi(\frac{2+m}{2}, \frac{1}{2}, -t^2) - \Phi(\frac{m}{2}, \frac{1}{2}, -t^2) \right) \\ &= -\left(\frac{1}{\sqrt{2}}\Gamma(\frac{2+m}{2})\Phi(\frac{n}{2}, \frac{1}{2}, -t^2) + j\sqrt{2}\Gamma(\frac{1+m}{2})\frac{m}{2}\Phi(\frac{1+m}{2}, \frac{3}{2}, -t^2)t \right) \end{aligned}$$

$$+\left(\frac{1}{\sqrt{2}}\Gamma(\frac{2+m}{2})\Phi(\frac{2+m}{2},\frac{1}{2},-t^2)+j\sqrt{2}\Gamma(\frac{3+m}{2})\Phi(\frac{3+m}{2},\frac{3}{2},-t^2)t\right)$$
$$= -\frac{m}{2}\varphi_{m-1}(t)+\varphi_{m+1}(t). \tag{E.11}$$

Replacing Eq. E.11 into Eq. E.8, we obtain

$$jth_m(t)=\sqrt{\frac{m+1/2}{2}}h_{m+1}(t)-\sqrt{\frac{m^2}{2m-1}}h_{m-1}(t). \tag{E.12}$$

Then we replace t in Eq. E.12 by Eq. E.7. After some manipulations, we have

$$xh_m(x)=-j\sigma\left(\sqrt{m+\frac{1}{2}}h_{m+1}(x)-\sqrt{\frac{m^2}{m-1/2}}h_{m-1}(x)\right). \tag{E.13}$$

Similarly

$$xh_0(x) = -\frac{1}{2}j\sigma(h_1(x)-h_{-1}(x)), \tag{E.14}$$
$$xh_{-m}(x) = j\sigma\left(\sqrt{m+\frac{1}{2}}h_{-(m+1)}(x)-\sqrt{\frac{m^2}{m-1/2}}h_{-(m-1)}(x)\right). \tag{E.15}$$

About the recursive property of H filters about the differential operation, we have,

$$\begin{aligned} h'_m(x) &= \mathcal{F}^{-1}[\mathcal{F}[h'_m(x)]] \\ &= \mathcal{F}^{-1}[jfH_m(f)] \\ &= j\frac{c_m}{c_{m+1}}\mathcal{F}^{-1}[H_{m+1}(f)] \\ &= j\frac{c_m}{c_{m+1}}h_{m+1}(x). \end{aligned} \tag{E.16}$$

Similarly

$$h'_0(x) = j\frac{c_0}{c_1}(h_1(x)-h_{-1}(x)), \tag{E.17}$$
$$h'_{-m}(x) = -j\frac{c_m}{c_{m+1}}h_{-(m+1)}(x). \tag{E.18}$$

Appendix F

Sherman-Morrison-Woodbury Inversion

Given a full-rank $n \times n$ matrix $\mathbf{C}$ which is perturbed by a rank m matrix $\mathbf{UV}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are both $n \times m$ matrices, the Sherman-Morrison-Woodbury formula [30] (page 225) states that the inverse of the perturbed matrix is the sum of the inverse of $\mathbf{C}$ and another rank m matrix, i.e.,

$$(\mathbf{C} + \mathbf{UV}^T)^{-1} = \mathbf{C}^{-1} - \mathbf{C}^{-1}\mathbf{U}(\mathbf{I}_m + \mathbf{V}^T\mathbf{C}^{-1}\mathbf{U})^{-1}\mathbf{V}^T\mathbf{C}^{-1}, \tag{F.1}$$

where $\mathbf{I}_m$ is the $m \times m$ unit matrix. The validity of this inverse can be easily verified by multiplying both sides by $(\mathbf{C} + \mathbf{UV}^T)$.

In a more concise format, we can write the above equation as

$$(\mathbf{C} + \mathbf{UV}^T)^{-1} = \mathbf{C}_1 + \mathbf{U}_1\mathbf{V}_1^T, \tag{F.2}$$

where

$$\mathbf{C}_1 = \mathbf{C}^{-1}, \tag{F.3}$$

$$\mathbf{U}_1 = -\mathbf{C}^{-1}\mathbf{U}(\mathbf{I}_m + \mathbf{V}^T\mathbf{C}^{-1}\mathbf{U})^{-1}, \tag{F.4}$$

$$\mathbf{V}_1 = \mathbf{C}^{-T}\mathbf{V}. \tag{F.5}$$

In our application, because $\mathbf{C}$ is a diagonal matrix, we can compute its inverse $\mathbf{C}_1$ accurately. Therefore, the only source of numerical error is $\mathbf{A} = (\mathbf{I}_m + \mathbf{V}^T\mathbf{C}^{-1}\mathbf{U})^{-1}$. Suppose the error is

$$\mathbf{E} = \mathbf{I}_m - (\mathbf{I}_m + \mathbf{V}^T\mathbf{C}^{-1}\mathbf{U})\mathbf{A}, \tag{F.6}$$

the final error is

$$(\mathbf{C} + \mathbf{U}\mathbf{V}^T)(\mathbf{C}_1 + \mathbf{U}_1\mathbf{V}_1^T) - \mathbf{I} = \mathbf{U}\mathbf{E}\mathbf{V}^T\mathbf{C}^{-1}. \tag{F.7}$$

Eq. F.7 shows that there is a potential danger that the error $\mathbf{E}$ could be magnified in the final error. That is the source of fragility of the Sherman-Morrison-Woodbury formula. In our system, we reduce that risk by computing $\mathbf{A}$ in double precision and limiting the magnification factor (roughly the ratio between eigenvalues of $\mathbf{U}\mathbf{V}^T$ and those of $\mathbf{C}$) to be less than 10^4 as we did in Section 6.

Appendix G

Eigen Analysis of Symmetric Outer Products

In this section, we consider eigen analysis and singular value decomposition of a special kind of matrices, i.e., symmetric outer products of two low-rank matrices. Because those matrices are symmetric, the problem of computing eigenvalues and eigenvectors is identical to the problem of singular value decomposition because

$$\mathbf{U}\mathbf{V}^T = \begin{pmatrix} & & & \\ \vec{e}_1 & \vec{e}_2 & \cdots & \vec{e}_m \\ & & & \end{pmatrix} \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_m \end{pmatrix} \begin{pmatrix} & \vec{e}_1^T & \\ & \vec{e}_2^T & \\ & \vdots & \\ & \vec{e}_m^T & \end{pmatrix}, \tag{G.1}$$

where $\mathbf{U}$ and $\mathbf{V}$ are both $n \times m$ ($n >> m$) matrices, $\lambda_i, i = 1, 2, \cdots, m$ are eigenvalues, and $\vec{e}_i, i = 1, 2, \cdots, m$ are normalized eigenvectors.

Theorem G.1 *Suppose* $\lambda_i^0, i = 1, 2, \cdots, m$ *and* $\vec{e}_i^0, i = 1, 2, \cdots, m$ *are eigenvalues and normalized eigenvectors of the* $m \times m$ *matrix* $\mathbf{V}^T\mathbf{U}$*, we have the eigenvalues and eigenvectors of* $n \times n$ *matrix* $\mathbf{U}\mathbf{V}^T$ *as*

$$\lambda_i = \lambda_i^0, \tag{G.2}$$

$$\vec{e}_i = \frac{\mathbf{U}\vec{e}_i^0}{\| \mathbf{U}\vec{e}_i^0 \|}, \tag{G.3}$$

where $\| \mathbf{U}\vec{e}_i^0 \|$ *is the norm.*

The proof of the above theorem is straightforward. Since λ_i^0 and $\vec{e}_i^0$ are an eigenvalue and eigenvector of $\mathbf{V}^T\mathbf{U}$, we have

$$\mathbf{V}^T\mathbf{U}\vec{e}_i^0 = \lambda_i^0\vec{e}_i^0. \tag{G.4}$$

Multiplying both sides by $\mathbf{U}$, we obtain

$$(\mathbf{U}\mathbf{V}^T)\mathbf{U}\vec{e}_i^0 = \lambda_i^0\mathbf{U}\vec{e}_i^0. \tag{G.5}$$

Thus we have the eigenvalue and eigenvector of $\mathbf{U}\mathbf{V}^T$ as λ_i^0 and $\mathbf{U}\vec{e}_i^0$.

Additionally, if the outer product $\mathbf{U}\mathbf{V}^T$ is also positive semi-definite, which is true if it represents covariance, we can rewrite it as

$$\mathbf{U}\mathbf{V}^T = \mathbf{B}\mathbf{B}^T, \tag{G.6}$$

where

$$\mathbf{B} = \begin{pmatrix} & & & \\ \vec{e}_1 & \vec{e}_2 & \cdots & \vec{e}_m \\ & & & \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_1} & & & \\ & \sqrt{\lambda_2} & & \\ & & \ddots & \\ & & & \sqrt{\lambda_m} \end{pmatrix}. \tag{G.7}$$

Appendix H

Kalman Filtering for Large Covariance Matrices

In our system, we assume that the motion is discontinuous, i.e., the current motion is unrelated to previous motions. Under this assumption, *a priori* correlation between the structure and the current motion $\mathbf{C}_p$ in Eq. 5.21 is zero. For simplicity, we will assume $\mathbf{C}_p$ is zero in the following sections, although we also have similar results in situations where this assumption is not true. If $\mathbf{P}_-$ is represented as in Eq. 5.21, after some manipulation, we have

$$\begin{aligned} \mathbf{H}\mathbf{P}_-\mathbf{H}^T + \mathbf{R} &= (\mathbf{S}\mathbf{C}_s\mathbf{S}^T + \mathbf{R}) + \begin{pmatrix} \mathbf{A}\mathbf{C}_m & \mathbf{S}\mathbf{U} \end{pmatrix} \begin{pmatrix} \mathbf{A}^T \\ \mathbf{V}^T\mathbf{S}^T \end{pmatrix} \\ &= \mathbf{C}_1 + \mathbf{U}_1\mathbf{V}_1^T, \end{aligned} \tag{H.1}$$

where $\mathbf{C}_1 = (\mathbf{S}\mathbf{C}_s\mathbf{S}^T + \mathbf{R})$ is an $N \times N$ diagonal block matrix with each block a 2×2 matrix, $\mathbf{U}_1$ and $\mathbf{V}_1$ are $2N \times (k+6)$ matrices as

$$\mathbf{U}_1 = \begin{pmatrix} \mathbf{A}\mathbf{C}_m & \mathbf{S}\mathbf{U} \end{pmatrix}, \mathbf{V}_1 = \begin{pmatrix} \mathbf{A} & \mathbf{S}\mathbf{V} \end{pmatrix}. \tag{H.2}$$

By applying the Sherman-Morrison-Woodbury formula as in [30] and Appendix F, we can invert the above matrix

$$(\mathbf{H}\mathbf{P}_-\mathbf{H}^T + \mathbf{R})^{-1} = (\mathbf{C}_1 + \mathbf{U}_1\mathbf{V}_1^T)^{-1} = \mathbf{C}_2 + \mathbf{U}_2\mathbf{V}_2^T, \tag{H.3}$$

where $\mathbf{C}_2$ is also an $N \times N$ diagonal block matrix with each block a 2×2 matrix, $\mathbf{U}_2$ and $\mathbf{V}_2$ are $2N \times (k+6)$ matrices.

Substituting Eq. H.3 back into Eq. 5.13, we obtain the Kalman gain

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_m \\ \mathbf{C}_3 + \mathbf{U}_3\mathbf{V}_3^T \end{pmatrix}, \tag{H.4}$$

where $\mathbf{K}_m$ is a $6 \times 2N$ matrix

$$\mathbf{K}_m = \mathbf{C}_m\mathbf{A}^T\mathbf{C}_2 + \mathbf{C}_m\mathbf{A}^T\mathbf{U}_2\mathbf{V}_2^T, \tag{H.5}$$

$\mathbf{C}_3$ is an $N \times N$ diagonal block matrix with each block a 1×2 matrix

$$\mathbf{C}_3 = \mathbf{C}_s\mathbf{S}^T\mathbf{C}_2, \tag{H.6}$$

$\mathbf{U}_3$ is a $(N+6) \times (2k+6)$ matrix

$$\mathbf{U}_3 = \begin{pmatrix} \mathbf{U} & \mathbf{C}_s\mathbf{S}^T\mathbf{U}_2 \end{pmatrix}, \tag{H.7}$$

and $\mathbf{V}_3$ is a $2N \times (2k+6)$ matrix

$$\mathbf{V}_3 = \begin{pmatrix} (\mathbf{C}_2 + \mathbf{V}_2\mathbf{U}_2^T)\mathbf{S}\mathbf{V} & \mathbf{V}_2 \end{pmatrix}. \tag{H.8}$$

Finally, the updated covariance $\mathbf{P}_+$ is

$$\mathbf{P}_+ = \begin{pmatrix} \mathbf{C}_{mp} & \mathbf{C}_{pp}^T \\ \mathbf{C}_{pp} & (\mathbf{C}_4 + \mathbf{U}_4\mathbf{V}_4^T) \end{pmatrix}, \tag{H.9}$$

where $\mathbf{C}_{mp}$ is a 6×6 *a posteriori* covariance matrix of the motion parameters

$$\mathbf{C}_{mp} = \mathbf{C}_m - \mathbf{K}_m\mathbf{A}\mathbf{C}_m, \tag{H.10}$$

$\mathbf{C}_{pp}$ is an $N \times 6$ *a posteriori* uncertainty correlation between the structure and current motion[1]

$$\mathbf{C}_{pp} = -(\mathbf{C}_s + \mathbf{U}\mathbf{V}^T)\mathbf{S}^T\mathbf{K}_m^T, \tag{H.11}$$

[1]Note that we assumed *a priori* correlation between structure and motion $\mathbf{C}_p$ is zero. But the *a posteriori* correlation $\mathbf{C}_{pp}$ is not zero.

$\mathbf{C}_4$ is an $N \times N$ diagonal matrix representing the independent uncertainty in the structure estimation

$$\mathbf{C}_4 = \mathbf{C}_s - \mathbf{C}_3\mathbf{S}\mathbf{C}_s, \tag{H.12}$$

and $\mathbf{U}_4$ and $\mathbf{V}_4$ are $N \times (3k+6)$ matrices, whose outer product represents the correlated uncertainty in the structure information

$$\mathbf{U}_4 = \begin{pmatrix} (\mathbf{I} - \mathbf{C}_3\mathbf{S} - \mathbf{U}_3\mathbf{V}_3^T\mathbf{S})\mathbf{U} & -\mathbf{U}_3 \end{pmatrix}, \tag{H.13}$$

$$\mathbf{V}_4 = \begin{pmatrix} \mathbf{V} & \mathbf{C}_s\mathbf{S}^T\mathbf{V}_3 \end{pmatrix}. \tag{H.14}$$

If we are careful about the ordering of matrix multiplications in the above equations, we then have an algorithm which updates the state vector and its covariance using $O(kN)$ computation and memory.

Bibliography

[1] M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* U.S. Government Printing Office, Washington, DC, 1972.

[2] Edward H. Adelson and James R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of Optical Society of America, A*, pages 284–299, 1984.

[3] Golad Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):384–401, July 1985.

[4] Golad Adiv. Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):477–489, May 1989.

[5] P. Anandan. Computing dense displacement fields with confidence measures in scenes containing occlusion. In *Proc. ARPA Image Understanding Workshop*, pages 236–246, 1985.

[6] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, pages 283–310, 1989.

[7] A. Azarbayejani and Alex Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):562–575, June 1995.

[8] Ali Azarbayejani, Bradley Horowitz, and Alex Pentland. Recursive estimation of structure and motion using relative orientation constraints. In *Proc. Computer Vision and Pattern Recognition*, pages 294–299, 1993.

[9] Edward J. Baranoski. Triangular factorization of inverse data covariance matrices. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 2245–2248, Toronto, Canada, May 1991.

[10] S. T. Barnard and W. B. Thompson. Disparity analysis of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(4):333–340, 1980.

[11] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. System and experiment: Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[12] V. Michael Bove, Jr. Discrete Fourier transform based depth-from-focus. In *Digest of OSA Topical Meeting on Image Understanding and Machine Vision*, pages 118–121, 1989.

[13] V. Michael Bove, Jr. *Synthetic Movies Derived from Multi-Dimensional Image Sensors.* PhD thesis, MIT, 1989.

[14] Alan C. Bovik, Marianna Clark, and Wilson S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 12(1):55–73, January 1990.

[15] Phil. Brodatz. *Textures: A Photographic Album for Artists and Designers.* New York: Dover, 1966.

[16] Ted J. Broida, S. Chandrashekhar, and Rama Chellappa. Recursive 3-D motion estimation from a monocular image sequence. *IEEE Transactions on Aerospace and Electronic Systems*, 26(4):639–656, July 1990.

[17] Anna R. Bruss and Berthold K. P. Horn. Passive navigation. *Computer Vision, Graphics and Image Processing*, 21:3–20, 1983.

[18] John J. Craig. *Introduction to Robotics: Mechanics and Control.* Addison-Wesley Publishing Company, Inc., second edition, 1989.

[19] John G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of Optical American Society A*, 2(7):1160–1168, 1985.

[20] John Ens. *An Investigation of Methods For Determining Depth From Focus.* PhD thesis, University of British Columbia, 1990.

[21] John Ens and Peter Lawrence. A matrix based method for determining depth from focus. In *Proceedings of Computer Vision and Pattern Recognition*, pages 600–606, 1991.

[22] John Ens and Peter Lawrence. An investigation of methods for determining depth from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(2):97–108, 1993.

[23] Haian Fang and Joseph H. Nurre. Analysis of three dimensional point position for skewed axes stereo vision systems. In *Proc. SPIE Vol. 1832: Vision Geometry*, pages 256–265, 1992.

[24] David Fleet and Allan Jepson. Stability of phase information. In *Proc. of IEEE Workshop on Visual Motion*, pages 52–60, Princeton, New Jersey, October 1991.

[25] David J. Fleet. *Measurement of Image Velocity.* Kluwer Academic Publisher, Norwell, MA, 1992.

[26] David J. Fleet and Allan D. Jepson. Computation of normal velocity from local phase information. In *Proceedings of Computer Vision and Pattern Recognition*, pages 379–386, 1989.

[27] David J. Fleet, Allan D. Jepson, and Michael Jenkin. Phase-based disparity measurement. *CVGIP: Image Understanding*, 53(2):198–210, 1991.

[28] D. Gabor. Theory of communication. *Journal of the IEE*, 93:429–457, 1946.

[29] Arthur Gelb, editor. *Applied Optimal Estimation.* The MIT Press, 1989.

[30] Gene Golub and James M. Ortega. *Scientific Computing: An Introduction with Parallel Computing.* Academic Press, Boston, 1993.

[31] W. E. L. Grimson. A computer implementation of a theory of human stereo vision. *Philosophical Transactions of Royal Society of London*, B292:217–253, 1981.

[32] W Eric L. Grimson. Computational experiments with a feature-based stereo algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(1):17–34, Jan 1985.

[33] William W. Hager. Updating the inverse of a matrix. *SIAM Review*, 31(2):221–239, June 1989.

[34] David J. Heeger. Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, pages 279–302, January 1988.

[35] David J. Heeger and Allan D. Jepson. Subspace methods for recovering rigid motion I: Algorithm and implementation. *International Journal of Computer Vision*, 7(2):95–117, 1992.

[36] Joachim Heel. Temporally integrated surface reconstruction. In *Proceedings of International Conference on Computer Vision*, pages 292–295, 1990.

[37] Anil K. Jain and Farshid Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[38] Allan D. Jepson and Michael R. M. Jenkin. The fast computation of disparity from phase differences. In *Proceedings of Computer Vision and Pattern Recognition*, pages 398–403, 1989.

[39] David G. Jones. *Computational Models of Binocular Vision.* PhD thesis, Dept. of Computer Science, Stanford University, 1991.

[40] David G. Jones and Jitendra Malik. A computational framework for determining stereo correspondence from a set of linear spatial filters. In *European Conference on Computer Vision*, pages 395–410, 1992.

[41] Takeo Kanade and Masatoshi Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. In *DARPA Image Understanding Workshop*, pages 383–398, 1990.

[42] Takeo Kanade and Masatoshi Okutomi. A stereo matching algorithm with an adaptive windows: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, September 1994.

[43] Kenichi Kanatani. Structure and motion from optical flow under orthographic projection. *Computer Vision, Graphics, and Image Processing*, 35:181–199, 1986.

[44] Kenichi Kanatani. Structure and motion from optical flow under perspective projection. *Computer Vision, Graphics, and Image Processing*, 36:122–146, 1987.

[45] John Krumm and Steven A. Shafer. Local spatial frequency analysis for computer vision. Technical Report CMU-RI-TR-90-11, The Robotics Institute, Carnegie Mellon University, 1990.

[46] John Krumm and Steven A. Shafer. Shape from periodic texture using the spectrogram. In *Proceedings of Computer Vision and Pattern Recognition*, pages 284–289, 1992.

[47] John Krumm and Steven A. Shafer. Segmenting textureed 3D surfaces using the space/frequency representation. Technical Report CMU-RI-TR-93-14, The Robotics Institute, Carnegie Mellon University, 1993.

[48] K. Langley, T. J. Atherton, R. G. Wilson, and M. H. E. Larcombe. Vertical and horizontal disparities from phase. In *European Conference on Computer Vision*, pages 315–325, 1990.

[49] H. C. Longuet-Higgens. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, September 1981.

[50] H. C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Philosophical Transactions of Royal Society of London*, B-208:385–397, 1980.

[51] B. Lucas. *Generalized Image Matching by the Method of Differences.* PhD thesis, Department of Computer Science, Carnegie Mellon University, 1984.

[52] Bruce Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of DARPA Image Understanding Workshop*, pages 121–130, 1981.

[53] Jitendra Malik and Ruth Rosenholtz. A differential method for computing local shape-from-texture for planar and curved surfaces. In *Proceedings of Computer Vision and Pattern Recognition*, pages 267–273, 1993.

[54] R. Manmatha and John Oliensis. Extracting affine deformations from image patches I: Finding scale and rotation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 754–755, 1993.

[55] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proceedings of Royal Society London*, B204:301–328, 1979.

[56] Larry Matthies, Richard Szeliski, and Takeo Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–236, 1989.

[57] J. E. W. Mayhew and J. P. Frisby. Psychophysics and computational studies towards a theory of human stereopsis. *Artificial Intelligence*, 17:349–385, 1981.

[58] Amar Mitiche. Three-dimensional space from optical flow correspondence. *Computer Vision, Graphics, and Image Processing*, 42:306–317, 1988.

[59] H. P. Moravec. Towards automatic visual obstacle avoidance. In *Proceedings of 5th International Joint Conference on Artificial Intelligence*, page 584, 1977.

[60] Kenichi Mori, Masatsugu Kidode, and Haruo Asada. An iterative prediction and correction method for automatic stereocomparison. *Computer Graphics and Image Processing*, 2:393–401, 1973.

[61] Hans-Hellmut Nagel. Displacement vectors derived from second-order intensity variations in image sequences. *Computer Graphics and Image Processing*, 21:85–117, 1983.

[62] Hans-Hellmut Nagel. Image sequences - ten (octal) years - from phenomenology towards a theoretical foundation. In *Proceedings of the Eighth International Conference on Pattern Recognition*, pages 1174–1185, October 1986.

[63] Hans-Hellmut Nagel. On the estimation of optical flow: Relations between different approaches and some new results. *Artificial Intelligence*, 33:299–324, 1987.

[64] Mark Nardin, W.F. Perger, and Atul Bhalla. Algorithm 707 CONHYP: A numerical evaluator of the confluent hypergeometric function for complex arguments of large magnitudes. *ACM Transactions on Mathematical Software*, 18(3):345–349, 1992.

[65] Mark Nardin, W.F. Perger, and Atul Bhalla. Numerical evaluation of the confluent hypergeometric function for complex arguments of large magnitudes. *Journal of Computational and Applied Mathematics*, 39:193–200, 1992.

[66] Shree Nayar, Masahiro Watanabe, and Minori Noguchi. Real-time focus range sensor. In *Proc. International Conference on Computer Vision*, pages 995 – 1001, 1995.

[67] Arnold F. Nikiforov and Vasilii B. Uvarov. *Special Functions of Mathematical Physics*. Birkhauser Verlag Basel, 1988.

[68] A. Pentland, T. Darrell, M. Turk, and W. Huang. A simple, real-time range camera. In *Proceedings of Computer Vision and Pattern Recognition*, pages 256–261, 1989.

[69] Alex P. Pentland. A new sense for depth of field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):523–531, 1987.

[70] S. B. Pollard and J. P. Frisby J. E. W. Mayhew. PMF: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1981.

[71] K. Prazdny. Egomotion and relative depth map from optical flow. *Biological Cybernetics*, 36:87–102, 1980.

[72] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.

[73] James M. Rehg and Andrew P. Witkin. Visual tracking with deformation model. In *Proceedings of International Conference on Robotics and Automation*, pages 844–850, 1991.

[74] Olivier Rioul and Martin Vetterli. Wavelets and signal processing. *IEEE Signal Processing Magazine*, pages 14–38, October 1991.

[75] Luc Robert and Martial Hebert. Deriving orientation cues from stereo images. In *European Conference on Computer Vision*, pages 377–388, 1994.

[76] T. D. Sanger. Stereo disparity computation using Gabor filters. *Biological Cybernetics*, 59:405–418, 1988.

[77] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proc. Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[78] A. Singh. An estimation-theoretic framework for image-flow computation. In *Proceedings of Third International Conference on Computer Vision*, pages 168–177, 1990.

[79] A. Singh. *Optical Flow Computation: A Unified Perspective.* IEEE Computer Society Press, 1992.

[80] Randall C. Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainty. *The International Journal of Robotics Research*, 5(4):56–68, 1986.

[81] Jerome Spanier and Keith B. Oldham. *An Atlas of Functions.* Washington: Hemisphere Pub. Corp., 1987.

[82] Murali Subbarao. Parallel depth recovery by changing camera parameters. In *2nd International Conference on Computer Vision*, pages 149–155, 1988.

[83] Murali Subbarao and Gopal Surya. Depth from defocus: A spatial domain approach. Technical Report 92.12.03, Department of Electrical Engineering, State University of New York at Stony Brook, 1992.

[84] Murali Subbarao and Tse-Chung Wei. Depth from defocus and rapid autofocusing: A practical approach. In *Proceedings of Computer Vision and Pattern Recognition*, pages 773–776, 1992.

[85] Richard Szeliski and Sing Bing Kang. Recovering 3D shape and motion from image streams using nonlinear least squares. *Journal of Visual Communication and Image Representation*, 5(1):10–28, March 1994.

[86] J. Inigo Thomas and J. Oliensis. Incorporating motion error in multi-frame structure from motion. Technical Report COINS TR91-36, Computer and Information Science, University of Massachusetts at Amherst, 1991.

[87] Carlo Tomasi. *Shape and Motion from Image Streams: a Factorization Method.* PhD thesis, The School of Computer Science, Carnegie Mellon University, 1991.

[88] Carlo Tomasi. Shape and motion from image streams: A factorization method. Technical Report CMU-CS-91-172, The School of Computer Science, Carnegie Mellon University, 1991.

[89] Roger Y. Tsai and Thomas S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(1):13–27, January 1984.

[90] S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. *Biological Cybernatics*, 60:79–97, 1988.

[91] Juyang Weng. A theory of image matching. In *International Conference on Computer Vision*, pages 200–209, 1990.

[92] Juyang Weng. Image matching using the windowed Fourier phase. *International Journal of Computer Vision*, 11(3):211–236, December 1993.

[93] Juyang Weng, Narendra Ahuja, and Thomas Huang. Closed-form solution + maximum likelihood: A robust approach to motion and structure estimation. In *Proceedings of Computer Vision and Pattern Recognition*, pages 381–386, 1988.

[94] Juyang Weng, Narendra Ahuja, and Thomas Huang. Two-view matching. In *International Conference on Computer Vision*, pages 64–73, 1988.

[95] Juyang Weng, Thomas Huang, and Narendra Ahuja. Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):451–476, May 1989.

[96] Peter Whaite and Frank P. Ferrie. From uncertainty to visual exploration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1038–1049, 1991.

[97] Reg G. Willson and Steven A. Shafer. Precision imaging and control for machine vision research at Carnegie Mellon University. Technical Report CMU-CS-92-118, School of Computer Science, Carnegie Mellon University, 1992.

[98] Yalin Xiong and Steven Shafer. Depth from focusing and defocusing. Technical Report CMU-RI-TR-93-07, The Robotics Institute, Carnegie Mellon University, 1993.

[99] Yalin Xiong and Steven A. Shafer. Depth from focusing and defocusing. In *Proceedings of Computer Vision and Pattern Recognition*, pages 68–73, 1993.

[100] Yalin Xiong and Steven A. Shafer. Moment and hypergeometric filters for high precision computation of focus, stereo, and optical flow. Technical Report CMU-RI-TR-94-28, The Robotics Institute, Carnegie Mellon University, 1994.

[101] Yalin Xiong and Steven A. Shafer. Variable window Gabor filters and their use in focus and correspondence. In *Proc. Computer Vision and Pattern Recognition*, pages 668–671, 1994.

[102] Yalin Xiong and Steven A. Shafer. Dense structure from a dense optical flow sequence. Technical Report CMU-RI-TR-95-10, The Robotics Institute, Carnegie Mellon University, 1995.

[103] Yalin Xiong and Steven A. Shafer. Dense structure from a dense optical flow sequence. In *Proceedings of International Symposium on Computer Vision*, Coral Gables, FL, November 1995. to appear.

[104] Yalin Xiong and Steven A. Shafer. Hypergeometric filters for optical flow and affine matching. In *International Conference on Computer Vision*, pages 771–776, 1995.