Moment and Hypergeometric Filters For High Precision Computation of Focus, Stereo and Optical Flow

Yalin Xiong Steven A. Shafer

CMU-RI-TR-94-28

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

September, 1994

A short version of this technical report will appear in the *Proceedings of Image Understanding Workshop 1994* under the title "*Recursive Filters For High Precision Computation of Focus, Stereo and Optical Flow*"

Contents

List of Figures

List of Tables

Abstract

Many low level visual computation problems such as focus, stereo, optical flow, etc. can be formulated as problems of extracting one or more parameters of a non-stationary transformation between two images. Because of the non-stationary nature, finite-width windows are widely used in various algorithms to extract spatially local information from images. While the choice of window width has a very profound impact on the quality of results of algorithms, there has been no quantitative way to measure or eliminate the negative effects of finite-width windows. To address this problem, We introduce two sets of filters, "moment" filters and "hypergeometric" filters. Due to their recursive properties, these filters allow the effects of finite-width windows and foreshortening to be explicitly analyzed and eliminated.

We develop one paradigm to solve general one-parameter extraction problems using moment filters, and another one to solve general multiple parameter extraction problem using hypergeometric filters. We apply these paradigms to problems of focus and stereo, in which one parameter is extracted at every pixel location, and optical flow, in which two parameters are extracted. We demonstrate that our algorithms based on moment filters and hypergoemetric filters achieve much higher precision than other techniques.

Keywords: *Focus, Stereo, Optical Flow, Gabor Filter, Moment Filter, Hypergeometric Filter, Low-level Processing, Computer Vision, Image Processing.*

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 very profound impact on the quality of results of algorithms, there has been no quantitative way to measure or eliminate the negative effects of finite-width windows. We propose two sets of filters, "moment" filters and "hypergeometric" filters. Due to their recursive properties, the effects of finite-width windows can be explicitly analyzed and eliminated.

Many low level visual computing problems such as focus, stereo, optical flow, etc. can be formulated as problems of extracting one or more parameters of a "nonstationary" transformation between two images. By nonstationary we mean that the parameters are position dependent in the image. For example, we can formulate a general correspondence problem as one in which we extract disparity as one parameter in a position-dependent phase-shifting transformation. The nonstationary nature of the parameters forces us to use finite-width filters for extracting 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 implicitly combining the outputs of finite-width filters in the spatial domain (e.g. SSD in stereo), the effects of finite width are seldom analyzed. Most of times, the information extracted from finite-width filters are regarded as approximations of those from infinite-width filters. But such approximations make the computations based on the extracted information 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 nonstationary transform in the Fourier domain can be an arbitrary curve (or a surface) within the passband, the approximation in ignoring the effects of finite width is equivalent to using a constant to approximate the arbitrary curve in the passband. The moment filters and hypergeometric 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 parameters based upon the polynomial approximation is much more accurate.

We apply the technique of moment filters to the problems of focus and stereo. Both the focus problem and stereo problem are examples of the general one-parameter extraction problem. Traditionally, people estimate the parameter by comparing the outputs of convolutions of the two images with the same filter, usually a Gabor filter. Our moment filter approach provides a new insight into this traditional approach by showing that the output of the convolution of one image with the Gabor filter can be represented as an infinite sum of the output of convolutions of the other image with moment filters. From this point of view, the traditional 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 parameters by using finite-width filters, people usually assume that the parameters are constant within the width of filters. But another aspect of the nonstationarity is that the parameters actually change within the effective width of the filters, which we refer to as the shift variance problem. Such problem is referred to as foreshortening in stereo, and affine matching problem in 2D image registration. Traditional infinite-width filters are unable to deal with the situation because the infinite-width Fourier transform doesn't converge. The recursive properties of the moment filters allow modeling this effect as a modification of the Gabor filter convolution by convolutions of the first-order moment filters. Applying this technique to the focus and stereo problem, we demonstrate that we can not only eliminate the negative effect on the estimation of the parameter caused by the foreshortening, but also estimate the the degree of foreshortening quantitatively. Such a quantitative measurement provides a new approach to the shift variance problem.

Both the traditional approaches and the moment filter approach suffer from the frequency sampling problem, i.e. there is no way that the frequency domain can be sampled completely and nonredundantly. If the frequency domain is sampled sparsely, much information is abandoned, while if it is sampled densely, the results from different frequency bands are highly correlated, which makes the merge of the results difficult. Extending the idea of the moment filters, we developed another set of filters, "hypergeometric filters". The advantage of the hypergeometric filters is that they provide a complete and non-redundant decomposition of the local signal, even though they sample the frequency domain in a nonuniform fashion. Based on hypergeometric filters, the general problem of extracting parameters of nonstationary transformation can be formulated as a multidimensional minimization problem. Like the moment filters, the hypergeometric filters also have recursive properties in both the spatial and frequency domain, and therefore, are also capable of extracting parameters with high precision and dealing with the foreshortening problem.

We apply the hypergeometric filter technique to the optical flow problem, which can be formulated as extracting two parameters from a nonstationary transformation between two images. A 2D conjugate gradient minimization algorithm is employed to compute the optical flow between two adjacent frames. Not only can this algorithm produce very precise estimation of the optical flow, but also a meaningful error estimation, which represents the aperture problem as covariance matrices, is computed based on the minimization. Finally, we quantitatively compare the performance of the hypergeometric filter approach with other popular techniques.

2 Related Research

The paradigm of extracting parameters by convolving images with filters which are local in both the spatial and frequency domain, has been successfully used in many low-level vision tasks, such as, motion analysis, stereo matching, texture analysis, and focus measure in literature. Adelson and Bergen [2] and Heeger [10] modeled motions in 2D image space as orientations in 3D spatiotemporal space by introducing 3D oriented filters to measure image velocities. More recently, Fleet and Jepson [7] modeled 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, Weng [27], Sanger [23], Fleet et al. [8], and Langley et al. [16] proposed using filters to extract phase information, then computed disparities. Jones and Malik [12], on the other hand, applied a set of linear spatial filters to images and used responses from those filters as matching features. Because the focus difference between two images can be represented as a blurring convolution, many researchers have also proposed using spatial frequency analysis to compute the focus difference [29, 4, 26, 21]. The spatial frequency approach also achieved great success in texture segmentation [5, 11, 15] and shape recovery from texture [14, 17].

The problem with finite-width windows mentioned in the previous section appears in every application. Realizing finite-width filters cause errors and instability, some researchers found ways to deal with this problems in literature. One way is to assume that there are one or more dominating frequency components in images. Then around the dominating frequencies, the effects of finite width are negligible. Though this approach is effective when the assumption is valid, ordinary images usually don't contain any dominating components. Another way proposed in literature is using two techniques, a stability criterion for detecting the situations where the filter output is dominated by window contamination [6, 30] and instantaneous frequency as the spatial derivative of phase. In relating to our approaches proposed in this report, both techniques are actually special cases of the moment filter approach. Both the stability criterion and instantaneous frequency can be directly computed from the first order moment filter. From this point of view, the moment filter approach generalizes these two techniques.

Another common practice in literature in using finite-width filters is that low frequency components are usually abandoned due to two reasons. One is that the low frequency components are usually contaminated by the DC component because Gabor filters have non-zero DC bias, and the DC component is usually very strong. The other reason is that the negative effects of finite width is much severer when the frequency is low than that when the frequency is high. Unlike Gabor filters, the hypergeometric filter contains no DC component as long as the peak frequency is not zero. And since we eliminate the effects of finite-width windows, the low frequency components can also be used to compute high precision results. We will show that the hypergeometric filter approach provides a canonical way of sampling the frequency domain so that all information is gathered, including low frequency components which actually contain surprisingly rich information.

3 Moment Filter Approach

In this section, we will propose the "moment filter" decomposition of the signal and demonstrate how this decomposition can be effectively applied to problems of focus and stereo. When we apply such a decomposition to problems of focus and stereo, we show that the effects of finite width can be represented by high order moment filters though Taylor's expansion of the transformation. Furthermore, we are able to solve the foreshortening problem in both cases by using the spatially recursive property of moment filters, while the traditional Fourier analysis fails because the Fourier integral doesn't converge.

3.1 Theory of Moment Filters

3.1.1 Definitions of Moment Filters

We define the ith order moment filter in the spatial domain as

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

in which

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

where n is an integer, j is $\sqrt{-1}$, and $L(n,\alpha,x)$ is the generalized Laguerre polynomial [20], with $L(0,c,x) = 1$. Figure 1 illustrates some of the moment filters in the spatial domain.

Apparently the zero-order moment filter is the Gabor filter $G(x;f_0)$ with the peak frequency f_0:

$$G(x;f_0) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}}e^{-jf_0x}. \tag{3}$$

In the Fourier domain, the moment filter can be defined as

$$M_i(f) = \mathcal{F}[m_i(x)] = (f-f_0)^i e^{-(f-f_0)^2\sigma^2/2}. \tag{4}$$

Therefore, the peak frequency of the ith moment filter is

$$\text{Peak}[M_i(f)] = f_0 \pm \frac{\sqrt{i}}{\sigma}, \tag{5}$$

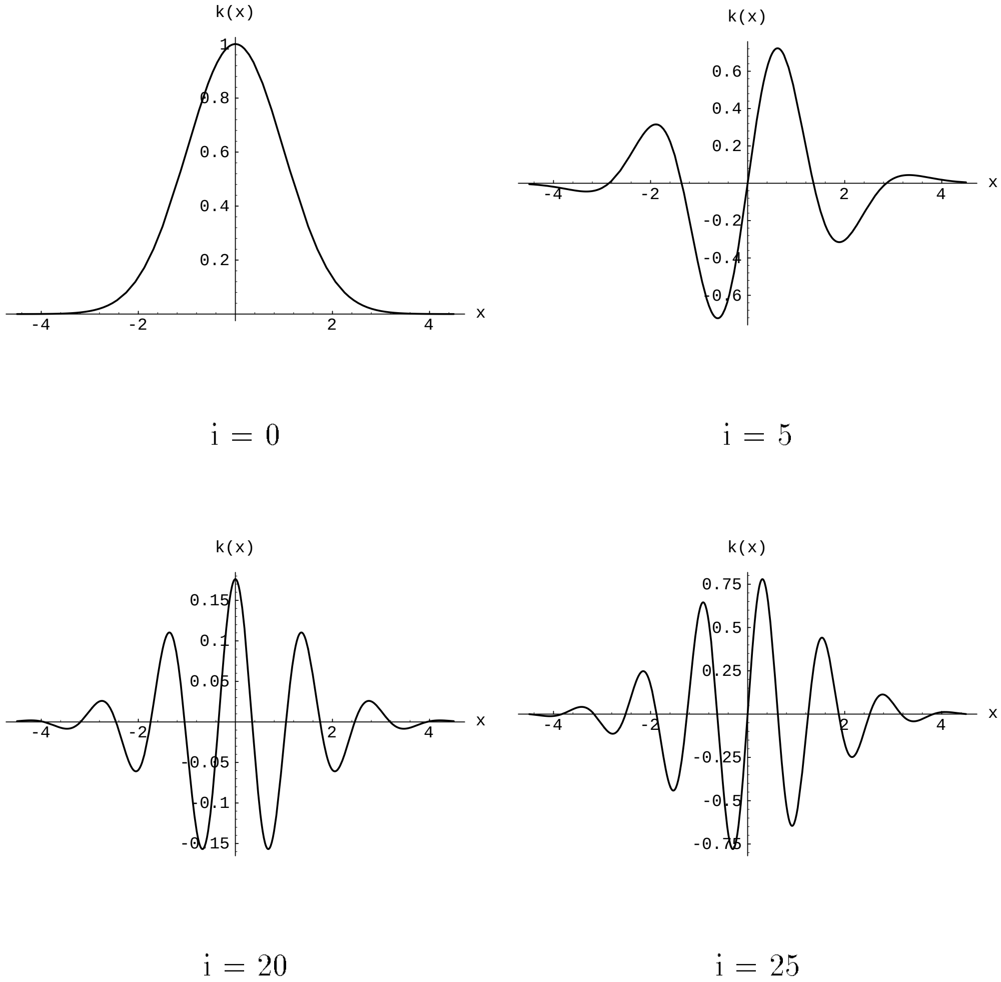
which moves away from the original peak frequency as i increases. Figure 2 illustrates the pass bands of moment filters as i increases, assuming $f_0 = 0$. Note that only zero-order moment filter has one single peak, all others have two peaks.

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 $g(x)$:

$$\mathcal{F}[\frac{d^n}{dx^n}g(x)] = (jf)^n\mathcal{F}[g(x)] = j^n f^n e^{-f^2\sigma^2/2}. \tag{6}$$

From Eq. 4 and Eq. 1, we then have

$$m_i(x) = j^{-i}e^{-jf_0x}\frac{d^i}{dx^i}g(x). \tag{7}$$



i = 0

i = 5

i = 20

i = 25

Figure 1: The Moment Filters in The Spatial Domain

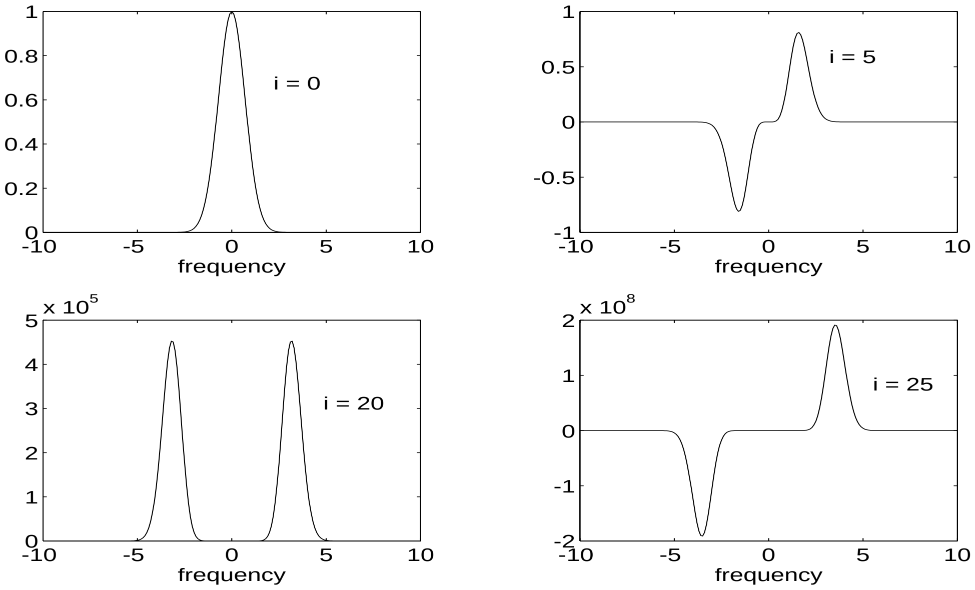


Figure 2: The Pass Band Moves as i Increases

3.1.2 Decomposition by Moment Filters

In this section, we are considering 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

$$k_i(x) = p_i^*(x)e^{jf_0x}, \tag{8}$$

and the definition of an inner production over the function space as

$$< g(x), h(x) >= \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{+\infty} g(x)h^*(x)e^{-\frac{x^2}{2\sigma^2}}dx, \tag{9}$$

where $*$ denotes complex conjugate, we then have

1. The functions k_i in Eq. 8 are orthogonal with each other under the definition of the inner production, i.e.

$$< k_i(x), k_j(x) >= 0, \text{when} i \neq j. \tag{10}$$

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

$$\begin{aligned} w_i &= < \xi(x), k_i(x) >, & (11)\\ \xi(x) &= \sum_{i=0}^{+\infty} \frac{w_i}{< k_i(x), k_i(x) >} k_i(x). & (12)\end{aligned}$$

The proof can be found in Appendix A.

Let us look into the inner production in Eq. 11,

$$\begin{aligned} w_i &= \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{+\infty}\xi(x)k_i^*(x)e^{-\frac{x^2}{2\sigma^2}}dx \\ &= \int_{-\infty}^{+\infty}\xi(x)p_i(x)\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}}e^{-jf_0x}dx \\ &= \int_{-\infty}^{+\infty}\xi(x)m_i(x)dx, \end{aligned} \tag{13}$$

This equation can also be represented in the Fourier domain as,

$$\begin{aligned} w_i &= \mathcal{F}[\xi(x)]\otimes\mathcal{F}[p_i(x)\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}}] \\ &= \int_{-\infty}^{+\infty}\mathcal{F}[\xi(x)](f_0-f)^ie^{-(f_0-f)^2\sigma^2/2}df \\ &= (-1)^i\int_{-\infty}^{+\infty}\mathcal{F}[\xi(x)]M_i(f)df. \end{aligned} \tag{14}$$

In other words, the coefficients w_i from convolutions of the signal with the moment filters can be used to losslessly reconstruct the original signal. Therefore, the decomposition of the original signal by moment filters is complete and non-redundant. Eq. 13 and Eq. 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.

3.1.3 Important Properties of the Moment Filters

We here list the important properties of the moment filters. The proof of these properties can be found in Appendix B.

- They are recursive in the Fourier domain:

$$(f-f_0)M_i(f) = M_{i+1}(f). \tag{15}$$

- They are recursive in the spatial domain:

$$xm_i(x) = j(im_{i-1}(x) - \sigma^2 m_{i+1}(x)), \tag{16}$$

  where $j = \sqrt{-1}$.

- They are recursive with respect to the differential operation:

$$\begin{aligned} m_i'(x) &= j^{-i}\frac{d}{dx}\left(e^{-jf_0x}\frac{d^i}{dx^i}g(x)\right) \\ &= j(m_{i+1}(x) - f_0m_i(x)). \end{aligned} \tag{17}$$

- The instantaneous frequency is defined as the spatial derivative of the phase in [23, 6]. Usually replacing the peak frequency of the Gabor filter by instantaneous frequency will increase the accuracy as claimed in [6]. The instantaneous frequency can also be represented by moment filters as

$$\frac{d}{dx_0}\phi(x_0) = f_0 - \mathrm{Re}\left(\frac{w_1}{w_0}\right), \tag{18}$$

  where Re means the real part, ϕ is the phase, and w_0 and w_1 are coefficients in Eq. 13.

- The stability criterion is proposed by Fleet et al. in [6] to locate where the information extracted by finite-width filters is substantially different from those 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

$$T = \| \frac{1}{a}\frac{da}{dx_0} \|^2 + \| \frac{d\phi}{dx_0} - f_0 \|^2 = \frac{\| w_1 \|^2}{\| w_0 \|^2}. \tag{19}$$

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 *ad hoc* utilization of information from the first-order moment filter.

3.2 Moment Filters For Focus

3.2.1 Problem Definition

The method of depth from defocus is to 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 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:

$$\begin{aligned} i_1(x) &= i_0(x) \otimes \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{x^2}{2\sigma_0^2}} = \int_{-\infty}^{+\infty} i_0(t) \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x-t)^2}{2\sigma_0^2}} dt && (20) \\ I_1(f) &= I_0(f) e^{-f^2\sigma_0^2/2} && (21) \end{aligned}$$

The difference of focus between two images is characterized by σ_0. If σ_0 is not a constant within the window but changes linearly in neighborhood of window center x_0 (referred as $\epsilon(x_0)$), i.e

$$\sigma_0 = s(1 + \mu x), x \in \epsilon(x_0), \tag{22}$$

we can expand the second integral term in Eq. 20 as follow:

$$\frac{1}{\sqrt{2\pi}\sigma_0}e^{-\frac{(x-t)^2}{2\sigma_0^2}} \approx \frac{1}{\sqrt{2\pi}s}e^{-\frac{(x-t)^2}{2s^2}} + \mu x \frac{(x-t)^2 - s^2}{\sqrt{2\pi}s^3}e^{-\frac{(x-t)^2}{2s^2}}, x \in \epsilon(x_0), \tag{23}$$

where we truncated terms of higher order of μx based upon the assumption that μ is small.

Now we need to solve the following two problems:

- *Finite Window Problem*: Since Eq. 21 is based upon an infinite window, how can we compute σ_0 when the window is actually finite?
- *Shift Variance Problem*: Since Eq. 21 is based on the assumption that σ_0 is a constant within the window, how can we compute s in Eq. 22 when σ_0 is not a constant, namely $\mu \neq 0$?

Another way to understand Eq. 22 and 23 is that s represents the blurring difference at the pixel and μ the gradient of the blurring difference at the pixel. When the blurring difference s is converted into depth, the gradient μ is converted into the surface gradient, or surface tilt. 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.

3.2.2 Finite Window Problem

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

$$\begin{aligned} U_i &= \int_{-\infty}^{+\infty} i_0(x)m_i(x)dx \\ &= (-1)^i \int_{-\infty}^{+\infty} I_0(f)M_i(f)df, \end{aligned} \tag{24}$$

$$\begin{aligned} V_i &= \int_{-\infty}^{+\infty} i_1(x)m_i(x)dx \\ &= (-1)^i \int_{-\infty}^{+\infty} I_1(f)M_i(f)df \\ &= (-1)^i \int_{-\infty}^{+\infty} I_0(f)e^{-f^2\sigma_0^2/2}M_i(f)df. \end{aligned} \tag{25}$$

By Taylor's expansion, in the neighborhood of f_0, i.e. $\epsilon(f_0)$, we have,

$$e^{-f^2\sigma_0^2/2} \approx e^{-f_0^2\sigma_0^2/2}\left(1 - f_0\sigma_0^2(f-f_0) - \frac{\sigma_0^2(1-f_0^2\sigma_0^2)}{2}(f-f_0)^2\right), f \in \epsilon(f_0), \tag{26}$$

where higher order expansions are truncated.

Replacing Eq. 26 into Eq. 25, and using the recursive property of moment filters in the Fourier domain, we obtain,

$$\begin{aligned} V_0 &\approx \int_{-\infty}^{+\infty} I_0(f)\left(1 - f_0\sigma_0^2(f-f_0) - \frac{\sigma_0^2(1-f_0^2\sigma_0^2)}{2}(f-f_0)^2\right) e^{-(f_0-f)^2\sigma^2/2} df \\ & \quad e^{-f_0^2\sigma_0^2/2} \\ &= e^{-f_0^2\sigma_0^2/2}(U_0 + f_0\sigma_0^2 U_1 - \frac{\sigma_0^2(1-f_0^2\sigma_0^2)}{2}U_2). \end{aligned} \tag{27}$$

Therefore, we have the equation to compute σ_0^2,

$$\sigma_0^2 = \frac{2}{f_0^2}\left(\ln(U_0 + f_0\sigma_0^2 U_1 - \frac{\sigma_0^2(1-f_0^2\sigma_0^2)}{2}U_2) - \ln V_0\right). \tag{28}$$

Compared with the equation from Eq. 21, i.e.

$$\sigma_0^2 = \frac{2}{f_0^2}(\ln I_0(f_0) - \ln I_1(f_0)), \tag{29}$$

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

$$\ln\frac{U_0}{V_0} = \ln\frac{I_0(f_0)}{I_1(f_0)}, \tag{30}$$

this may result into substantial errors if either U_1 or U_2 happens to be not very small comparing to U_0.

A more intuitive way of looking at the difference between Eq. 28 and Eq. 29 is illustrated in Figure 3. The solid line is the transformation curve $e^{-f^2\sigma_0^2/2}$, with $\sigma_0 = 1.0$. The peak frequency $f_0 = 0.8$, and the passband of the Gabor filter is $(0.4, 0.8)$. The dotted curve is the zeroth order approximation, i.e. Eq. 29 is equivalent to using a constant to approximate the transformation curve within the passband. The first order approximation (dash-dot curve), i.e. Eq. 28 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. 28, 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. Certainly, it will drastically decrease errors comparing to the constant approximation.

Now, let us look back the truncation in Eq. 26. 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. 27. 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. 13, the expected magnitude of U_i will be proportional to the square root of the energy of the filter:

$$E_i = \left(\int_{-\infty}^{+\infty} \| m_i(x) \|^2 \, dx\right)^{1/2}$$

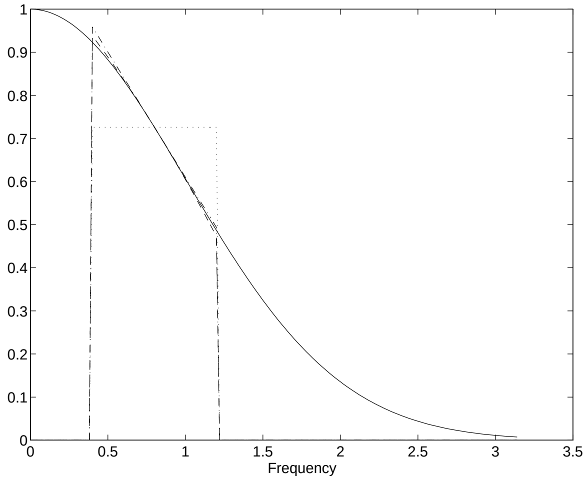


Figure 3: Polynomial Fitting Of The Transformation Curve

$$
\begin{aligned}
&= \left(\frac{1}{2\pi}\int_{-\infty}^{+\infty} \| M_i(f) \|^2 \, df\right)^{1/2} \\
&= \sqrt{\frac{,\,(i+\frac{1}{2})}{2\pi}}\sigma^{-(i+1/2)}, \qquad (31)
\end{aligned}
$$

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

Solving Eq. 28 is not as trivial as solving Eq. 29. But since the expected magnitude of U_0 is larger than those of U_1 and U_2, we propose to use Eq. 28 as an iterative procedure to find the qth time estimation of $\sigma_0^{2(q)}$ such as:

$$
\begin{aligned}
\sigma_0^{2(q)} &= \frac{2}{f_0^2}\left(\ln(U_0 + f_0\sigma_0^{2(q-1)}U_1 - \frac{\sigma_0^{2(q-1)}(1 - f_0^2\sigma_0^{2(q-1)})}{2}U_2) - \ln V_0\right) \\
&= T(\sigma_0^{2(q-1)}). \qquad (32)
\end{aligned}
$$

And the sufficient condition for the iteration to converge is

$$
\| T'(\sigma_0^{2(q)}) \| = \| \frac{2}{f_0^2}\frac{f_0U_1 - \frac{1}{2}(1 - 2f_0^2\sigma_0^{2(q)})U_2}{U_0 + f_0\sigma_0^{2(q)}U_1 - \frac{1}{2}\sigma_0^{2(q)}(1 - f_0^2\sigma_0^{2(q)})U_2} \| < 1. \qquad (33)
$$

For some locations, the convergence condition is not satisfied. And we can think of the convergence condition as a generalized stability criterion. In fact, if we consider only the linear term in the Taylor's expansion, this convergence condition is the same as the stability criterion in [6].

3.2.3 Shift Variance Problem

When $\mu \neq 0$ in Eq. 22 and Eq. 23, we must consider the effect of the shift variance. Let us define,

$$\begin{aligned} V_i &= \int_{-\infty}^{+\infty} i_1(x) m_i(x) dx, & (34) \\ \mathcal{V}_i &= \int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} i_0(t) \frac{1}{\sqrt{2\pi} s} e^{-\frac{(x-t)^2}{2s^2}} dt \right) m_i(x) dx. & (35) \end{aligned}$$

From convolutions of the two images with moment filters, we obtain U_i and V_i. The defined $\mathcal{V}_i$ can not be obtained from convolutions. But to estimate s, we need to compute $\mathcal{V}_i$ first. In other words, we need to virtually rotate back the slanted surface to make it front-parallel, then compute the convolutions $\mathcal{V}_i$ of the rotated surface. Replacing Eq. 23 into Eq. 34, we have

$$\begin{aligned} V_i = & \int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} i_0(t) \frac{1}{\sqrt{2\pi} s} e^{-\frac{(x-t)^2}{2s^2}} dt \right) m_i(x) dx + \\ & \mu \int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} i_0(t) \frac{(x-t)^2 - s^2}{\sqrt{2\pi} s^3} e^{-\frac{(x-t)^2}{2s^2}} dt \right) x m_i(x) dx. \qquad (36) \end{aligned}$$

Using the recursive property of the moment filter in the spatial domain in Eq. 16, and converting the above equation into the Fourier domain, we obtain

$$\begin{aligned} V_i &= \mathcal{V}_i - \mu \int_{-\infty}^{+\infty} I_0(f)(-f^2 s^2 e^{-f^2 s^2/2}) j(\sigma^2 M_{i+1}(f) - i M_{i-1}(f)) df \\ &= \mathcal{V}_i + j s^2 \mu (\sigma^2 (\mathcal{V}_{i+3} + 2 f_0 \mathcal{V}_{i+2} + f_0^2 \mathcal{V}_{i+1}) - \\ &\quad i(\mathcal{V}_{i+1} + 2 f_0 \mathcal{V}_i + f_0^2 \mathcal{V}_{i-1})). \qquad (37) \end{aligned}$$

In other words, the effect of the linear shift variance is that the original filter outputs $\mathcal{V}_i$ when the parameter is constant within the window 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. 37:

$$\begin{aligned} \mathcal{V}_i &\approx V_i - j s^2 \mu (\sigma^2 (V_{i+3} + 2 f_0 V_{i+2} + f_0^2 V_{i+1}) - \\ &\quad i(V_{i+1} + 2 f_0 V_i + f_0^2 V_{i-1}) \\ &= V_i + s^2 \mu P_i, \qquad (38) \end{aligned}$$

i.e. 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 the parameter is a constant within the window. Once we have computed $\mathcal{V}_i$, we can compute s by the iteration proposed in the previous section as

$$s^2 = \frac{2}{f_0^2} \left(\ln(U_0 + f_0 s^2 U_1 - \frac{1}{2} s^2 (1 - f_0^2 s^2) U_2) - \ln(V_0 + s^2 \mu P_0) \right). \qquad (39)$$

Now, the problem is that we don't know μ. Therefore, we also need an algorithm to estimate μ. Fortunately, we can compute μ by taking the ratio of the zeroth and the first moment by using Eq. 27 and Eq. 38:

$$\frac{V_1 + s^2\mu P_1}{V_0 + s^2\mu P_0} \approx \frac{U_1 + f_0 s^2 U_2 - \frac{1}{2}s^2(1 - f_0^2 s^2)U_3}{U_0 + f_0 s^2 U_1 - \frac{1}{2}s^2(1 - f_0^2 s^2)U_2} \tag{40}$$

Combining Eq. 39 and Eq. 40 together, we have two equations for two unknowns, i.e. s and μ. Regarding all the moments as constants in Eq. 39, and assuming the iteration in Eq. 32 indeed converges, we can simplify Eq. 39 as:

$$s^2 = \mathcal{M}(s^2\mu), \tag{41}$$

where $\mathcal{M}$ can be regarded as a complicated function of μ. Similarly, Eq. 40 can be simplified as:

$$s^2\mu = \mathcal{N}(s^2), \tag{42}$$

where $\mathcal{N}$ is a complicated function of s^2.

Eq. 41 and Eq. 42 suggest another iterative scheme to solve both s and μ simultaneously. Let $u = s^2$ and $v = s^2\mu$, and 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:

$$\begin{aligned} u^{(q)} &= \mathcal{M}(v^{(q-1)}) & (43)\\ v^{(q)} &= \mathcal{N}(u^{(q-1)}) & (44) \end{aligned}$$

And the sufficient condition for the iteration to converge is that the magnitudes of both the complex eigenvalues $E_i, i = 0, 1$ of the Jacobian matrix J are less than one, i.e.

$$\parallel E_i \parallel < 1.0, i = 0, 1, \tag{45}$$

where,

$$J = \begin{bmatrix} 0 & \frac{\partial \mathcal{M}}{\partial v^{(q)}} \\ \frac{\partial \mathcal{N}}{\partial u^{(q)}} & 0 \end{bmatrix}. \tag{46}$$

Therefore, when the convergence condition Eq. 45 is satisfied, we can iteratively estimate depth and slope simultaneously as in Eq. 43 and Eq. 44.

3.2.4 Error Estimation

Error estimation is necessary if we have to merge results from different frequency bands. The convergence conditions (Eq. 33 and Eq. 45) can identify serious cases of the weak texture. But for those less serious cases, 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. 39 and Eq. 40 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 encounter the following situation.

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

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

we then have,

$$E[u_r v_r] = E[\frac{A\delta + A^*\delta^*}{2}\frac{B\delta + B^*\delta^*}{2}] = \mathrm{Re}(AB^*)E[\delta\delta^*]/2. \tag{47}$$

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

$$E[\mathbf{u}_r\mathbf{v}_r^T] = \mathrm{Re}(\mathbf{A}E[\delta\delta^{*T}]\mathbf{B}^{*T})/2. \tag{48}$$

There are two types of error sources we need to take into considerations. The first is the noises in images, and the other is the truncation error in Eq. 26 and Eq. 23. To simplify the problem, we make the following assumptions:

1. The error caused by truncation in Eq. 23 is negligible.

2. The noise is white additive noise.

Based on these two types of error sources, the error in estimating u and v in Eq. 43 and Eq. 44 can be represented as (Appendix C)

$$\left[\begin{array}{c} du \\ dv \end{array}\right] = \mathbf{AW}, \tag{49}$$

where $\mathbf{A}$ is a 2x6 complex matrix, and $\mathbf{W}$ is the 6x1 error vector. Applying Eq. 48 to Eq. 49, we have the covariance of u_r and v_r as

$$E\left[\begin{array}{cc} du_r du_r & du_r dv_r \\ du_r dv_r & dv_r dv_r \end{array}\right] = \mathrm{Re}(\mathbf{A}E[\mathbf{WW}^{*T}]\mathbf{A}^{*T})/2, \tag{50}$$

where the covariance matrix $E[\mathbf{WW}^{*T}]$ is a 6x6 diagonal real matrix.

3.3 Moment Filters For Stereo

3.3.1 Problem Definition

The method of depth from stereo is to obtain depth information using a quantitative measure of 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,

$$\begin{aligned} i_1(x) &= i_0(x+D_0), && (51)\\ I_1(f) &= I_0(f)e^{jfD_0}. && (52) \end{aligned}$$

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 within the window. If D_0 varies, instead, in the neighborhood of window center x_0 (denoted as $\epsilon(x_0)$), i.e.

$$D_0 = D + \mu x, x \in \epsilon(x_0), \tag{53}$$

we then have

$$i_0(x+D_0) \approx i_0(x+D) + \mu x \frac{d}{dx} i_0(x+D), x \in \epsilon(x_0), \tag{54}$$

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

We need to solve the following two problems:

- *Finite Window Problem*: Since Eq. 52 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. 52 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. 53?

3.3.2 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. Let us define U_i and V_i as results from convolving the two images with the ith moment filter, i.e.

$$\begin{aligned} U_i &= \int_{-\infty}^{+\infty} i_0(x)m_i(x)dx \\ &= (-1)^i \int_{-\infty}^{+\infty} I_0(f_0)M_i(f)df && (55)\\ V_i &= \int_{-\infty}^{+\infty} i_1(x)m_i(x)dx \\ &= (-1)^i \int_{-\infty}^{+\infty} I_0(f)e^{jfD_0}M_i(f)df && (56) \end{aligned}$$

In the neighborhood of f_0, i.e. $\epsilon(f_0)$, the transformation curve can be expanded according to Taylor's expansion

$$e^{jfD_0} \approx e^{jf_0D_0}(1 + jD_0(f-f_0) - \frac{D_0^2}{2}(f-f_0)^2), f \in \epsilon(f_0), \tag{57}$$

where higher order expansions are truncated.

Replacing Eq. 57 into Eq. 56 and using the recursive property of the moment filter in the Fourier domain, we have

$$V_0 = e^{jf_0 D_0}(U_0 - jD_0U_1 - \frac{D_0^2}{2}U_2). \tag{58}$$

Therefore, we have the equation to compute D_0 as

$$D_0 = \frac{j}{f_0}\left(\ln(U_0 - jD_0U_1 - \frac{D_0^2}{2}U_2) - \ln V_0\right) \tag{59}$$

Again, as we showed in the focus problem, the difference between Eq. 59 and Eq. 52 is 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. 59 the same as Eq. 52. Therefore, when the magnitude of U_1 or U_2 is not small, using Eq. 52 to compute D_0 will result into large errors.

By analyzing this way, it is not a surprise to learn that the stability constraint used by David Fleet is actually the magnitude ratio of U_1 and U_0 , i.e.

$$\| \frac{\partial\phi}{\partial x} + j\frac{1}{a}\frac{\partial a}{\partial x} \| = \frac{\| U_1 \|}{\| U_0 \|} \tag{60}$$

where ϕ is the phase, and a is the magnitude. Eq. 60 also justified the so-called phase singularity, which is location where $U_1(f_0)$ is so large compared with $U_0(f_0)$ that Eq. 52 isn't even approximately valid. We can see that such situation is exactly what Eq. 59 characterized.

While David Fleet figured out the constraint to find the locations of singularity introduced by a finite window, he didn't find a specific way to correct the error caused by a finite window. As the same as we did in the problem of depth from defocus, we can use Eq. 59 as an iterative procedure to find qth time estimation of D_0 as

$$\begin{aligned} D_0^{(q)} &= \frac{j}{f_0}\left(\ln(U_0 - jD_0^{(q-1)}U_1 - \frac{(D_0^{(q-1)})^2}{2}U_2) - \ln V_0\right) \\ &= T(D_0^{(q-1)}). \end{aligned} \tag{61}$$

And the sufficient condition for the iteration to converge is

$$\| \frac{d}{dD_0}T(D_0^{(j)}) \| < 1. \tag{62}$$

3.3.3 Relations to the Phase-Based Method

Some researchers [8, 23, 27] recently advocated a phase-based approach to the stereo problem. This approach simply divides the phase difference by the peak frequency or the instantaneous frequency to infer the disparity. Fleet[8] showed that when the stability criterion is small, more precise results can be produced by using the instantaneous frequency rather than the peak frequency.

Suppose we truncate the Taylor's expansion after the linear term, then Eq. 58 becomes

$$V_0 = e^{jf_0 D_0}(U_0 - jD_0 U_1). \tag{63}$$

Rewriting the above equation, we then have

$$jf_0 D_0 = \ln V_0 - \ln(U_0 - jD_0 U_1). \tag{64}$$

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

$$\ln(U_0 - jD_0 U_1) \approx \ln U_0 - jD_0 \frac{U_1}{U_0}. \tag{65}$$

Replacing Eq. 65 into Eq. 64, after some manipulations, we obtain,

$$(f_0 - \frac{U_0}{U_1})D_0 = j \ln U_0 - j \ln V_0. \tag{66}$$

Taking the real part of the both sides in the above equation, and referring to the definition of the instantaneous frequency in Eq. 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.

3.3.4 Shift Variance Problem

When $\mu \neq 0$ in Eq. 53 and Eq. 54, we must consider the effect of the shift variance. Let

$$\begin{aligned} V_i &= \int_{-\infty}^{+\infty} i_1(x) m_i(x) dx, & (67) \\ \mathcal{V}_i &= \int_{-\infty}^{+\infty} i_0(x + D) m_i(x) dx & (68) \\ &= (-1)^i \int_{-\infty}^{+\infty} I_0(f) e^{jfD} M_i(f) df. & (69) \end{aligned}$$

From convolutions of the two images with moment filters, we obtain U_i and V_i. The define $\mathcal{V}_i$ can not be computed from convolutions. But to estimate s, we need to compute $\mathcal{V}_i$ first. In other words, we need to virtually rotate back the slanted surface to make it front-parallel, compute the convolution of the rotated surface. Replacing Eq. 54 into Eq. 67, we have

$$\begin{aligned} V_i \approx & \int_{-\infty}^{+\infty} i_0(x + D) m_i(x) dx + \\ & \mu \int_{-\infty}^{+\infty} \frac{di_0(x + D)}{dx} x m_i(x) dx. & (70) \end{aligned}$$

Using the recursive property of the moment filter, after some manipulations, we can rewrite Eq. 70 as

$$V_i = \mathcal{V}_i - \mu(i(f_0\mathcal{V}_{i-1} - \mathcal{V}_i) - \sigma^2(f_0\mathcal{V}_{i+1} - \mathcal{V}_{i+2})). \tag{71}$$

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

$$\begin{aligned} \mathcal{V}_i &\approx V_i + \mu(i(f_0V_{i-1} - V_i) - \sigma^2(f_0V_{i+1} - V_{i+2})) \\ &= V_i + \mu P_i, \end{aligned} \tag{72}$$

where P_i is used for simplicity of illustration.

Combining Eq. 59 and Eq. 72, we have the equations to solve D and μ,

$$D = \frac{j}{f_0}\left(\ln(U_0 - jDU_1 - \frac{D^2}{2}U_2) - \ln(V_0 + \mu P_0)\right) \tag{73}$$

$$\frac{V_1 + \mu P_1}{V_0 + \mu P_0} \approx \frac{U_1 - jDU_2 - \frac{1}{2}D^2U_3}{U_0 - jDU_1 - \frac{1}{2}D^2U_2} \tag{74}$$

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 considerations. in error estimation as we did in focus. The first one is noises in images, and the other is the truncation error in Taylor's expansion. To simplify the problem, we make the following assumptions :

1. The error caused by truncation in Eq. 54 is negligible.

2. The noise is white additive noise.

Based on these two types of error sources, the error in estimating D and μ can be represented as (Appendix C)

$$\begin{bmatrix} dD \\ d\mu \end{bmatrix} = \mathbf{AW}, \tag{75}$$

where $\mathbf{A}$ is a 2x6 complex matrix, and $\mathbf{W}$ is a 6x1 error vector. Applying Eq. 48 to Eq. 75, we have the covariance of D_r and μ_r as

$$E\begin{bmatrix} dD_r dD_r & dD_r d\mu_r \\ dD_r d\mu_r & d\mu_r d\mu_r \end{bmatrix} = \mathrm{Real}(\mathbf{A}E[\mathbf{WW}^{*T}]\mathbf{A}^{*T})/2, \tag{76}$$

where the covariance matrix $E[\mathbf{WW}^{*T}]$ is a 6x6 diagonal real matrix.

3.4 Summary

We have shown how the moment filters can be used 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 single parameter in the transformation between two images. As long as the transformation is well modeled, the parameter and its gradient can be accurately estimated from one frequency band 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's expansion.

2. The recursive property of the moment filters in the Fourier domain allows expanding 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 obtained an equation of the parameter, which can be usually solved by iteration.

3. The recursive property 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 correct the error in estimating the parameter caused by the gradient.

Though we are very successful in solving transformations with one unknown parameter, such as problems of focus and stereo, there are some intrinsic limitations with the paradigm based on *one* frequency band.

- It is usually difficult to solve transformations which have more than one parameter because it is an under-constrained problem for one frequency band. For example, it is not easy to extend the algorithm of stereo to solve the 2D correspondence problem because 2D shift involves two independent parameters in the transformation. Yet because we can have tens or even hundreds of frequency bands, the overall problem should be overconstrained on the other hand. Therefore, to solve problems with more than one parameter, we can no longer consider each frequency band separately, we have to consider all together.

- There is another related problem, which we refer to as the frequency sampling problem. Though we can optimally combine results from different frequency bands through Kalman filtering, the selections of peak frequency f_0 are still *ad hoc*. They by no means represent the image content either *completely* or *nonredundantly*.

To overcome these problems, we propose another set of filters, "hypergeometric" filters, which have similar recursive properties as the moment filters do, and don't suffer from the above two problems.

4 Hypergeometric Filter Approach

Hypergeometric filters have the similar recursive properties as the moment filters have, and two additional properties:

- All hypergeometric filters have only one single peak frequency. Because the high order moment filters have two peak frequencies, we have to base our analysis on zero-order filters which have one single peak frequency. To hypergeometric filters, We can certainly use all high order ones.

- The set of hypergeometric filters samples the frequency domain completely and nonredundantly.

We will show that, by using the hypergeometric filters, a general multiple parameter extraction problem can be formulated as a multidimensional minimization problem. Finally, we will apply this new technique to the optical flow problem.

4.1 Theory of Hypergeometric Filters

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

$$\begin{aligned}
H_m(f) &= \begin{cases} c_m f^m e^{-f^2\sigma^2/2} & \text{when } f \geq 0 \\ 0 & \text{when } f < 0 \end{cases} && (77)\\
H_0(f) &= (2\sqrt{\pi}\sigma)^{1/2} e^{-f^2\sigma^2/2} = c_0 e^{-f^2\sigma^2/2} && (78)\\
H_{-m}(f) &= \begin{cases} 0 & \text{when } f > 0 \\ c_m(-f)^m e^{-f^2\sigma^2/2} & \text{when } f \leq 0 \end{cases}, && (79)
\end{aligned}$$

where c_m is a real normalization constant as

$$c_m = 2\sigma^m \sqrt{\frac{\pi\sigma}{,\,(m+1/2)}}. \tag{80}$$

Intuitively, a H filter represents either one peak of the two of a moment filter in Figure 2 with $f_0 = 0$. Therefore, we can easily verify that $H_m(f)$ is an asymmetric bell-shaped band pass filter with the peak frequency at:

$$\begin{aligned}
\text{Peak}[H_m(f)] &= \frac{\sqrt{m}}{\sigma}, && (81)\\
\text{Peak}[H_0(f)] &= 0, && (82)\\
\text{Peak}[H_{-m}(f)] &= -\frac{\sqrt{m}}{\sigma}. && (83)
\end{aligned}$$

Doing inverse Fourier transform in $H_m(f)$ in Eq. 77 and $H_{-m}(f)$ in Eq. 79, we then have the H filters represented in the spatial domain

$$h_m(x) \;=\; \sqrt{\frac{2^m}{\pi,\,(m+1/2)\sigma}}\, e^{-\frac{x^2}{2\sigma^2}}$$

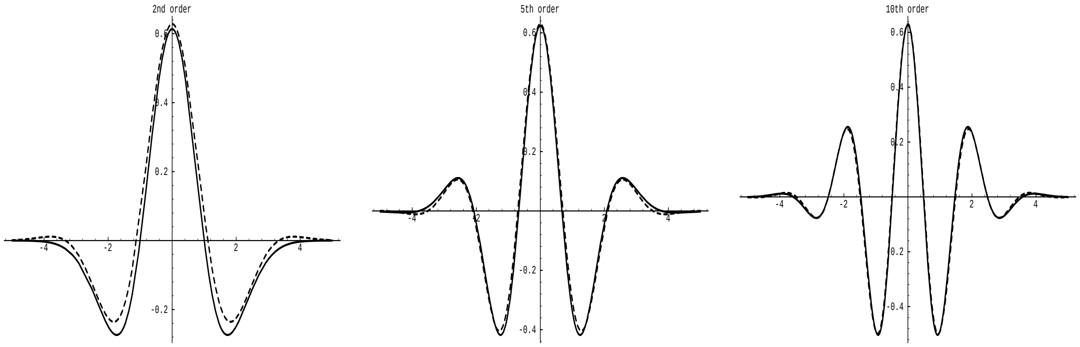


Figure 4: H Filters and Their Closest Gabor Filters

$$
\begin{aligned}
& \left(\frac{1}{\sqrt{2}}, \; (\frac{1+m}{2})\Phi(-\frac{m}{2}, \frac{1}{2}, \frac{x^2}{2\sigma^2}) + j, \; (1+\frac{m}{2})\Phi(\frac{1-m}{2}, \frac{3}{2}, \frac{x^2}{2\sigma^2})\frac{x}{\sigma}\right) \\
= & \left(a_m\Phi(-\frac{m}{2}, \frac{1}{2}, \frac{x^2}{2\sigma^2}) + jb_m x\Phi(\frac{1-m}{2}, \frac{3}{2}, \frac{x^2}{2\sigma^2})\right) e^{-\frac{x^2}{2\sigma^2}}, \qquad (84) \\
h_0(x) = & \; (\sqrt{\pi}\sigma)^{-1/2} e^{-\frac{x^2}{2\sigma^2}}, \qquad (85) \\
h_{-m}(x) = & \; h_m^*(x), \qquad (86)
\end{aligned}
$$

where $\Phi(a, c, x)$ is a special type of hypergeometric function, usually called "confluent hypergeometric function" or "Kummer's function" in literature, and a_m and b_m are constants decided by m. Eq. 84 has already been normalized so that the energy of the filter $h_m(x)$ is one. The solid curves in Figure 4 are H filters of second, fifth and tenth order.

As shown in Appendix D, the H filters have the following property:

$$
\begin{aligned}
h_{2i}(x) + h_{-2i}(x) &= k_1 L(i, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}} \qquad (87) \\
h_{2i+1}(x) - h_{-(2i+1)}(x) &= jk_2 x L(i, \frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}}, \qquad (88)
\end{aligned}
$$

where k_1 and k_2 are real constants decided by i, and $L(n, \alpha, x)$ is the generalized Laguerre's polynomial. Comparing Eq. 87 and Eq. 88 with Eq. 2 and Eq. 13, referring to the fact that moment filters are orthogonal and complete, we conclude that we can losslessly reconstruct the original signal from coefficients resulted from convolving the image with H filters.

Another important property of H filters is their recursive property (Appendix D) in the spatial and Fourier domains, i.e. in the Fourier domain,

$$
\begin{aligned}
fH_m(f) &= \frac{c_m}{c_{m+1}} H_{m+1}(f), \qquad (89) \\
fH_0(f) &= \frac{c_0}{c_1}(H_1(f) - H_{-1}(f)), \qquad (90) \\
fH_{-m}(f) &= -\frac{c_m}{c_{m+1}} H_{-(m+1)}(f), \qquad (91)
\end{aligned}
$$

and in the spatial domain,

$$\begin{aligned} xh_m(x) &= -j\sigma\left(\sqrt{m+\frac{1}{2}}h_{m+1}(x) - \sqrt{\frac{m^2}{m-1/2}}h_{m-1}(x)\right), && (92)\\ xh_0(x) &= -\frac{1}{2}j\sigma(h_1(x) - h_{-1}(x)), && (93)\\ xh_{-m}(x) &= j\sigma\left(\sqrt{m+\frac{1}{2}}h_{-(m+1)}(x) - \sqrt{\frac{m^2}{m-1/2}}h_{-(m-1)}(x)\right). && (94) \end{aligned}$$

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 which are extremely harmful when the peak frequency is low.

Now let's look at the effective bandwidth in the Fourier domain and effective spatial extension. According to Gabor [9], we have

$$\begin{aligned} \Delta x_m^2 &= \frac{\int_{-\infty}^{+\infty} x^2 \parallel h_m(x) \parallel^2 dx}{\int_{-\infty}^{+\infty} \parallel h_m(x) \parallel^2 dx}, && (95)\\ \Delta f_m^2 &= \frac{\int_{-\infty}^{+\infty} (f - \text{Peak}[H_m(f)])^2 \parallel H_m(f) \parallel^2 df}{\int_{-\infty}^{+\infty} \parallel H_m(f) \parallel^2 df}. && (96) \end{aligned}$$

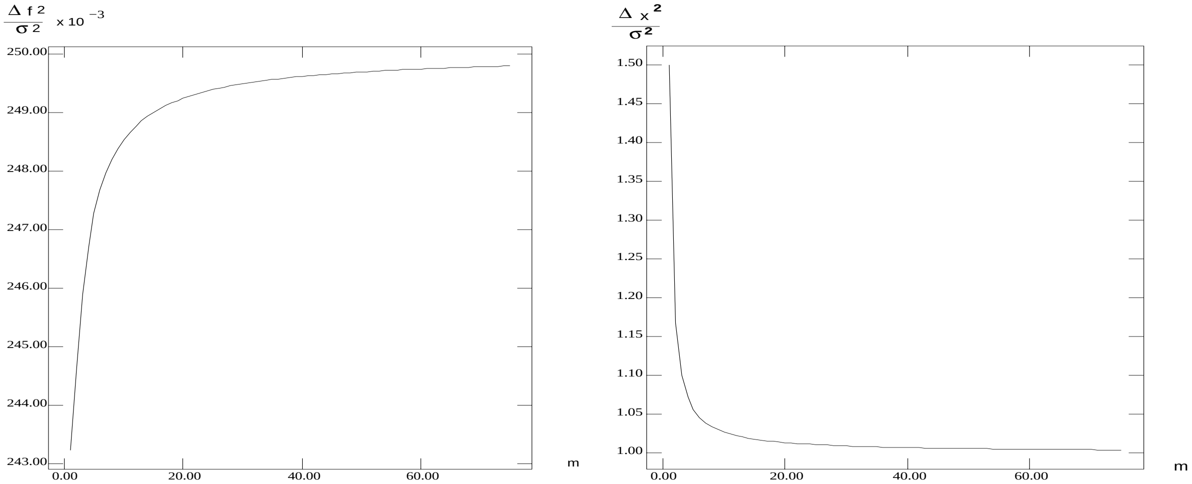
Figure 5 shows the effective bandwidth and effective spatial extension. As we all know the uncertainty principle in local frequency analysis, the effective bandwidth and effective spatial extension must satisfy the following uncertainty relation:

$$\varepsilon_m = \sqrt{\Delta x_m^2 \Delta f_m^2} \geq \frac{1}{2}. \quad (97)$$

And Figure 6 shows that, as Gabor filters have the minimal uncertainty, the product of the spatial and the spectral uncertainties of the H filters is almost the minimal though it is slightly larger ($\varepsilon_m < 0.51$, when $m > 5$).

Since Gabor filters are the only filters which have 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, in which solid curves are H filters, dashed curves are the closest Gabor filters, in the sense that they have the same peak frequency and the same bandwidth.

The H filters sampled the frequency space automatically as illustrated in Eq. 81 and Eq. 83. Unlike the frequency sampling in short time Fourier transform, which is uniform, the H filters sample the frequency space in a non-uniform fashion. Figure 7 shows the samplings of the frequency space in STFT and H filters, in case of $\sigma = 2.0$ and window size is 21. Such a sampling in frequency domain will be shown to have great advantage in combining information contained in different frequency bands and solving the multiple parameter problem.



Effective Bandwidth

Effective Spatial Extension

Figure 5: Uncertainty in The Spatial and Frequency Domains

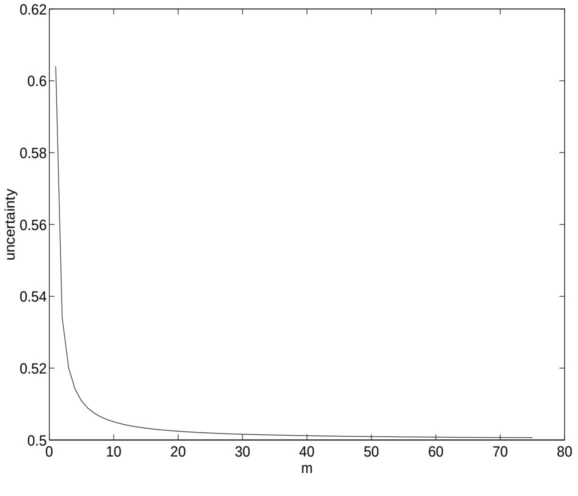
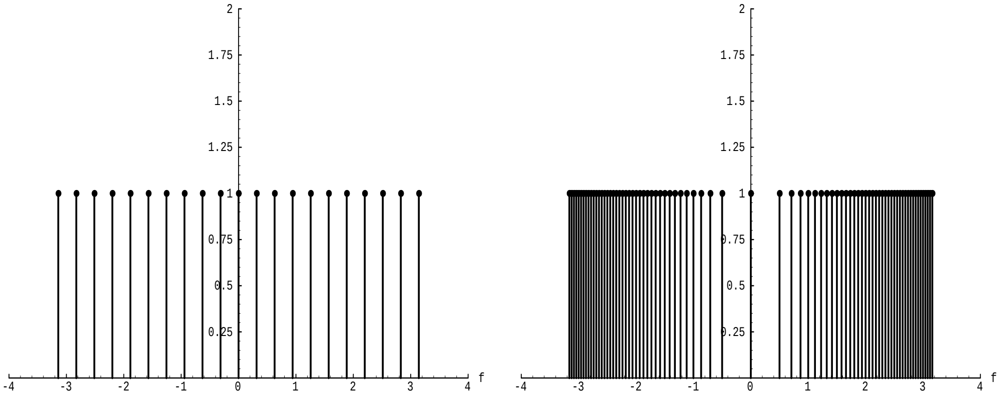


Figure 6: Product of the Spatial and Frequency Uncertainty



STFT

H Filters

Figure 7: Frequency Sampling in STFT and H Filters

4.1.1 A General Approach to Multiple Parameter Problems

As an abstract example, let us consider a problem of trying to recover two parameters (p_1, p_2) in the transformation of $T(f; p_1, p_2)$ between two images $I_1(f)$ and $I_2(f)$. As we stated before, analyzing a single frequency band as we did in the moment filter approach is not enough to compute two parameters. Therefore, we have to use multiple frequency bands together to solve this problem. For simplicity of notation, we only use H filters with positive frequency peaks $h_m(x), (m = 1, 2, ...)$ in this section.

The transformation can be represented as

$$I_2(f) = T(f; p_1, p_2)I_1(f) \tag{98}$$

Suppose the transformation $T(f; p_1, p_2)$ can be approximated by an Nth order polynomial T_p in the neighborhood of Peak$[H_m(f)] = \frac{\sqrt{m}}{\sigma}$:

$$T_p(f; p_1, p_2) = \sum_{i=0}^{N} A_i(m, p_1, p_2)f^i, f \in \epsilon(\frac{\sqrt{m}}{\sigma}) \tag{99}$$

and let

$$\begin{aligned} U_m &= \int_{-\infty}^{+\infty} i_1(x)h_m(-x)dx \\ &= \int_{0}^{+\infty} I_1(f)c_m f^m e^{-f^2\sigma^2/2}df & (100) \\ V_m &= \int_{-\infty}^{+\infty} i_2(x)h_m(-x)dx \\ &= \int_{0}^{+\infty} I_2(f)c_m f^m e^{-f^2\sigma^2/2}df \\ &= \int_{0}^{+\infty} I_1(f)T(f; p_1, p_2)c_m f^m e^{-f^2\sigma^2/2}df, & (101) \end{aligned}$$

and use the recursive property of the H filters in the Fourier domain,

$$f^i H_m(f) = \frac{c_m}{c_{m+i}} H_{m+i}(f), \tag{102}$$

we then have,

$$V_m = \sum_{i=0}^{N} \frac{c_m}{c_{m+i}} A_i(m, p_1, p_2) U_{m+i}. \tag{103}$$

Apparently, we can not robustly solve (p_1, p_2) from a single equation as Eq. 103 as we have two unknowns. But because the same relation should be satisfied by all frequency bands, we can therefore minimize the following to find (p_1, p_2):

$$S(p_1, p_2) = \sum_{m=M_1}^{M_2} \parallel V_m - \sum_{i=0}^{N} \frac{c_m}{c_{m+i}} A_i(m, p_1, p_2) U_{m+i} \parallel^2, \tag{104}$$

where M_1 and M_2 are the indexes of filters which have the lowest and highest peak frequencies respectively. A 2D conjugate gradient minimization algorithm can be applied to the above function to detect the minima quickly.

We could also represent the above minimization on matrix format. Let,

$$\frac{c_m}{c_{m+i}} A_i(m, p_1, p_2) = k_{mi}(p_1, p_2), \tag{105}$$

we can rewrite Eq. 103 as,

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} k_{10} & k_{11} & \cdots & k_{1N} & 0 & 0 & 0 & \cdots \\ 0 & k_{20} & k_{21} & \cdots & k_{2N} & 0 & 0 & \cdots \\ 0 & 0 & k_{30} & k_{31} & \cdots & k_{3N} & 0 & \cdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ \vdots \end{bmatrix}, \tag{106}$$

or in a more concise form,

$$\mathbf{V} = \mathbf{K}(p_1, p_2)\mathbf{U}, \tag{107}$$

where $\mathbf{V}$ and $\mathbf{U}$ are known vectors from convolutions, $\mathbf{K}(p_1, p_2)$ is a matrix whose elements are functions of (p_1, p_2). Therefore, the minimization of S in Eq. 104 is equivalent to minimizing the norm of difference between the left and the right side of Eq. 107.

Whenever we try to extract multiple parameters, there is a so-called aperture problem, which is caused by lack of constraint along certain direction in the parameter space. In context of multidimensional minimization, the aperture problem happens when the function S has a canyon instead of an obvious minima in the parameter space. Therefore, the aperture problem can be represented by an error covariance matrix with a large eigenvalue along one direction as we will show later in experiments.

4.2 Hypergeometric Filter Approach For Optical Flow

We will apply the technique of solving a general multiple parameter problem to the two-parameter optical flow problem. The 2D transformation in optical flow problem is assumed to be translation, i.e. assuming two adjacent frames are $i_0(x,y)$ and $i_1(x,y)$, and the image velocity is (v_x, v_y), we then have,

$$\begin{aligned} i_1(x,y) &= i_0(x+v_x, y+v_y), & (108)\\ I_1(f_x,f_y) &= I_0(f_x,f_y)e^{j(f_xv_x+f_yv_y)}. & (109) \end{aligned}$$

For simplicity of illustration, we only approximate the above transformation by a second-order polynomial according to Eq. 99 as

$$\begin{aligned} T_p(f_x,f_y;v_x,v_y) &= e^{j(f_xv_x+f_yv_y)} \\ &\approx (1+jv_x(f_x-f_{0x})+jv_y(f_y-f_{0y})-v_x^2(f_x-f_{0x})^2 \\ &\quad -v_xv_y(f_x-f_{0x})(f_y-f_{0y})-v_y^2(f_y-f_{0y})^2)e^{j(f_{0x}v_x+f_{0y}v_y)} \\ &= \sum_{i=0}^{2}\sum_{j=0}^{2-i} C_{ij}(f_{0x},f_{0y},v_x,v_y)f_x^if_y^j. \qquad (110) \end{aligned}$$

Considering only positive peak frequency H filters, i.e.

$$H_{mn}(f_x,f_y)=H_m(f_x)H_n(f_y),$$

we then obtain the following relation from Eq. 103:

$$V_{mn}=\sum_{i=0}^{2}\sum_{j=0}^{2-i}C_{ij}(\frac{\sqrt{m}}{\sigma},\frac{\sqrt{n}}{\sigma},v_x,v_y)\frac{c_m}{c_{m+i}}\frac{c_n}{c_{n+j}}U_{(m+i)(n+j)}. \qquad (111)$$

Therefore, the function to be minimized will be,

$$S(v_x,v_y)=\sum_{m=M_1}^{M_2}\sum_{n=N_1}^{N_2}\parallel V_{mn}-\sum_{i=0}^{2}\sum_{j=0}^{2-i}C_{ij}(\frac{\sqrt{m}}{\sigma},\frac{\sqrt{n}}{\sigma},v_x,v_y)\frac{c_mc_n}{c_{m+i}c_{n+j}}U_{(m+i)(n+j)}\parallel^2 . \qquad (112)$$

H filters with negative or zero peak frequency can be similarly applied, and they can all summed into $S(v_x,v_y)$.

4.2.1 Error Estimation

Two error sources are going to be considered in error estimation. One is the white noise contained in the original images. The other is the truncation error in Eq. 110. Suppose the intensity of the white noise contained in the two images are both N_w^2, it is equivalent to that the first image contains $2N_w^2$ noise, and the second image contains no noise. Therefore, both error sources result in inaccuracy of V_{mn} in Eq. 112, which consequently causes the estimation error of (v_x,v_y).

Suppose the errors of V_{mn}, v_x and v_y are dV_{mn}, dv_x and dv_y respectively, we can differentiate both sides of Eq. 112 as

$$\frac{\partial S}{\partial v_x} = 2\mathrm{Re}\sum_{m=M_1}^{M_2}\sum_{n=N1}^{N_2}((V_{mn}^*-\sum_{i=0}^{2}\sum_{j=0}^{2-i}C_{ij}^*\frac{c_mc_n}{c_{m+i}c_{n+j}}U_{(m+i)(n+j)}^*)$$

$$
\begin{aligned}
& (-\sum_{i=0}^{2}\sum_{j=0}^{2-i}\frac{\partial C_{ij}}{\partial v_x}\frac{c_m c_n}{c_{m+i}c_{n+j}}U_{(m+i)(n+j)})), && (113)\\
\frac{\partial S}{\partial v_y} &= 2\mathrm{Re}\sum_{m=M_1}^{M_2}\sum_{n=N1}^{N_2}((V^*_{mn}-\sum_{i=0}^{2}\sum_{j=0}^{2-i}C^*_{ij}\frac{c_m c_n}{c_{m+i}c_{n+j}}U^*_{(m+i)(n+j)})\\
& (-\sum_{i=0}^{2}\sum_{j=0}^{2-i}\frac{\partial C_{ij}}{\partial v_y}\frac{c_m c_n}{c_{m+i}c_{n+j}}U_{(m+i)(n+j)})). && (114)
\end{aligned}
$$

At the minima, the above gradient should be zero. Then the task of the error estimation is to estimate how the minima will move when there is a small random perturbation in V_{mn}. Let

$$
\mathcal{C}_{mn}(v_x, v_y) = \sum_{i=0}^{2}\sum_{j=0}^{2-j}C_{ij}(\frac{\sqrt{m}}{\sigma}, \frac{\sqrt{n}}{\sigma}, v_x, v_y)\frac{c_m c_n}{c_{m+i}c_{n+j}}U_{(m+i)(n+j)}, \tag{115}
$$

we can take differentiation of right sides of Eq. 113 and Eq. 114,

$$
\begin{aligned}
& \mathrm{Re}\left(\sum_{m,n}(V^*_{mn}-\mathcal{C}^*_{mn})\frac{\partial^2\mathcal{C}_{mn}}{\partial v_x^2} - \parallel \frac{\partial \mathcal{C}_{mn}}{\partial v_x} \parallel^2\right) dv_x + \\
& \mathrm{Re}\left(\sum_{m,n}(V^*_{mn}-\mathcal{C}^*_{mn})\frac{\partial^2\mathcal{C}_{mn}}{\partial v_x \partial v_y} - \frac{\partial \mathcal{C}^*_{mn}}{\partial v_y}\frac{\partial \mathcal{C}_{mn}}{\partial v_x}\right) dv_y \\
= & \mathrm{Re}\left(-\sum_{m,n}\frac{\partial \mathcal{C}_{mn}}{\partial v_x}dV^*_{mn}\right) && (116)\\
& \mathrm{Re}\left(\sum_{m,n}(V^*_{mn}-\mathcal{C}^*_{mn})\frac{\partial \mathcal{C}^2_{mn}}{\partial v_x \partial v_y} - \frac{\partial \mathcal{C}^*_{mn}}{\partial v_x}\frac{\partial \mathcal{C}_{mn}}{\partial v_y}\right) dv_x + \\
& \mathrm{Re}\left(\sum_{m,n}(V^*_{mn}-\mathcal{C}^*_{mn})\frac{\partial^2\mathcal{C}_{mn}}{\partial v_y^2} - \parallel \frac{\partial \mathcal{C}_{mn}}{\partial v_y} \parallel^2\right) dv_y \\
= & \mathrm{Re}\left(-\sum_{m,n}\frac{\partial \mathcal{C}_{mn}}{\partial v_y}dV^*_{mn}\right), && (117)
\end{aligned}
$$

or in a more concise form,

$$
\mathbf{A}\begin{pmatrix} dv_x \\ dv_y \end{pmatrix} = \mathbf{W}, \tag{118}
$$

where $\mathbf{A}$ is a 2x2 real matrix, and $\mathbf{W}$ is a 2x1 real vector. From above equation, we can have the covariance error estimation,

$$
E\begin{bmatrix} dv_x dv_x & dv_x dv_y \\ dv_x dv_y & dv_y dv_y \end{bmatrix} = \mathbf{A}^{-1}E[\mathbf{W}\mathbf{W}^T]\mathbf{A}^{-T}, \tag{119}
$$

where we have

$$
E[\mathbf{W}\mathbf{W}^T] = \frac{1}{2}\mathrm{Re}\sum_{m,n}\begin{pmatrix} \parallel \frac{\partial \mathcal{C}_{mn}}{\partial v_x} \parallel^2 & \frac{\partial \mathcal{C}^*_{mn}}{\partial v_x}\frac{\partial \mathcal{C}_{mn}}{\partial v_y} \\ \frac{\partial \mathcal{C}_{mn}}{\partial v_x}\frac{\partial \mathcal{C}^*_{mn}}{\partial v_y} & \parallel \frac{\partial \mathcal{C}_{mn}}{\partial v_y} \parallel^2 \end{pmatrix} E[\parallel dV_{mn} \parallel^2], \tag{120}
$$

by applying Eq. 48.

4.3 Summary

Our new hypergeometric filter approach can in principle handle any problem of extracting one or more parameters from a nonstationary transform between two images. We showed how to approach the optical flow computation problem, which is a two parameter extraction problem. For different parameter extraction problems, the algorithms of the hypergeometric approach differ only in the Taylor's expansion for different transforms.

5 Comparisons of Moment Filters and Hypergeometric Filters

Because both moment filters and hypergeometric filters have similar recursive properties in the spatial and Fourier domains, they have the same strategy for solving parameter extraction problems. This strategy uses a polynomial to fit the underlying transformation curve, which is a surface in two parameter cases and hypersurface in more than two parameter cases, and therefore obtain results of much higher precision the traditional approach which uses a constant to approximate the curve. In this respect the two approaches are equally effective in computing accurately and handling the foreshortening problem. On the other hand, the hypergeometric filter approach can solve multiple parameter problems, while the moment filter is good at dealing with one parameter problems even though it is possible that multiple bands can be combined to solve multiple parameter problems.

The moment filter approach is based on a single frequency band. Thus we have the freedom to sample the frequency space as we like. The disadvantages 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 scheme such as the wavelet scheme in [30].

The advantage and disadvantage 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 are utilized. But on the other hand, we don't have the freedom to choose different window size for different frequency bands.

6 Implementation Issues and Experiments

6.1 Moment Filter Approach

In this section, we will explain some implementational issues of the moment filter approach, and quantitatively compare the focus and stereo algorithms based on moment filters with other algorithms in literature.

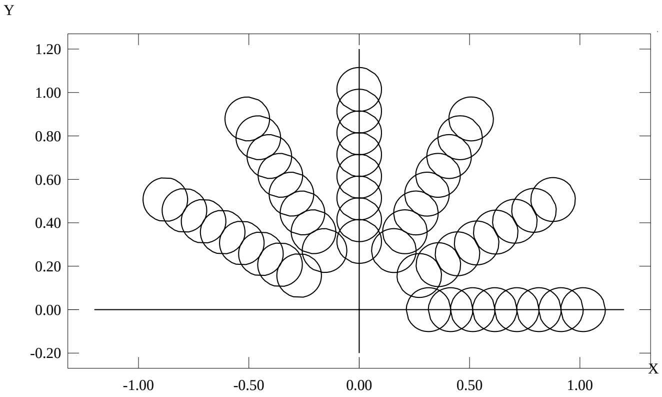


Figure 8: Frequency Sampling in Fixed Window Scheme

6.1.1 Sampling of the Frequency Domain

The moment filter approach described above is based on one single Gabor filter. Since, the information contained in a image is usually distributed in all frequency bands, we have to use multiple Gabor filters and their accompanying moment filter sets. 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 a polar coordinate, 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 totally 48 Gabor filters and 48 moment filter sets. In the experiments shown below, we use up to the 4th order moment filters.

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,

$$\sigma = k\lambda, \tag{121}$$

where k is a constant and λ is the peak wavelength of the Gabor filter, the radial frequency sampling is nonuniform. We don't intend to compare these two window schemes here, but rather provide two choices for the convenience of experimentation. Figure 8 and Figure 9 illustrate the sampling of the frequency plane under the window schemes. The circles represent the effective bandwidth of Gabor filters.

The major difficulty of the filter-based approach in implementation is its enormous amount of computation and memory requirements if we need to sample many frequency bands. We believe that the as parallel computers become more powerful, this problem will be much less severe. Because of this limitation, we only use 48 filters which is 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 be able to further

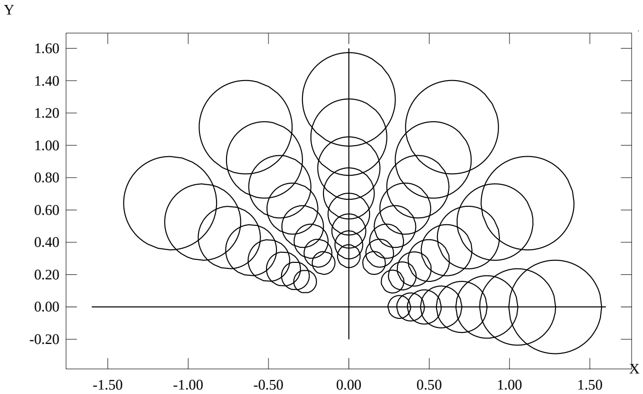


Figure 9: Frequency Sampling in Variable Window Scheme

improve the results reported here.

6.1.2 Focus

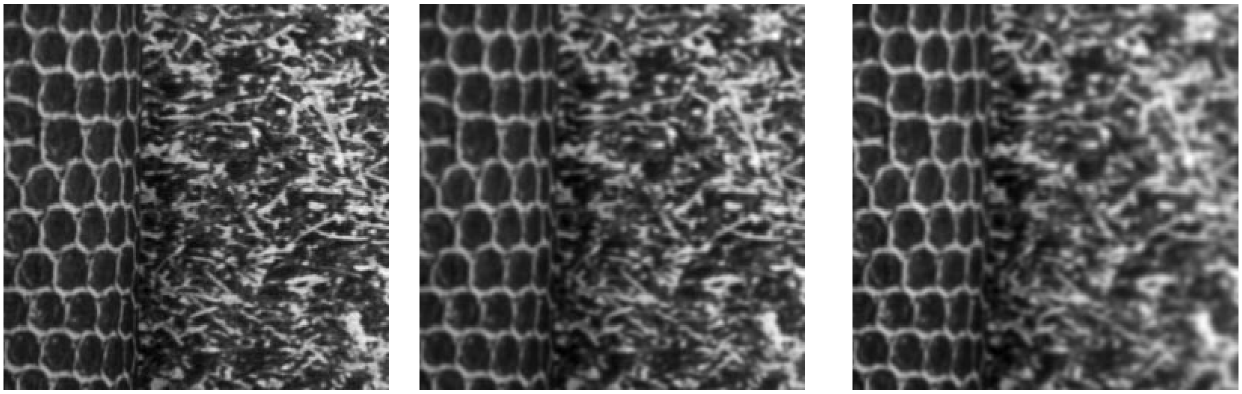
We implement the moment filter approach for the focus problem using the third order Taylor's expansion in Eq. 26. Because computation of P_1 in Eq. 40 requires the fourth order moment filter, we need up to the fourth order moment filters for every Gabor filter. To accommodate numerical errors, the convergence thresholds in Eq. 33 and Eq. 45 are set to 0.9. Results from different frequency bands are merged optimally through Kalman filtering as in [24].

Subbarao's algorithm in [26] is directly from Eq. 21, which can be regarded as a special case of our moment filter approach, in which the Taylor's expansion in Eq. 26 only keeps the constant term. But Subbarao's algorithm can not be extended to deal with the shift variance problem.

We implemented both algorithms under the variable window scheme, in which k in Eq. 121 is 0.8. The performance tests are done on both synthetic and real images. The original image is shown in Figure 10. We uniformly blur the image by a Gaussian function with parameter $\sigma_0 = 1.0$ to get the first synthetic image pair. We also non-uniformly blur the original image by a Gaussian function with parameter σ_0 increases linearly as the column increases, to get the second synthetic image pair.

Since in 2D images, the slope of a surface has to be represented by a vector instead of one number in Eq. 22, we have to solve three unknown equations in case of 2D image pairs instead of two unknown equations in 1D case.

We compare three different algorithms, namely, Subbarao's algorithm (SA), the moment filter algorithm without considering slope (MFF1), and the moment filter algorithm with considering slope (MFF2). Table 1 shows the root mean square errors of the synthetic image pairs using different algorithms. In the case of the first pair, MFF1 and MFF2 shows almost no error at all, while SA has a significant amount of error. In the case of 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 error source of Subbarao's algorithm is from the truncation of information from moment



Original Image Uniformly Blurred Image Nonuniformally Blurred Image

Figure 10: Synthetic Texture Images For Focus

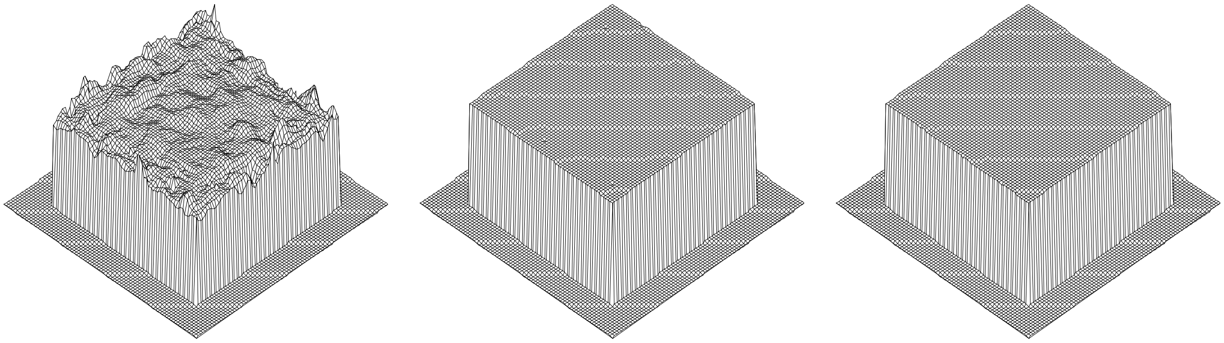
	SA	MFF1	MFF2
The First Pair	0.070518	0.000749	0.000316
The Second Pair	0.190362	0.046419	0.007976

Table 1: RMS Errors of Different Algorithms

filters. When the surface is slanted, i.e. the foreshortening happens, it also causes errors depends on how slanted the surface is. The MFF2 algorithm can correct the errors caused by the foreshortening.

To see the precision of different algorithms, Figure 11 shows the computed local blurring difference σ_0^2 of the first image pair, and Figure 12 shows the computed local blurring difference σ_0^2 of the second pair. Furthermore, Figure 13 illustrates the slope vectors computed by MFF2 for the second image pair. Note that the slope vector has been scaled by σ_0^2 as the left side of Eq. 42, which causes the magnitudes of slope vectors increases horizontally. We can see that MFF2 not only can correct the errors caused by the foreshortening, but also estimate the slope of the surface.

The real image pair is shown in Figure 14. These two pictures are taken by the



SA MFF1 MFF2

Figure 11: Computed Blurring Difference of The First Pair

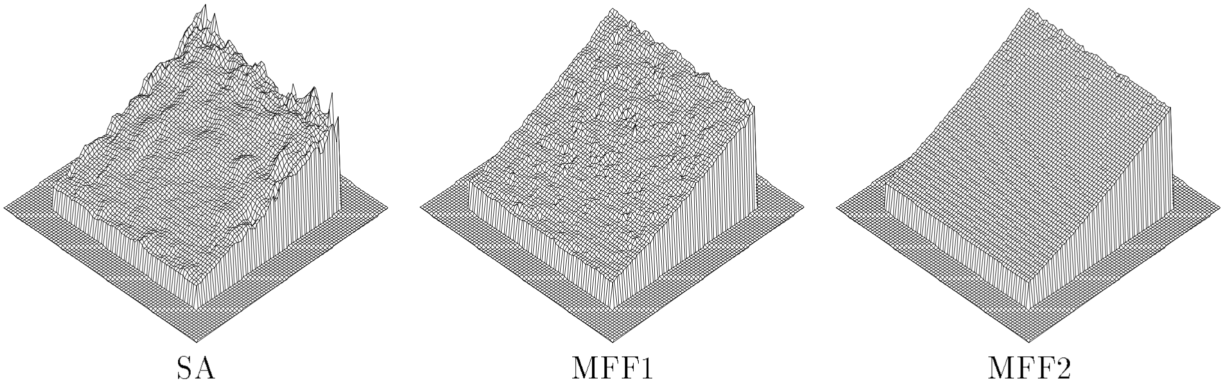


Figure 12: Computed Blurring Difference of The Second Pair

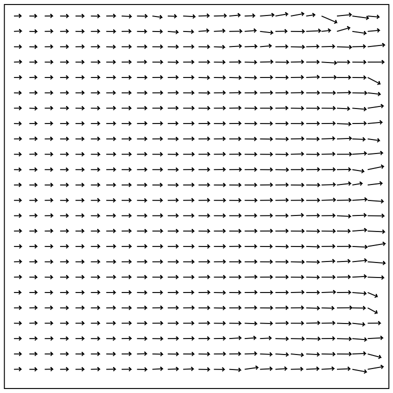


Figure 13: Computed Surface Slope of The Second Pair

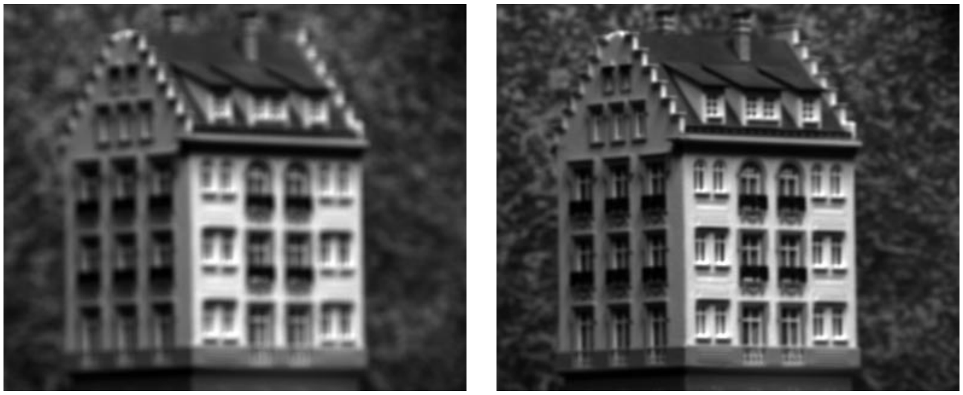


Figure 14: Real Image Pair of A Toy House

Photometrics camera [28] under different aperture and exposure time setting. 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. To compensate for the optical vignetting and different exposure amount between two images, we use an approximately uniform illumination to correct them by putting a diffuser in front of the camera, and a light source far away.

Figure 15 shows the computed blurring difference of the house image pair using Subbarao's algorithm. We can see that the result is so noisy that the shape of the house can not be observed with confidence. Figure 16 shows the result using MFF1. By now the shape of the house is clearly visible though the discontinuity still generates a lot of spikes. Finally Figure 17 shows the result using MFF2. We can see that the result is even more smooth. Additionally, Figure 18 shows the computed slope and estimated error ($E[du_r du_r]$ in Eq. 50) in MFF2. We can see that most of the slope vectors have the correct direction despite that the slope is very small ($\mu \approx 0.002$ in Eq. 22), and the discontinuity generates very large slope vectors and estimated errors which can be used to locate the discontinuity in later processing stages.

6.1.3 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 & Okutomi's adaptive window scheme [13]. A simple SSD-based matching is performed to generate pixel resolution disparity, and the same pixel resolution disparity is used for our algorithms and Kanade & Okutomi's algorithm.

We implemented out algorithms under the fixed window scheme, in which the window size $\sigma = 7.0$. The minimal radial frequency is $\pi/10$, and the maximal radial frequency is about $\pi/3$. There are totally 48 filters as shown before. For Kanade & Okutomi's algorithm, the range of the window size is from 3 to 21, and the iteration is 5 times. For simplicity, we will denote Kanade & Okutomi's algorithm as KOA, the moment filter approach without slope correction as MFS1, and the moment filter

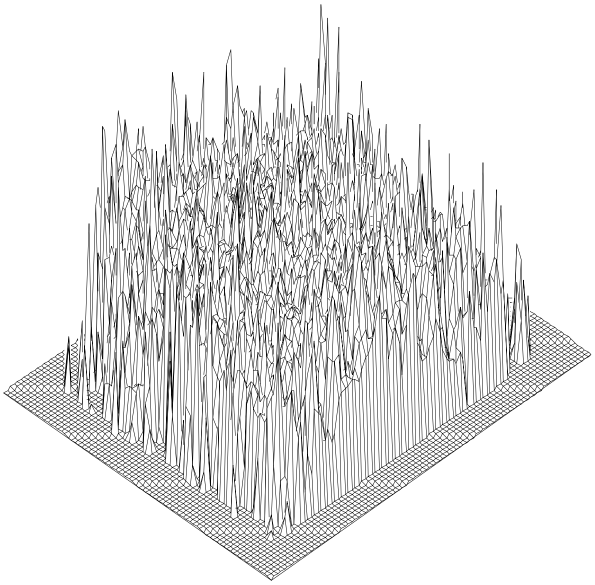


Figure 15: Blurring Difference of House From Subbarao's Algorithm

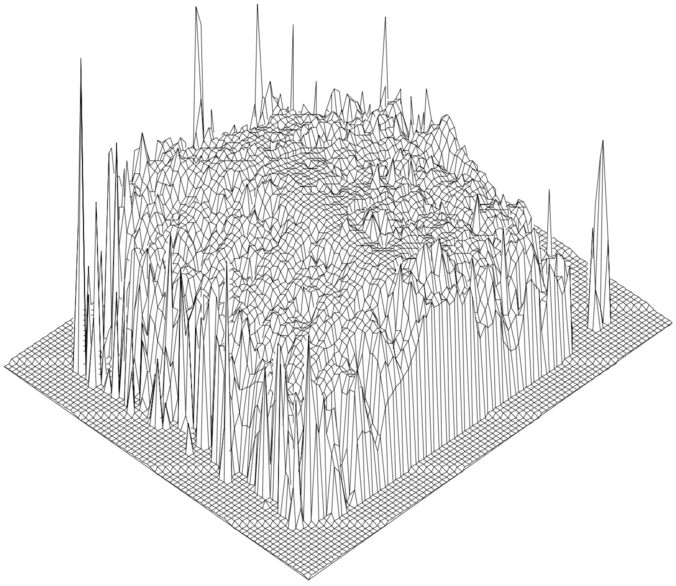


Figure 16: Blurring Difference of House From MFF1

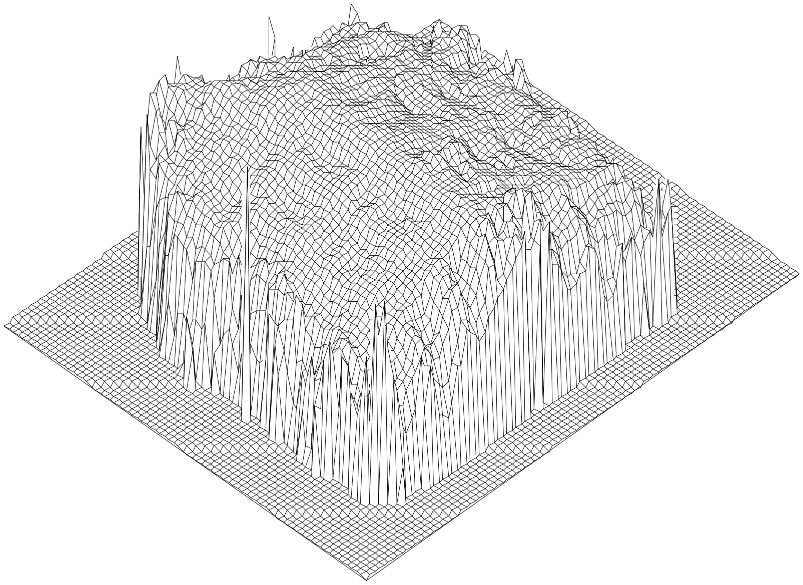
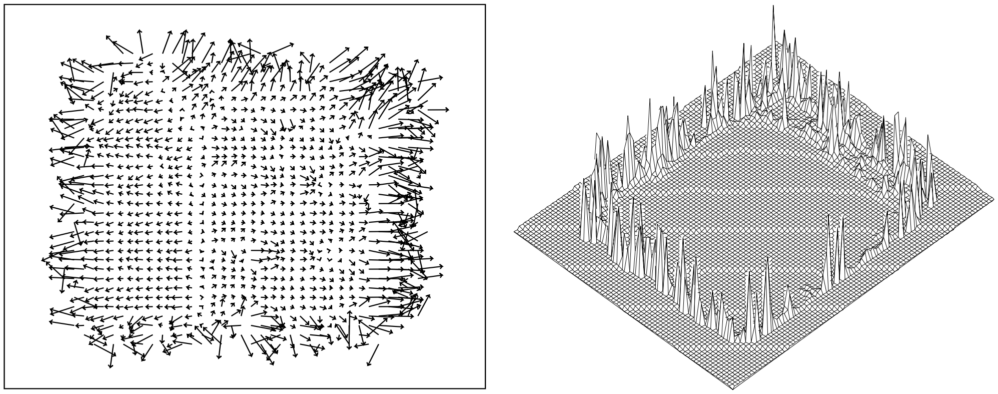


Figure 17: Blurring Difference of House From MFF2



Slope

Error Estimation

Figure 18: Slope and Error Estimations From MFF2

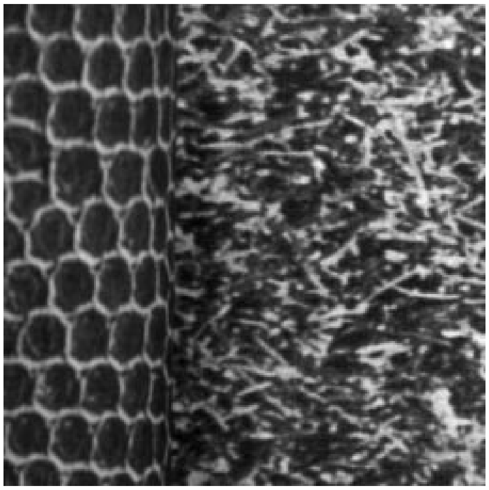


Figure 19: The Original Texture Image

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

We first test stereo algorithms on synthetic images. The first two test image pairs are obtained by artificially shifting an original image. The first image is by a uniform subpixel shift of the original one, and the second image by a nonuniform subpixel shift, which naturally results in stretching or compression. Technically the subpixel shift is achieved by linearly interpolating the original image. Figure 19, Figure 20 and Figure 21 shows the original and shifted images. The uniform shift in Figure 20 is 1.55 pixel size, and the nonuniform shift in Figure 21 increase linearly from 1.0 pixel size at zero column to 3.54 pixel size at the rightmost column. The third synthetic image pair is the 20th and 21st frames of the translating tree sequence [3] as in Figure 22. The scene is a planar surface textured by a picture of a tree. Because the ground truth of the motion is in horizontal direction, we can use them as a stereo pair. The disparity is between 1.73 and 2.26, and it increases linearly from left to right. The interpolation method for subpixel shifting is unknown.

Table 2 shows the RMS errors of disparity values computed by different algorithms. We see that for the first uniformly shifted image pair, all algorithms can do very well. Because the RMS errors in this case are 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 are significantly larger than those from the uniformly shifted pair. Figure 23 shows the disparity values of an arbitrary row for the nonuniformly shifted image pair. We see that the ripples generated by KOA are significant in this example. Furthermore, Figure 24 illustrates the slope computed by MFS2. For the tree pair, Figure 25 shows the disparity maps computed by different algorithms. And Figure 26 illustrates the slope computed by MFS2.

We tested these algorithms on two real image pairs, which were taken by the Photometrics Camera in CIL[28]. The targets were about 2.5 meters from the camera, and the camera was translated horizontally by 0.5 inch between left and right view.

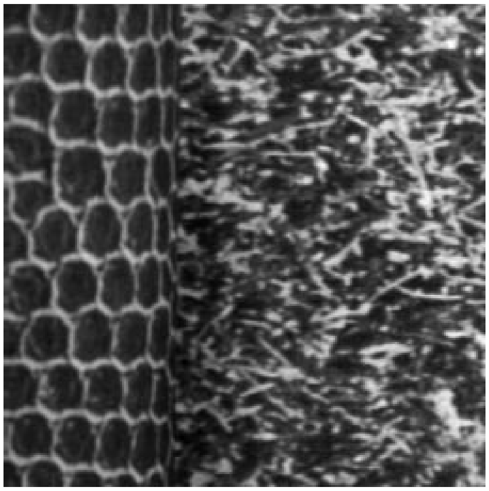


Figure 20: The Uniformly Shifted Image

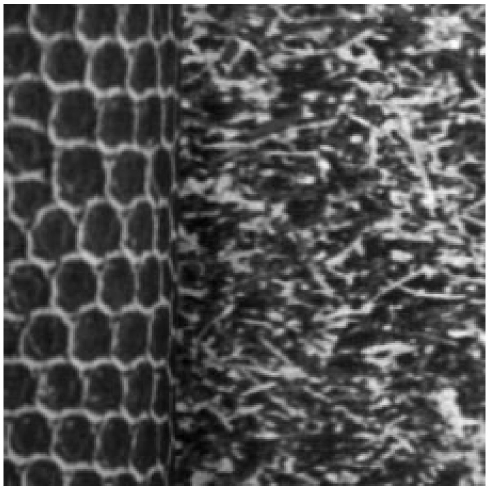


Figure 21: The Nonuniformly Shifted Image

	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

Table 2: RMS Errors of Disparity From Synthetic Images

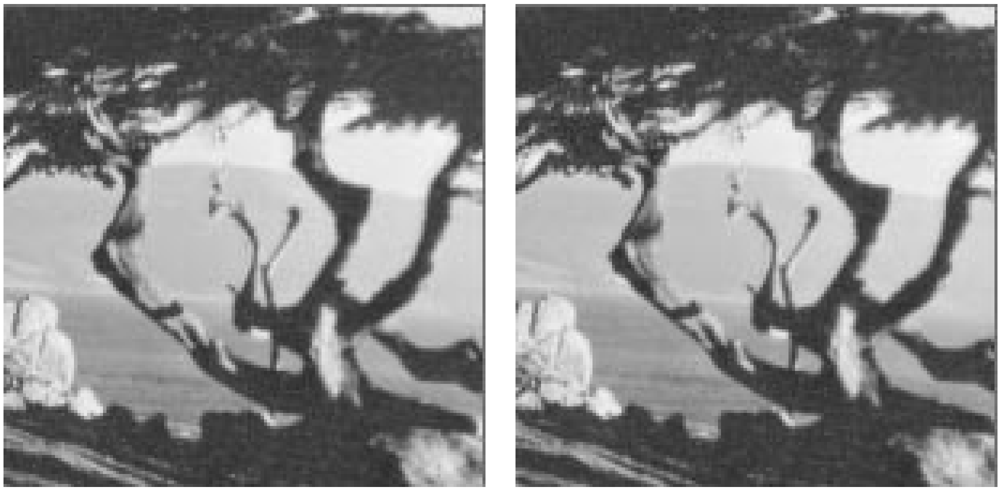


Figure 22: The 20th and 21st Frames From Translating Tree Sequence

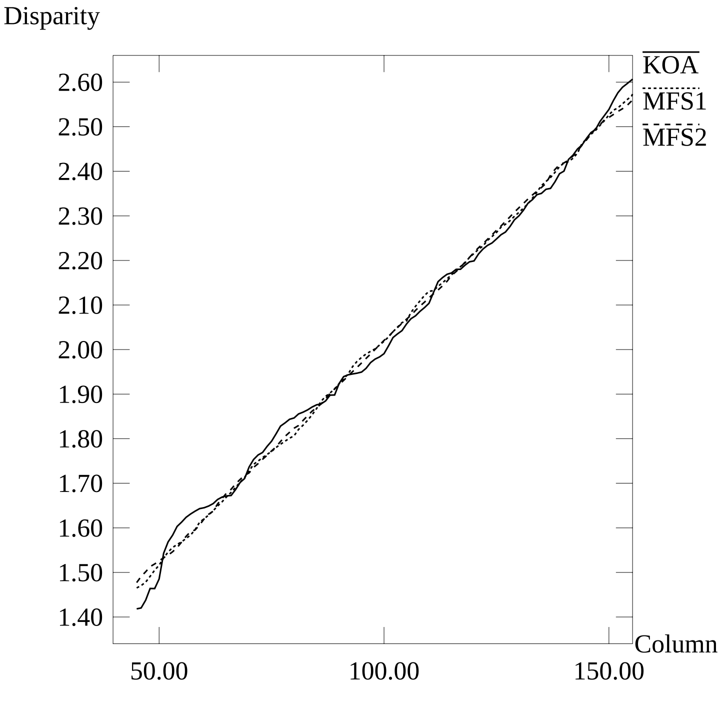


Figure 23: Computed Disparity From the Nonuniformly Shifted Pair

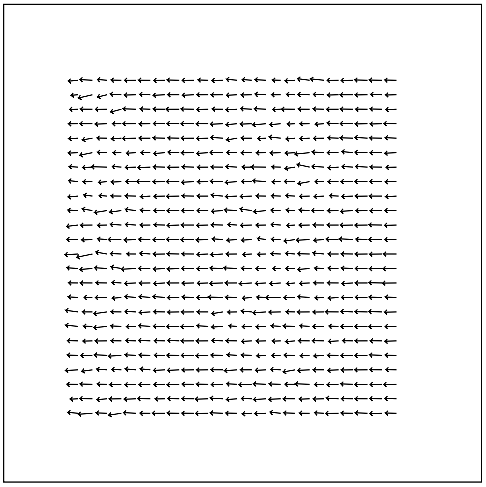
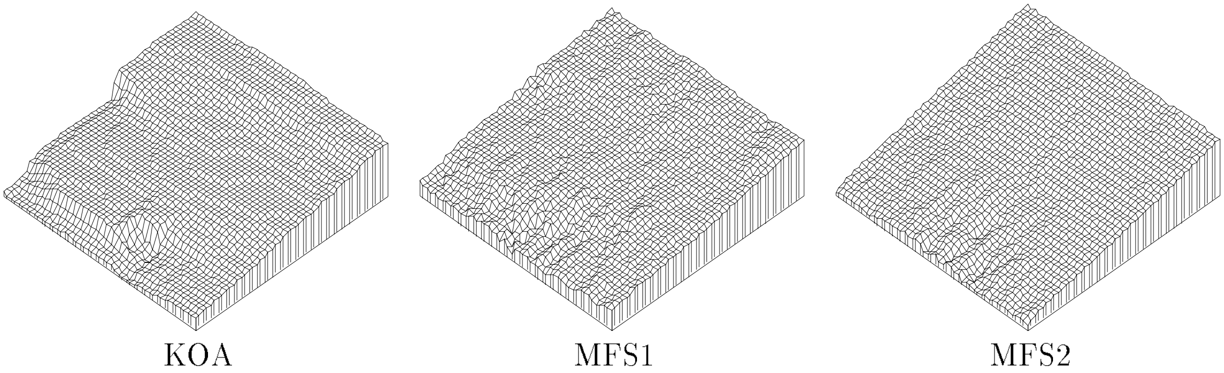


Figure 24: Slope Of The Texture Pair Computed by MFS2



KOA

MFS1

MFS2

Figure 25: Disparity Maps From the Tree Pair

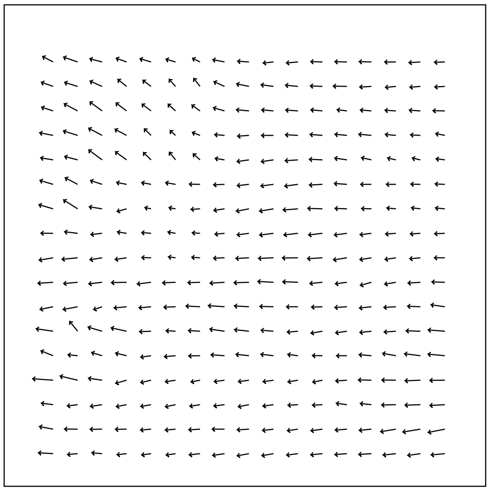


Figure 26: Slope of the Tree Pair by MFS2

	Top Patch	Left Patch	Right Patch
	(6690 Pixels)	(16025 Pixels)	(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: RMS Errors of The Cube

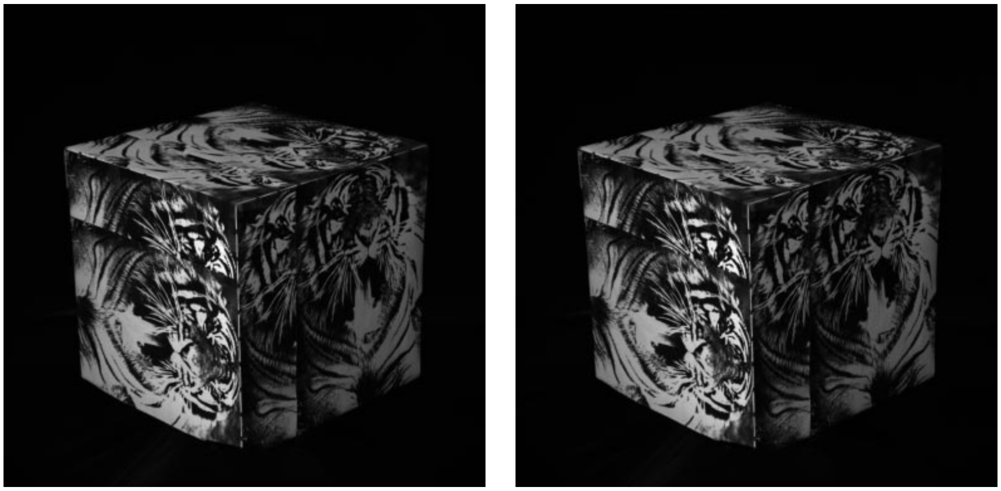


Figure 27: Images of A Textured Cube

Because there is 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 compare quantitatively the performance of stereo algorithms. Note that when we fit planar patches, we exclude those pixel near boundary so that the RMS errors we computed are not influenced by the depth discontinuities.

The first pair is images of a textured cube as in Figure 27. There are three large planar surfaces, which we denote as top patch, left patch, and right patch as in Figure 28. The difference between the largest and the smallest disparity is about 3.0 pixel widths. Figure 29, Figure 30, and Figure 31 show the computed disparity maps. We can see that the disparity map computed from KOA contains more ripples than those from MFS1 and MFS2. Furthermore, Figure 32 illustrates the computed slope from MFS2, which correctly indicates not only the direction of the slope, but also the magnitude of the slantness. Table 3 shows the RMS error after plane fittings. We observe that both MFS1 and MFS have better performance than KOA, and the RMS error of MFS2 keeps almost constant when the surface slantness increases, which we attribute to the slope correction MFS2 performed.

The second real pair is images of the same toy house we used in the focus experiments. Figure 33 shows the image pair we used for the stereo computation. There are two slightly slanted planar surfaces, which we denote as the left patch and the right patch as in Figure 34. The difference between the largest and the smallest disparity

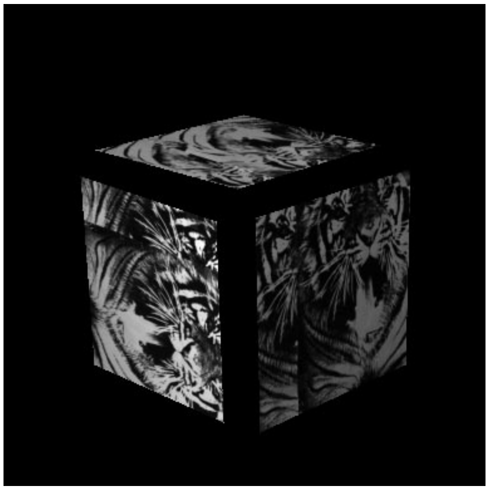


Figure 28: Planar Surfaces of The Cube

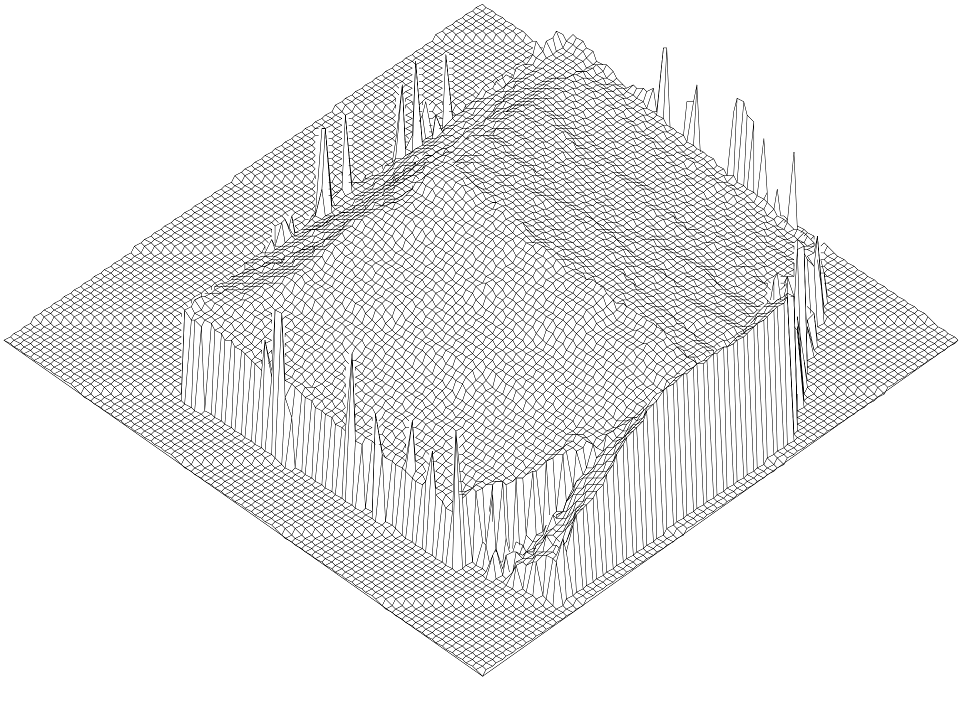


Figure 29: Disparity Map of The Cube From KOA

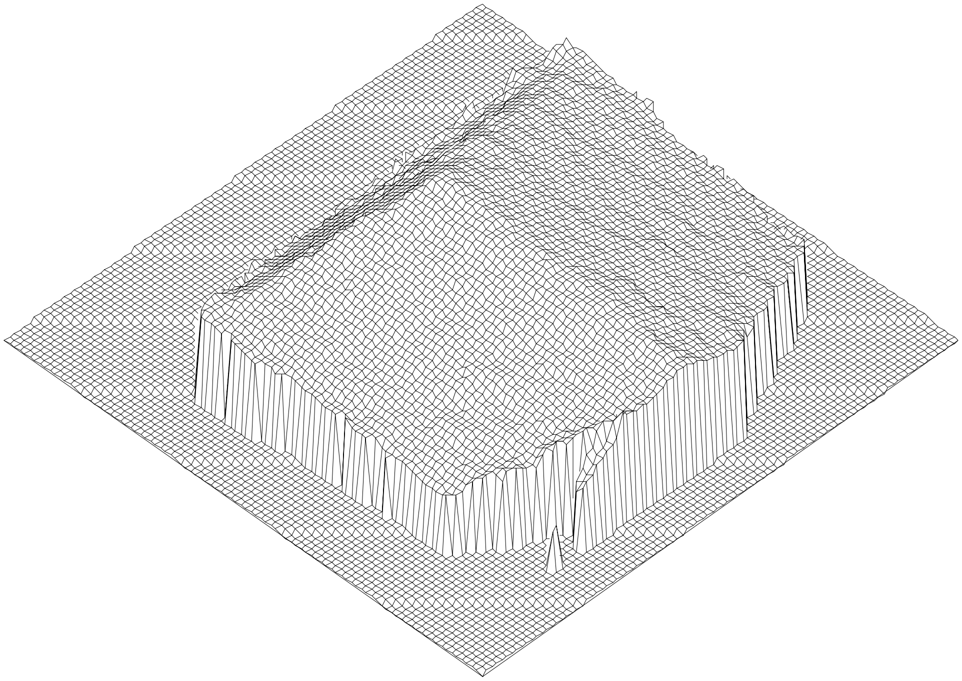


Figure 30: Disparity Map of The Cube From MFS1

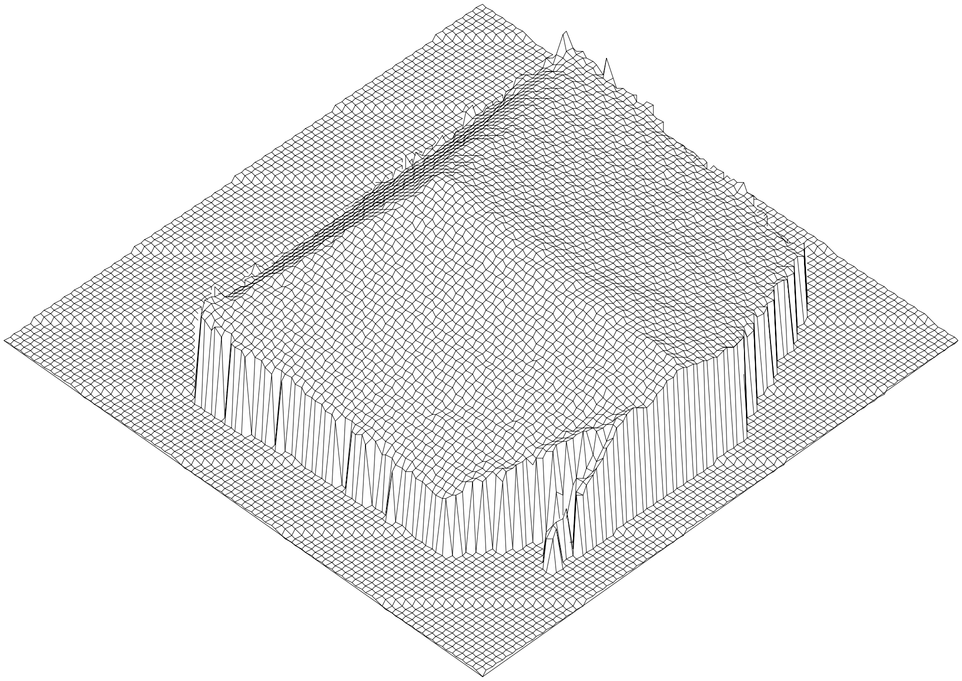


Figure 31: Disparity Map of The Cube From MFS2

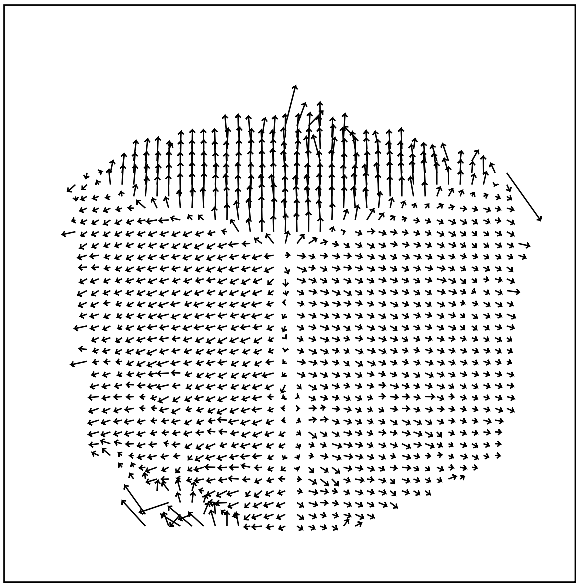


Figure 32: Slope of The Cube From MFS2

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

Table 4: RMS Errors of The Toy House

is only about 0.5 pixel size, excluding the background. Figure 35, Figure 36 and Figure 37 show the computed disparity maps from different algorithms. Again, we can see the consistent step-like ripples from KOA. Figure 38 illustrates the computed slope from MFS2, which indicates the and magnitude and direction of the slope on the surfaces, and locates the discontinuities by large slope values. Table 4 shows the RMS error after plane fittings of the two patches.

6.2 Hypergeometric Filter Approach

6.2.1 Computation of Hypergeometric Filters

Unlike most other filters, the numerical computation of a hypergeometric filter itself is a nontrivial problem. The infinite series representation of the confluent hypergeometric function in [25] (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 [19, 18] worked well most of time in computation of the hypergeometric filters with

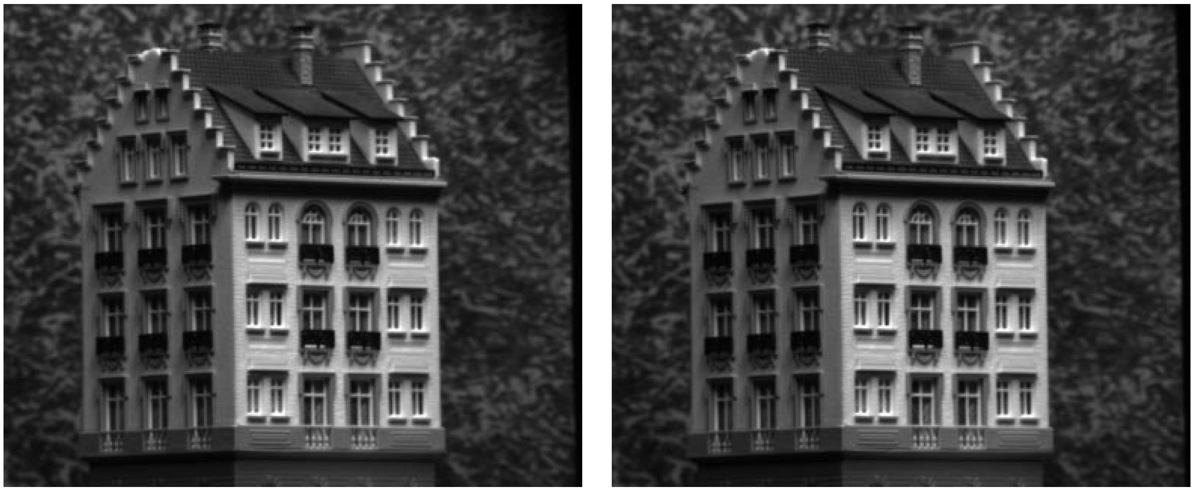


Figure 33: Stereo Images of A Toy House

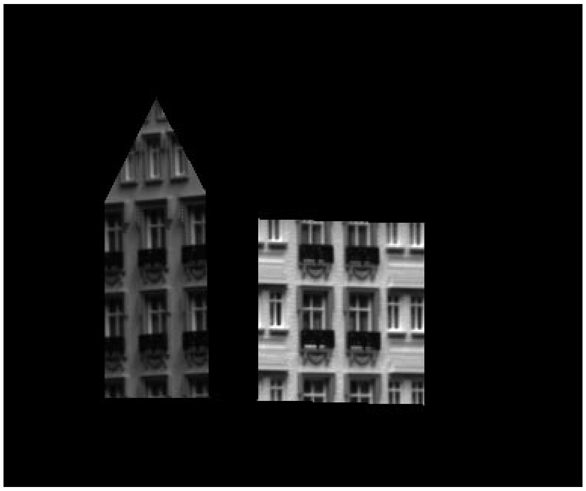


Figure 34: Planar Surfaces of The Toy House

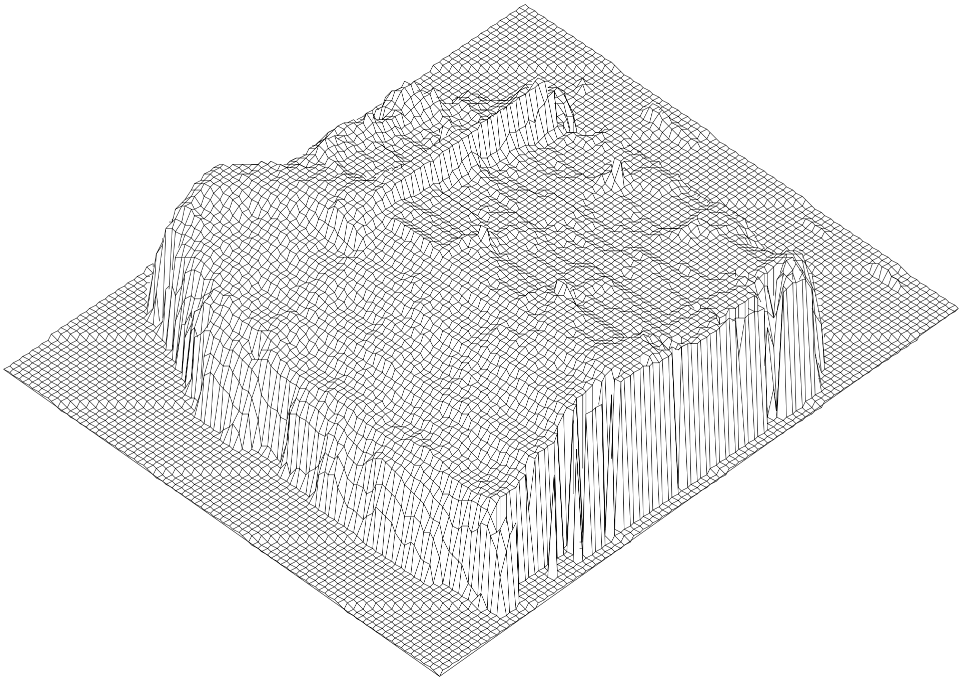


Figure 35: Disparity Map of the House From KOA

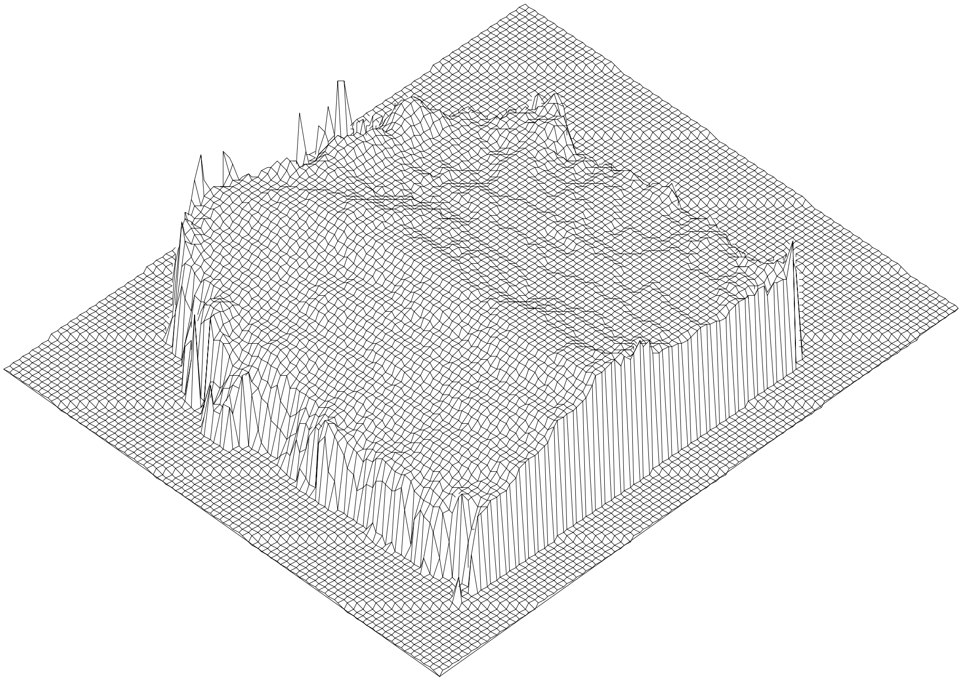


Figure 36: Disparity Map of the House From MFS1

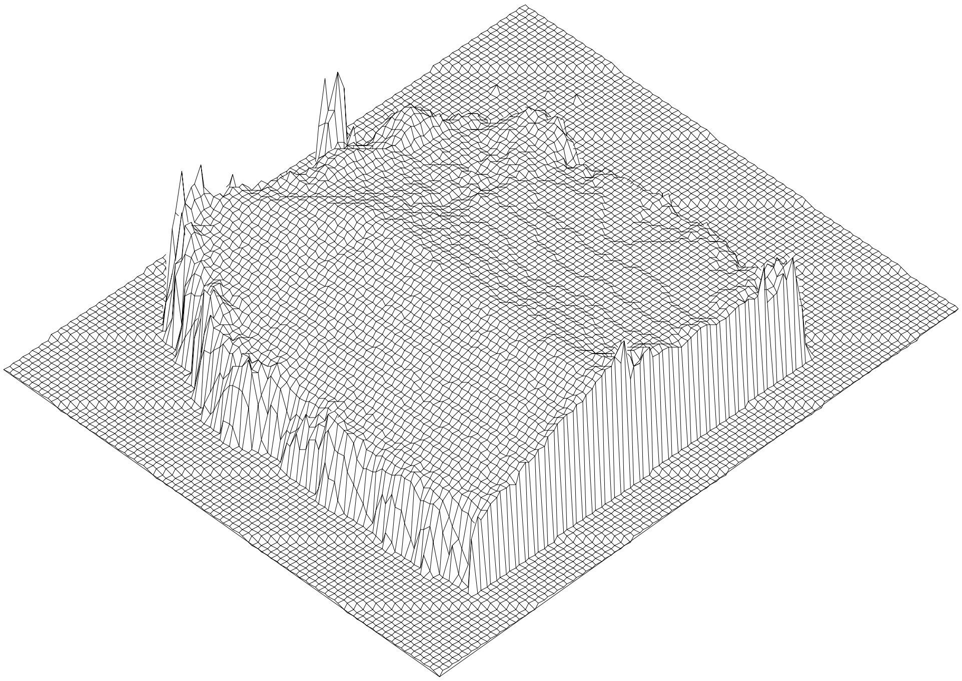


Figure 37: Disparity Map of the House From MFS2

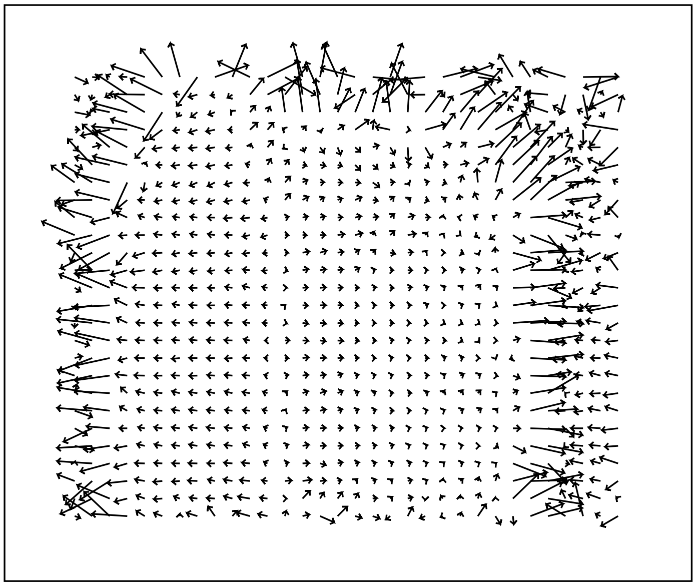


Figure 38: Slope of the House From MFS2

IP = 10. But it did fail sometimes indicated by either the numerical evaluator of the confluent hypergeometric function failed to terminate or the computed hypergeometric filter had a sharp spike. In case of its failures, we observed that applying Kummer's transformation first, and then evaluating the confluent hypergeometric function can solve this difficulty. Kummer's transformation is

$$\Phi(a, c, x) = e^x \Phi(c - a, c, -x). \tag{122}$$

Fortunately we don't have to generate the H filters online, which may be a very challenging numerical problem. Instead, we can generate the H filters according to Eq. 84, and in case the numerical algorithm fails for a filter, we use Kummer's transformation, and recompute the filter. Our experience is that, when $m < 40$, we should apply the Kummer's transformation first to Eq. 84, otherwise, we should evaluate Eq. 84 directly.

The 2D extension of the H filter is as straightforward as:

$$h_{ij}(x, y) = h_i(x) h_j(y). \tag{123}$$

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. 81, the Nyquist frequency π is reached when

$$m \approx \sigma^2 \pi. \tag{124}$$

Due to hardware limitations, we will not be able to use all the filters up to the Nyquist frequency in our implementation here. Instead, we will use the (i, j) (Eq. 123), correspondingly (f_x, f_y), contained in the shadow area in Figure 39, in which,

$$u_1 = \frac{\sqrt{M_1}}{\sigma} \tag{125}$$

$$u_2 = \frac{\sqrt{M_2}}{\sigma} \tag{126}$$

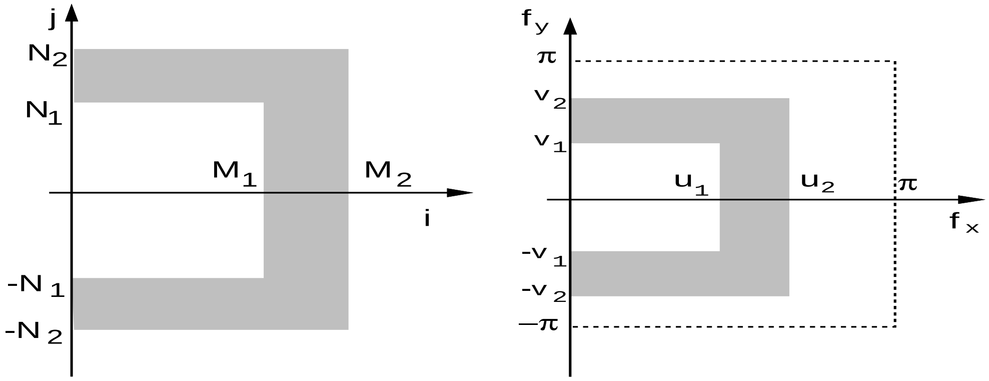
$$v_1 = \frac{\sqrt{N_1}}{\sigma} \tag{127}$$

$$v_2 = \frac{\sqrt{N_2}}{\sigma}. \tag{128}$$

Again, we believe that once we are able to fully explore the whole Fourier domain up to the Nyquist frequency by advancement of more powerful hardware, the results reported here should be improved considerably.

6.2.2 Optical Flow

We adopt the 2D conjugate gradient algorithm in [22] (function *frprmn*) to compute flow vectors. For stability of numerical computation, we actually minimize the



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

Figure 39: Frequency Sampling in Our Implementation

normalized version of $S(v_x, v_y)$ in Eq. 112,

$$S'(v_x, v_y) = \frac{S(v_x, v_y)}{\sum_m \sum_n \parallel V_{mn} \parallel^2}. \tag{129}$$

As we did in the moment filter approach, we truncate the Taylor's expansion after the third order term in Eq. 110.

There have been many algorithms to compute optical flow from two or more images. Among the published work, Barron et al [3] provided a quantitative measure for a few of the optical flow techniques. We will demonstrate the capability of our new technique on the same test images for comparison. While most of techniques presented in [3] use a sequence of images, our technique use just two adjacent frames. 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 [3],

$$\psi_e = \arccos\left(\frac{v_x v_{0x} + v_y v_{0y} + 1}{\sqrt{v_x^2 + v_y^2 + 1}\sqrt{v_{0x}^2 + v_{0y}^2 + 1}}\right), \tag{130}$$

where (v_x, v_y) is the computed optical flow, and (v_{0x}, v_{0y}) is the true optical flow. We don't use the standard deviation measure in [3] because it is hard to interpret. All angular errors are in degree ($^\circ$).

Without any post processing, the approach proposed here will certainly 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 computed will have one large eigenvalue in case of the aperture problem, and two large eigenvalues in case of no texture at all. The corresponding eigenvector should indicate the direction in which large uncertainty

Technique	Number
Horn & Schunck (original)	1
Horn & Schunck (original) $\| \nabla I \| \geq 5.0$	2
Horn & Schunck (modified)	3
Horn & Schunck (modified) $\| \nabla I \| \geq 5.0$	4
Lucas & Kanade ($\lambda_2 \geq 1.0$)	5
Lucas & Kanade ($\lambda_2 \geq 5.0$)	6
Uras et al. (unthresholded)	7
Uras et al. ($\det(H) \geq 1.0$)	8
Nagel	9
Nagel $\| \nabla I \|_2 \geq 5.0$	10
Anandan	11
Singh (step 1, $n = 2, w = 2$)	12
Singh (step 1, $n = 2, w = 2, \lambda_1 \leq 5.0$)	13
Singh (step 2, $n = 2, w = 2$)	14
Singh (step 2, $n = 2, w = 2, \lambda_1 \leq 0.1$)	15
Heeger	16
Heeger (level 0)	17
Heeger (level 1)	18
Heeger (level 2)	19
Waxman et al. ($\sigma_f = 2.0$)	20
Fleet & Jepson ($\tau = 1.0$)	21
Fleet & Jepson ($\tau = 1.25$)	22
Fleet & Jepson ($\tau = 2.5$)	23

Table 5: Numbering the Optical Flow Techniques

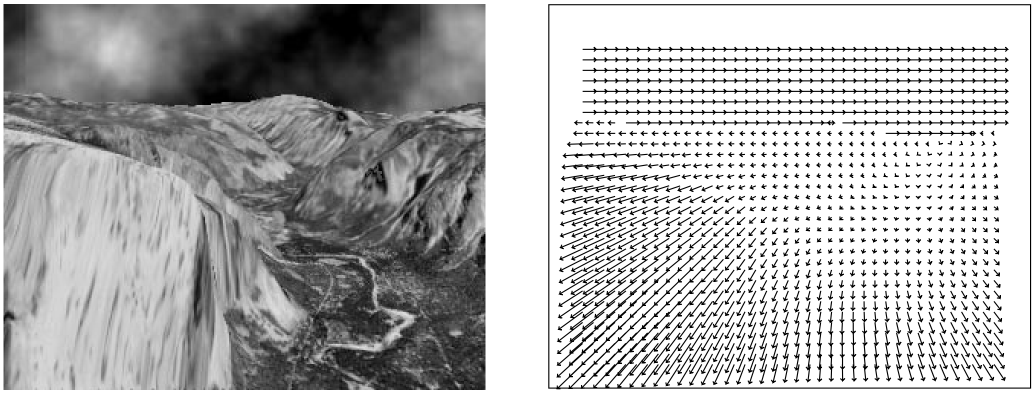
occurs. To compare our results with those from other techniques, we threshold the computed optical flow by the following formulae:

$$\frac{E_{\max}}{v_x^2 + v_y^2 + 1} < T_e, \tag{131}$$

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 convenience of comparisons, we number the various techniques implemented in [3] as in Table 5.

For the Yosemite sequence (Figure 40), we use the 9th and 10th frames to compute the optical flow as they are the two frames in the middle of the sequence. The size of the H filters is $\sigma = 7.0$, and the sampling limits in Figure 39 are: $M_1 = N_1 = 0, M_2 = N_2 = 10$, which correspond to the area in the Fourier domain: $(0.0 \leq f_x \leq 0.45, -0.45 \leq f_y \leq 0.45)$. In other words, we only use information whose frequency is lower than one-sixth of the Nyquist frequency. Figure 41 shows the raw



The Ninth Frame in Yosemite Sequence

The Correct Flow

Figure 40: Yosemite Image Sequence

(unthresholded) optical flow computed. Surprisingly, the movement of the clouds is captured accurately. The curve of the density versus average error is shown in Figure 42 along with the results reported in [3]. For further comparison, we listed the average error and the corresponding density in Table 6. The numbers in Figure 42 are the numbered techniques in Table 5. We can see that except for a couple of techniques which impose smoothness constraint at 100% density, our approach has the best results for all densities, despite the fact that we only use low frequency information from only two frames.

The monotonic increase of the average error with respect to more restrictive eigenvalue threshold indicates that the error estimation indeed represented the uncertainty properly. Furthermore, as a covariance matrix can be represented by a 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 43. First of all, from the size of ellipses, it indicates that the top portion and low-left portion have large errors which is consistent with Figure 41 and Figure 40. Also we observe that at locations where the aperture problem is obvious, e.g. near a single edge, the ellipses are elongated in the direction of edges, which represent the lack of information in those directions.

The second synthetic sequence is the translating-tree sequence (Figure 44) from [3]. 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 39 are: $M_1 = N_1 = 0, M_2 = N_2 = 15$, 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 is one-third of the Nyquist frequency. Figure 45 shows the unthresholded optical flow. Except sparse locations where there are no texture, the computed optical flow is pretty consistent with the ground truth. The curve of the density versus average error is shown in Figure 46, in which some results with large errors reported in [3] are not included, and and Table 7. Again we can see that our approach based upon only two adjacent frames and only low frequency information outperformed all the

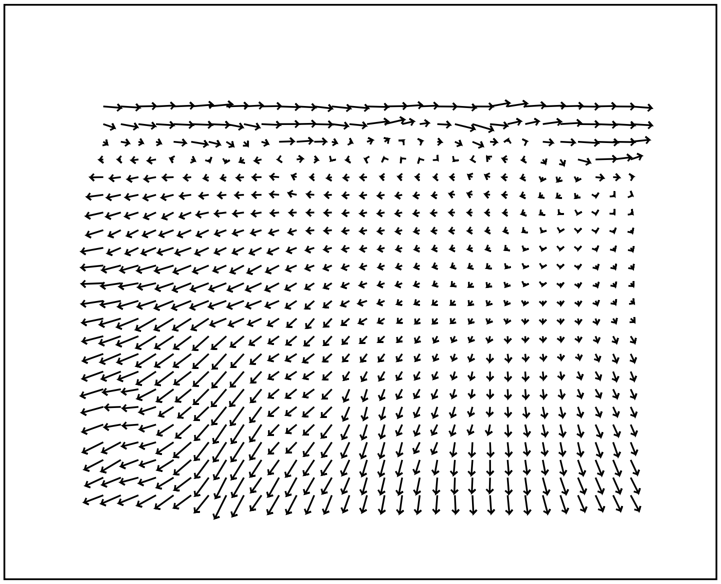


Figure 41: Computed Optical Flow of the Yosemite Scene

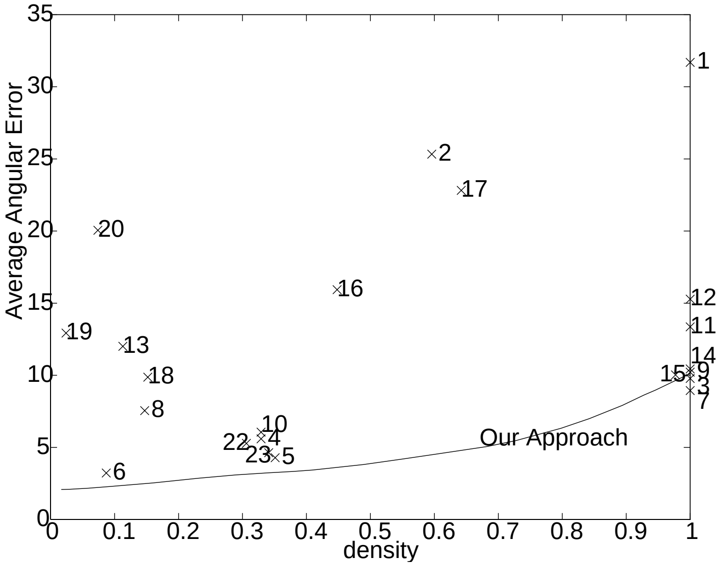


Figure 42: Average Error For the Yosemite Scene

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 6: Density vs. Average Error of Yosemite Scene

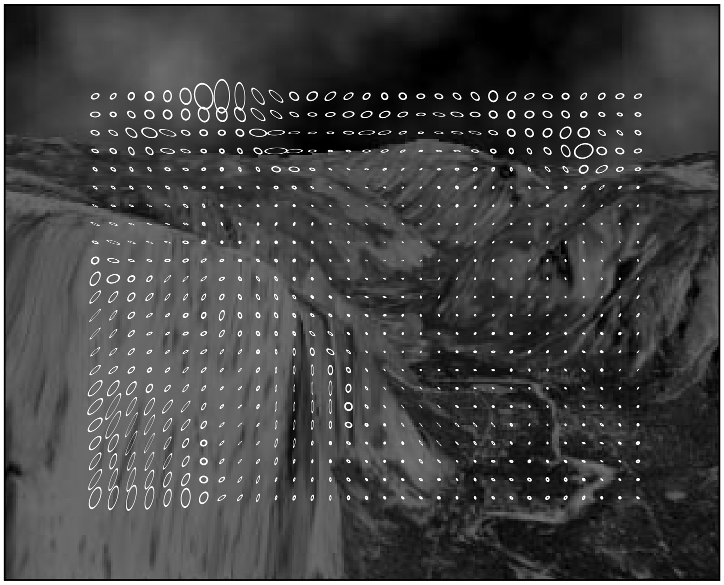
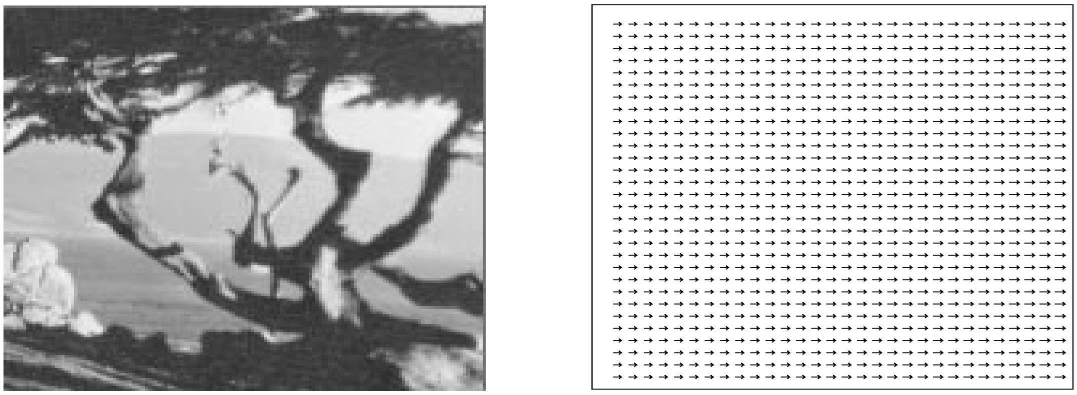


Figure 43: Uncertainty Estimation in Yosemite Scene



The 20th Frame in Translating-Tree Sequence

The Correct Flow

Figure 44: Translating Tree Image Sequence

algorithms at all densities.

Figure 47 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, it also represented large errors caused by blank areas between trunks.

A more tricky synthetic sequence is the diverging tree sequence from [3, 7] as illustrated in Figure 48. The significant difference between this sequence and previous sequences is that the underlying optical flow has a very high 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 and the original optical flow is 1.37° comparing 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 should be much larger than 1.37°. While some results reported in [3] is very close or even smaller than 1.37°, we believe that the only reason those algorithms can reach such a precision is due to the averaging in the temporal direction. Actually, as the authors pointed out in [3], the errors of the differential approaches went up substantially when only two images are used. And because the temporal averaging is crucial in this sequence, we argue that the comparisons made in [3] (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 8 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° can only be achieved by averaging along the temporal direction. Secondly, the average error of our algorithm after averaging the fourteen optical flow fields is smaller than all the results[1]ll results reported in [3] use at least fifteen frames except those from Fleet &

[1] A

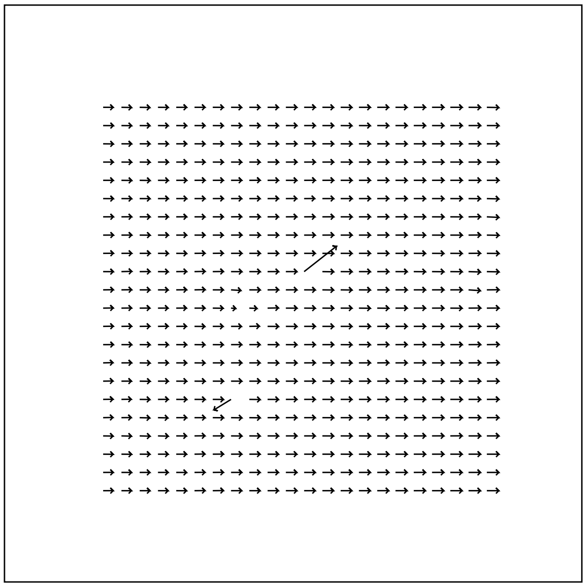


Figure 45: Computed Optical Flow of the Translating Tree Scene

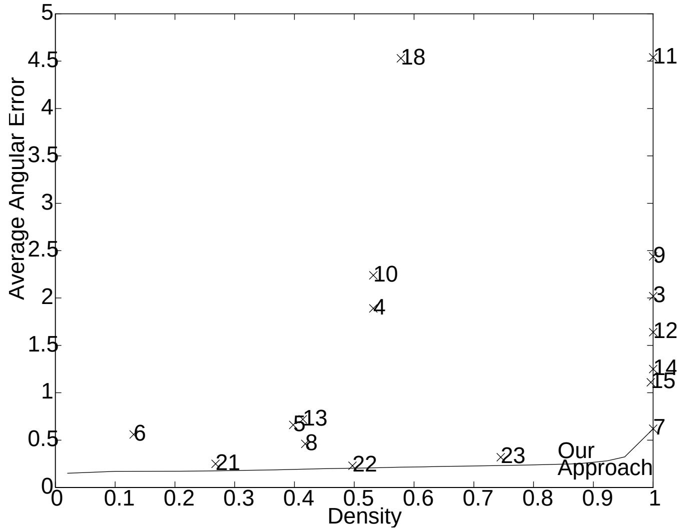


Figure 46: Average Error For the Translating Tree Scene

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 7: Density vs. Average Error of Translating Tree Scene

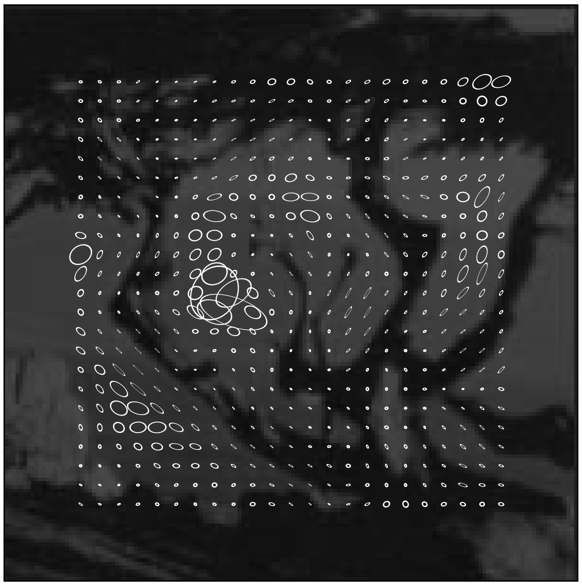
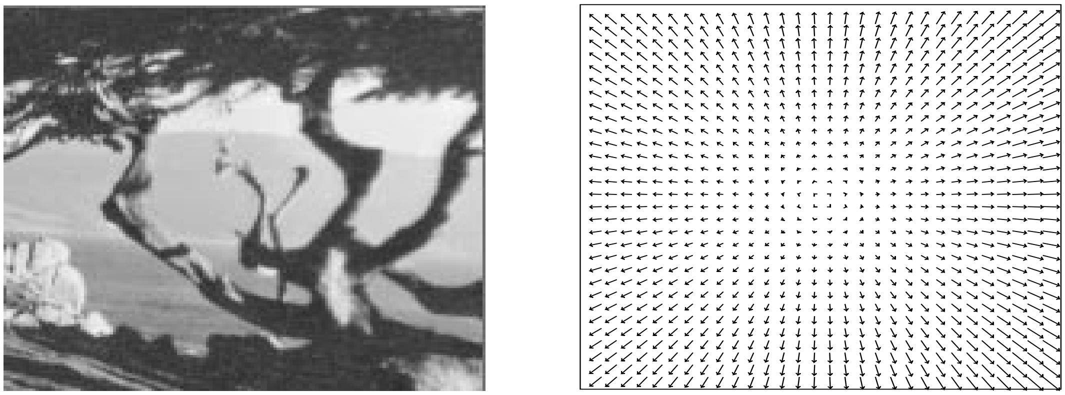


Figure 47: Uncertainty Estimation in Translating Tree Scene



The 20th Frame in Diverging-Tree Sequence The Correct Flow

Figure 48: The Diverging Tree Image Sequence

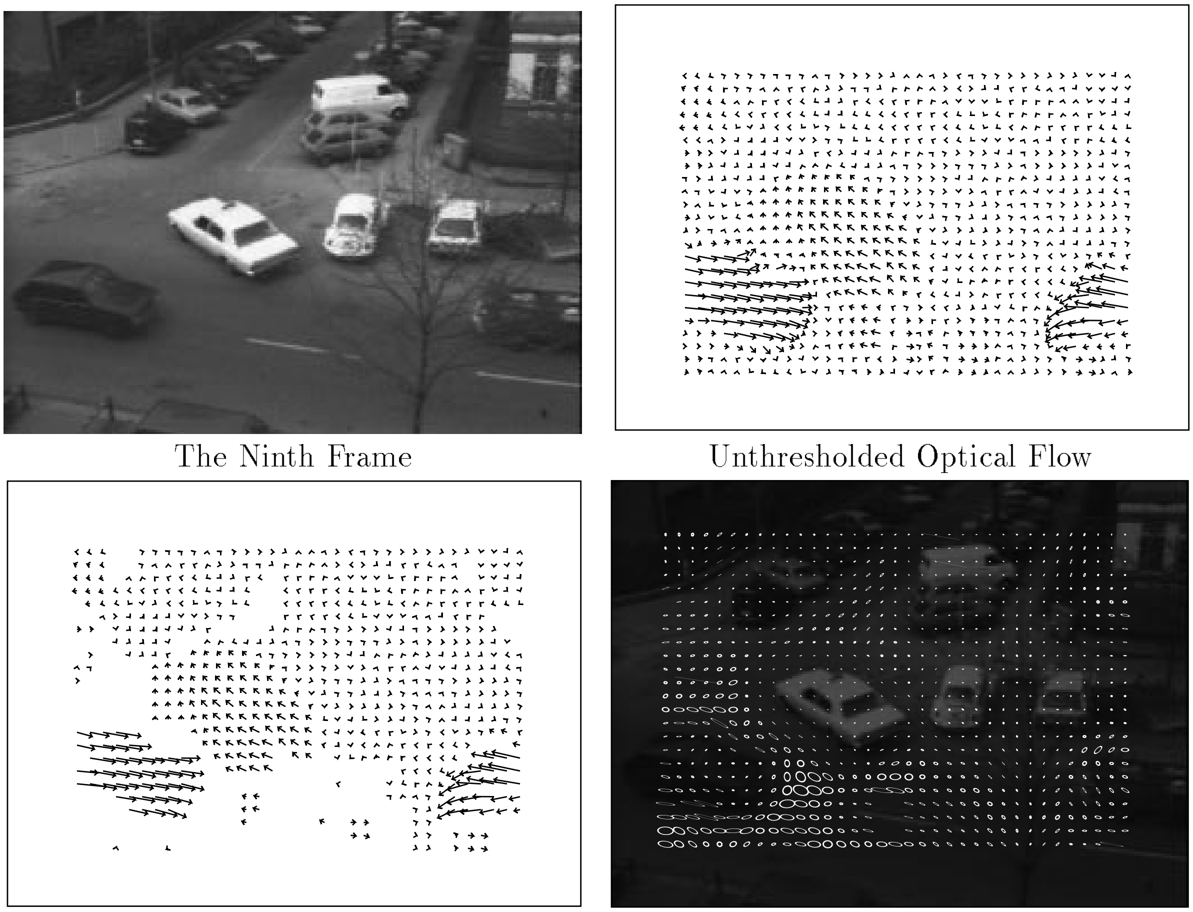
Density	Average Error From Two Frames	Average Error From Fifteen Frames
10%	2.191357	1.629412
20%	2.188124	1.595904
30%	2.292716	1.626959
40%	2.386003	1.647144
50%	2.483744	1.696696
60%	2.570150	1.724406
70%	2.686011	1.784787
80%	2.825161	1.863550
90%	3.038625	1.996772
100%	3.408894	2.195650

Table 8: Average Error of The Diverging Tree Sequence

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

We also tested our algorithm on the four real image sequences in [3]. For all the real image sequences, we take two frames in the middle of each sequence to compute optical flows. The size of the H filters $\sigma = 4.5$, and the sampling limit in Figure 39 is $M_1 = N_1 = 0, M_2 = N_2 = 15$, which correspond to the area in the Fourier domain $(0.0 \leq f_x \leq 0.86, -0.86 \leq f_y \leq 0.86)$. For each experiment, we show four images: the middle frame in the sequence, the unthresholded optical flow, thresholded optical flow and the uncertainty estimation.

The first sequence is a scene of a intersection. Three cars are moving. The motions



The Ninth Frame

Unthresholded Optical Flow

Thresholded Optical Flow $T_e = 0.005$

The Uncertainty Estimation

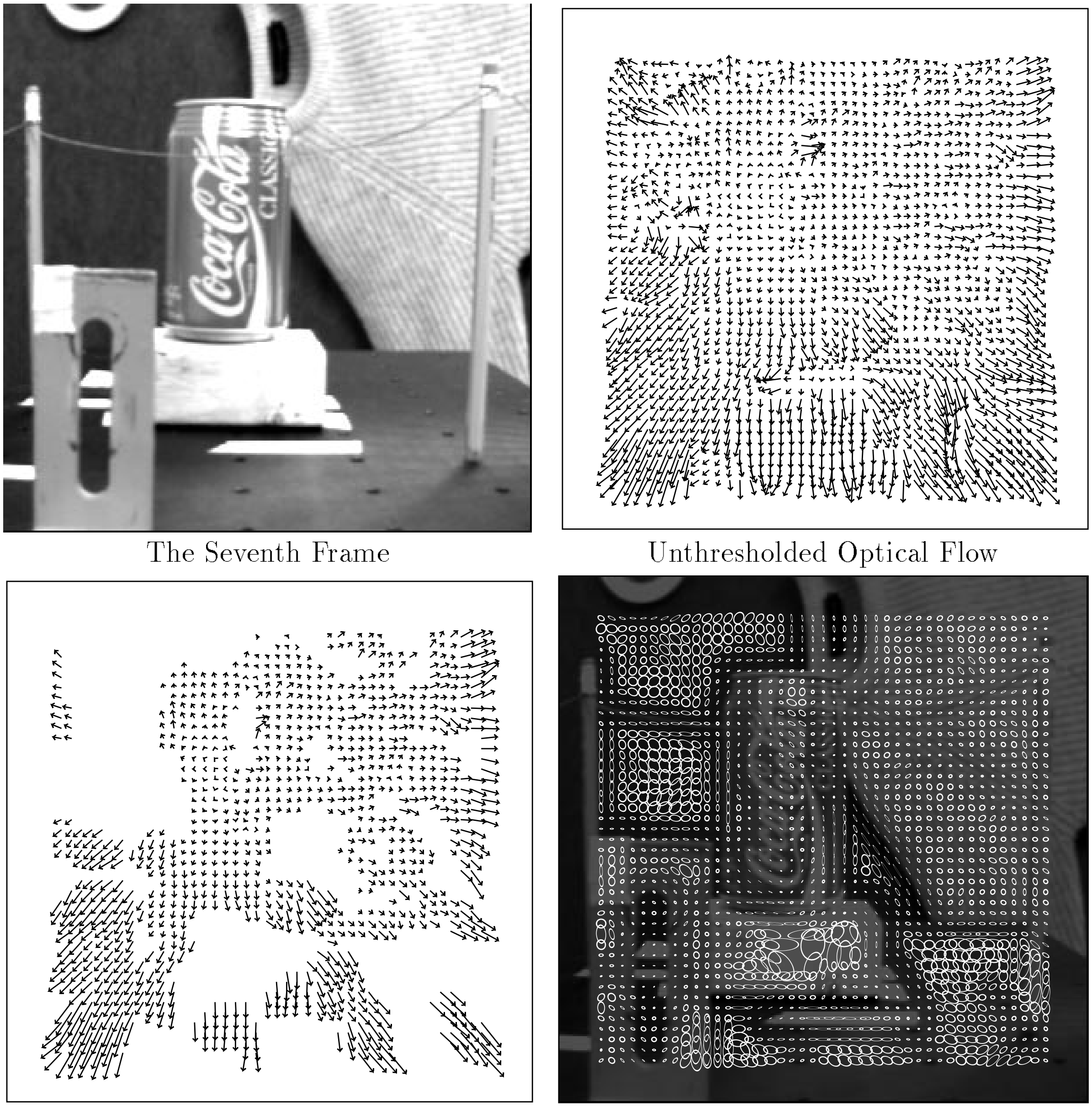
Figure 49: Optical Flow of The Taxi Sequence

of the two black cars are difficult to detect using traditional approaches since the contrasts are low. The white lines in the lower part of the image cause aperture problems which are also detected in the uncertainty estimation by extremely narrow and long ellipses. The second sequence is a static scene and the camera moves toward the coke can in the middle of the image. The aperture problem and weak texture problem happen almost everywhere as we can see from the uncertainty estimation. The third sequence is a scene of trees and the camera moves horizontally. And the fourth sequence is a scene of a rotating Rubic cube and a platform.

Though there is no objective measure to compare our results with those reported in [3], we can still conclude that in average the performance of our algorithm is superior to those in [3] in both density and precision by looking at them, despite the fact that we only used two frames in each sequence.

Acknowledgement

This research was sponsored by the Avionics Lab, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U. S. Air Force, Wright-



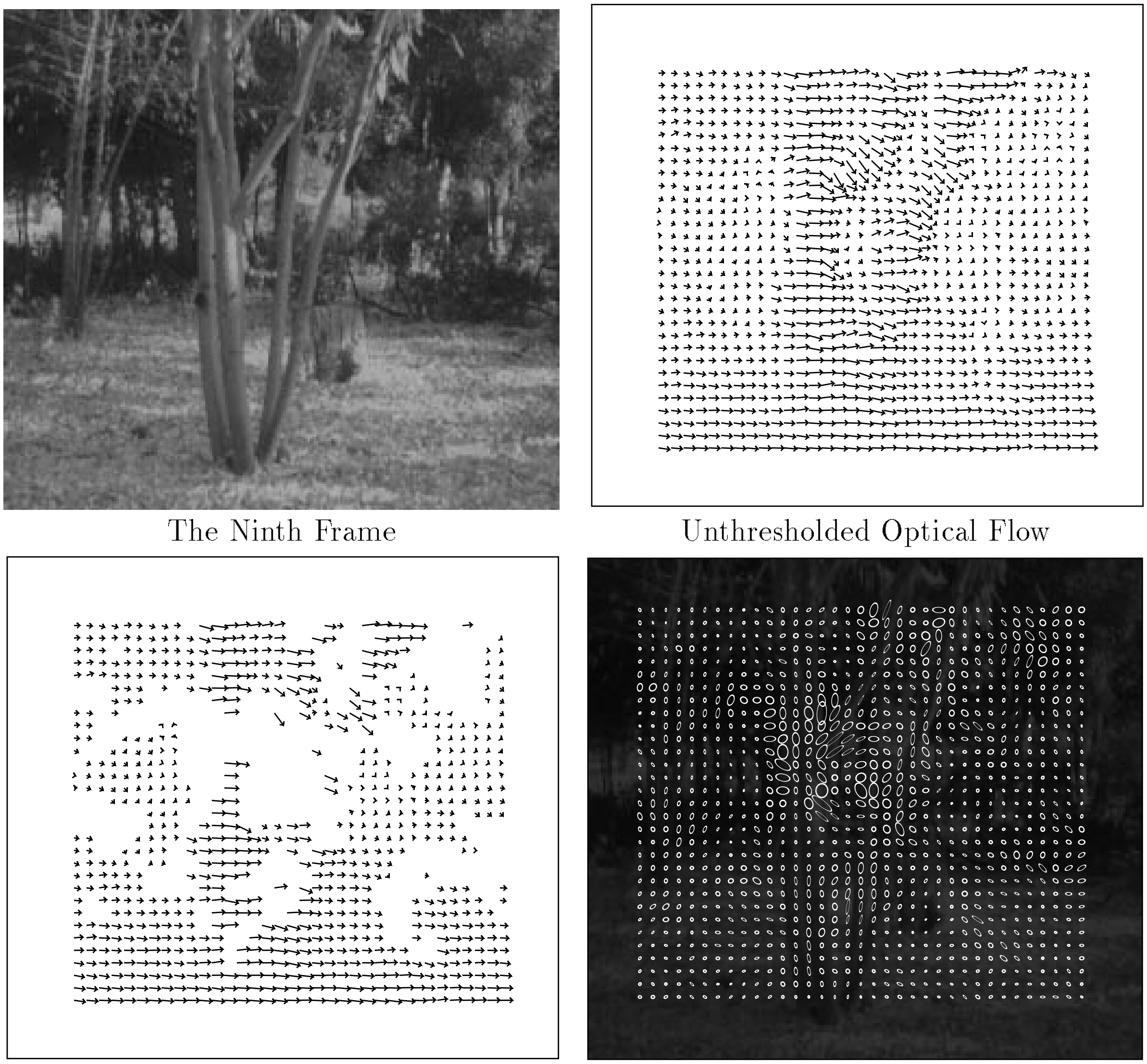
The Seventh Frame

Unthresholded Optical Flow

Thresholded Optical Flow $T_e = 0.001$

The Uncertainty Estimation

Figure 50: Optical Flow of The NASA Coke Sequence



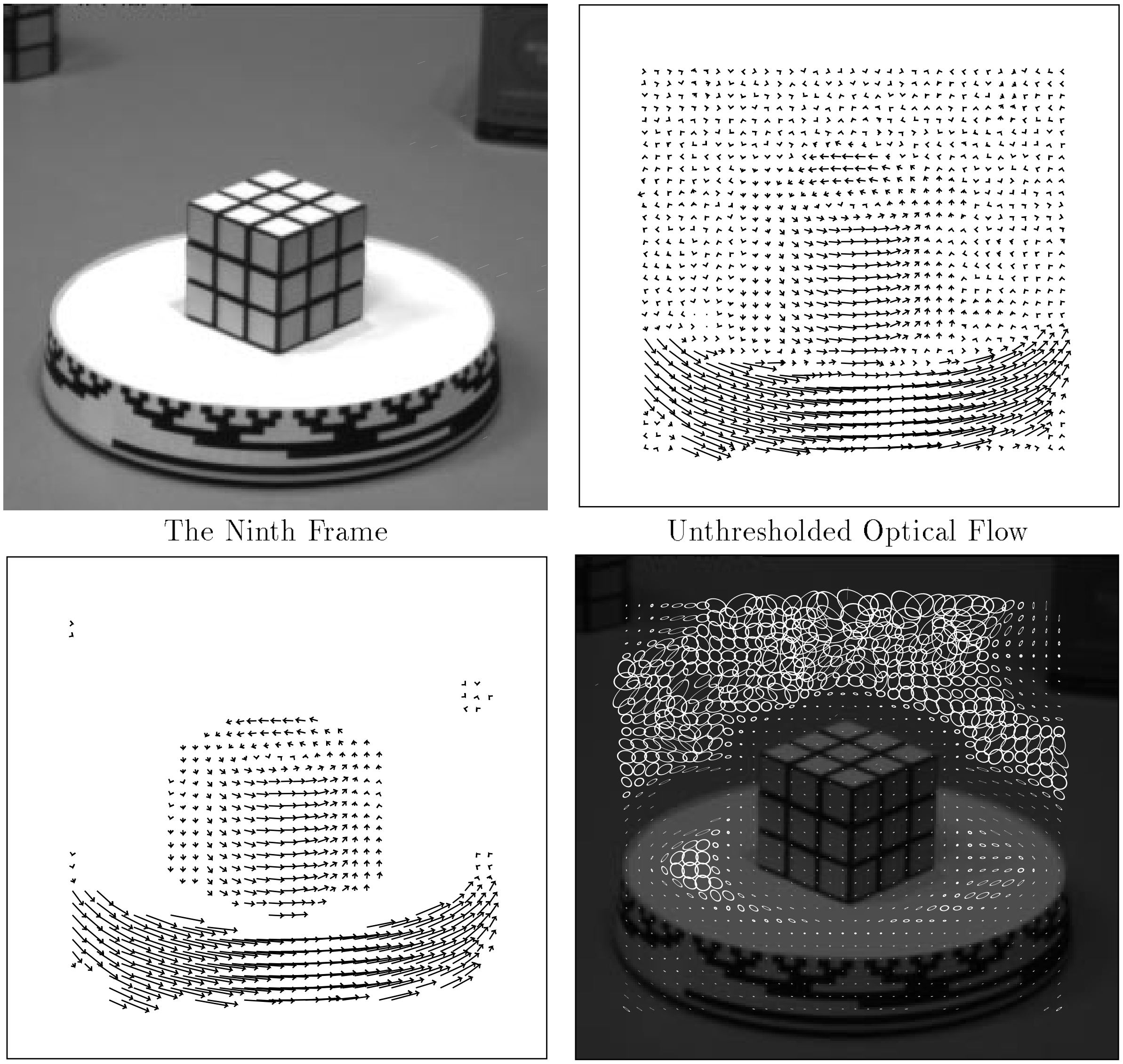
The Ninth Frame

Unthresholded Optical Flow

Thresholded Optical Flow $T_e = 0.001$

The Uncertainty Estimation

Figure 51: Optical Flow of The SRI Tree Sequence



The Ninth Frame

Unthresholded Optical Flow

Thresholded Optical Flow $T_e = 0.001$

The Uncertainty Estimation

Figure 52: Optical Flow of The Rubic Sequence

Patterson AFB, OH 45433-6543 under Contract F33615-90-C-1465, ARPA Order No. 7597. The views and conclusions are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of DARPA or the U. S. Government.

A Orthogonality and Completeness of Moment Filters

At first, let's rephrase the theorems from [20] (page 23, page 29, page 59) and list them as following:

Theorem A.1 *The generalized Laguerre's polynomial $L(n, \alpha, x)$ is the solution of the following differential equation:*

$$xy'' + (1 + \alpha - x)y' + ny = 0. \tag{132}$$

Theorem A.2 *If*

$$x^{k+1+\alpha} e^{-x}|_{x=a,b} = 0, (k = 0, 1, ...), \tag{133}$$

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.

$$\int_a^b L(n, \alpha, x)L(m, \alpha, x)x^\alpha e^{-x} dx = 0, when (m \neq n). \tag{134}$$

Theorem A.3 *Let $f(x)$ be continuous on $a < x < b$ and have a piecewise continuous derivative in this interval, If the integrals*

$$\int_a^b f^2(x)x^\alpha e^{-x} dx, \quad \textstyle\int_a^b [f'(x)]^2 x^{1+\alpha} e^{-x} dx \tag{135}$$

converge, then the following expansion is valid:

$$f(x) = \sum_{n=0}^{\infty} c_n L(n, \alpha, x), \tag{136}$$

where,

$$c_n = \frac{\int_a^b f(x)L(n, \alpha, x)x^\alpha e^{-x} dx}{\int_a^b L(n, \alpha, x)L(n, \alpha, x)x^\alpha e^{-x} dx}. \tag{137}$$

Let's set $f_0 = 0$ in Eq. 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

$$< k_{2n}(x), k_{2m+1}(x) >=< k_{2m+1}(x), k_{2n}(x) >= 0. \tag{138}$$

And

$$\begin{aligned} & < k_{2n}(x), k_{2m}(x) > \\ = \; & \frac{2^{m+n}m!n!}{\sqrt{2\pi}\sigma^{2(m+n)+1}} \int_{-\infty}^{+\infty} L(n, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}} dx \\ = \; & \frac{2^{m+n}m!n!}{\sqrt{2\pi}\sigma^{2(m+n)+1}} \frac{\sigma}{\sqrt{2}} \int_{0}^{+\infty} L(n, -\frac{1}{2}, t) L(m, -\frac{1}{2}, t) t^{-\frac{1}{2}} e^{-t} dt, \end{aligned} \tag{139}$$

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

$$< k_{2n}(x), k_{2m}(x) >= 0, \text{when}(m \neq n). \tag{140}$$

Similarly

$$\begin{aligned} & < k_{2n+1}(x), k_{2m+1}(x) > \\ = \; & -\frac{2^{m+n}m!n!}{\sqrt{2\pi}\sigma^{2(m+n)+5}} \int_{-\infty}^{+\infty} x^2 L(n, \frac{1}{2}, \frac{x^2}{2\sigma^2}) L(m, \frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}} dx \\ = \; & -\frac{2^{m+n}m!n!}{\sqrt{2\pi}\sigma^{2(m+n)+2}} \sqrt{2} \int_{0}^{+\infty} L(n, \frac{1}{2}, t) L(m, \frac{1}{2}, t) t^{\frac{1}{2}} e^{-t} dt \\ = \; & 0, \text{when}(m \neq n). \end{aligned} \tag{141}$$

From Eq. 138, Eq. 140 and Eq. 141, we prove the orthogonality of the moment filters, i.e.

$$< k_i(x), k_j(x) >= 0, \text{when}(i \neq j). \tag{142}$$

For any function $\xi(x)$, it can always be decomposed into sum of an odd function $\xi_o(x)$ and an even function $\xi_e(x)$ as,

$$\begin{aligned} \xi_o(x) &= \frac{1}{2}(\xi(x) - \xi(-x)) \\ \xi_e(x) &= \frac{1}{2}(\xi(x) + \xi(-x)) \\ \xi(x) &= \xi_e(x) + \xi_o(x). \end{aligned} \tag{143}$$

From Theorem A.3, in the interval $(0, +\infty)$, for any function $g(t)$ which satisfies Eq. 135, we have

$$t^{-1/2} g(t) = \sum_{n=0}^{\infty} c_n L(n, \frac{1}{2}, t), \tag{144}$$

$$g(t) = \sum_{m=0}^{\infty} d_m L(m, -\frac{1}{2}, t). \tag{145}$$

Therefore, let $t = \frac{x^2}{2\sigma^2}$, we have

$$g(\frac{x^2}{2\sigma^2}) = \sqrt{2}\sigma \sum_{n=0}^{\infty} c_n x L(n, \frac{1}{2}, \frac{x^2}{2\sigma^2}), \tag{146}$$

$$g(\frac{x^2}{2\sigma^2}) = \sum_{m=0}^{\infty} d_m L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}). \tag{147}$$

Because $g(t)$ can be any function, we can let $g(\frac{x^2}{2\sigma^2}) = \xi_o(x)$ in Eq. 146, and $g(\frac{x^2}{2\sigma^2}) = \xi_e(x)$ in Eq. 147. Replacing them into Eq. 143, in $(0, \infty)$ we have

$$\begin{aligned}\xi(x) &= \sqrt{2}\sigma \sum_{n=0}^{+\infty} c_n x L(n, \frac{1}{2}, \frac{x^2}{2\sigma^2}) + \sum_{m=0}^{+\infty} d_m L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) \\ &= \sum_{i=0}^{+\infty} \frac{w_i}{< k_i(x), k_i(x) >} k_i(x), \qquad (148)\end{aligned}$$

where, when $i = 2m$,

$$\begin{aligned}w_{2m} &= \frac{\sigma^{2m}}{2^m m!} < k_{2m}(x), k_{2m} > d_m \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \frac{2^m m!}{\sigma^{2m}} L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}} dx \\ &\quad \frac{\int_0^{+\infty} \xi_e(x) L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) (\frac{x^2}{2\sigma^2})^{-1/2} e^{-\frac{x^2}{2\sigma^2}} d(\frac{x^2}{2\sigma^2})}{\int_0^{+\infty} L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) (\frac{x^2}{2\sigma^2})^{-1/2} e^{-\frac{x^2}{2\sigma^2}} d(\frac{x^2}{2\sigma^2})} \\ &= \frac{2^m m!}{\sigma^{2m}} \frac{\sqrt{2}}{\sqrt{\pi}\sigma} \int_0^{+\infty} \xi_e(x) L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}} dx \\ &= \frac{2^m m!}{\sigma^{2m}} \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{+\infty} \xi_e(x) L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}} dx \\ &= \frac{2^m m!}{\sigma^{2m}} \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{+\infty} \xi(x) L(m, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}} dx \\ &= < \xi(x), k_{2m}(x) > . \qquad (149)\end{aligned}$$

Similarly when $i = 2n + 1$,

$$w_{2n+1} =< \xi(x), k_{2n+1}(x) > . \qquad (150)$$

Similarly in interval $(-\infty, 0)$, we also have

$$w_i =< \xi(x), k_i(x) > . \qquad (151)$$

B Properties of Moment Filters

B.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

$$L(n, \alpha, x) = \sum_{m=0}^{n} \frac{, (n + \alpha + 1)(-1)^m}{, (m + \alpha + 1) m! (n - m)!} x^m. \qquad (152)$$

Therefore, we have

$$\begin{aligned}
& L(n,1/2,x) - L(n-1,1/2,x) \\
= \;& \sum_{m=0}^{n} \frac{,\,(n+3/2)(-1)^m}{,\,(m+3/2)m!(n-m)!}x^m + \sum_{m=0}^{n-1} \frac{,\,(n+1/2)(-1)^m}{,\,(m+3/2)m!(n-1-m)!}x^m \\
= \;& \sum_{m=0}^{n-1} \frac{,\,(n+1/2)(-1)^m}{,\,(m+3/2)m!(n-m)!}((n+1/2)-(n-m))x^m + \frac{(-1)^n}{n!}x^n \\
= \;& \sum_{m=0}^{n} \frac{,\,(n-1/2+1)(-1)^m}{,\,(m-1/2+1)m!(n-m)!}x^m \\
= \;& L(n,-1/2,x). \qquad (153)
\end{aligned}$$

Similarly,

$$(n+1/2)L(n,-1/2,x) - (n+1)L(n+1,-1/2,x) = xL(n,1/2,x). \qquad (154)$$

Applying Eq. 153 and Eq. 154 to $p_i(x)$ in Eq. 2, we have

$$\begin{aligned}
xp_{2n}(x) &= \frac{2^n n!}{\sigma^{2n}} xL(n,-\frac{1}{2},\frac{x^2}{2\sigma^2}) \\
&= \frac{2^n n!}{\sigma^{2n}} \left(xL(n,\frac{1}{2},\frac{x^2}{2\sigma^2}) - xL(n,\frac{1}{2},\frac{x^2}{2\sigma^2}) \right) \\
&= -j(\sigma^2 p_{2n+1}(x) - 2nP_{2n-1}(x)) \qquad (155) \\
xp_{2n+1}(x) &= j\frac{2^n n!}{\sigma^{2n+2}} x^2 L(n,\frac{1}{2},\frac{x^2}{2\sigma^2}) \\
&= j\frac{2^n n!}{\sigma^{2n+2}} 2\sigma^2 \left((n+1/2)L(n,-\frac{1}{2},\frac{x^2}{2\sigma^2}) - (n+1)L(n+1,-\frac{1}{2},\frac{x^2}{2\sigma^2}) \right) \\
&= j((2n+1)p_{2n}(x) - \sigma^2 p_{2n+2}(x)). \qquad (156)
\end{aligned}$$

Actually Eq. 155 and Eq. 156 can be merged into a single recursive formula:

$$xp_i(x) = j(ip_{i-1}(x) - \sigma^2 p_{i+1}(x)). \qquad (157)$$

Replacing Eq. 157 into Eq. 1, we then obtain the recursive property of the moment filters in the spatial domain,

$$xm_i(x) = j(im_{i-1}(x) - \sigma^2 m_{i+1}(x)). \qquad (158)$$

Another recursive property of the moment filters is with respect to differentiation, i.e.

$$\begin{aligned}
m_i'(x) &= j^{-i}\frac{d}{dx}\left(e^{-jf_0 x}\frac{d^i}{dx^i}g(x) \right) \\
&= j(m_{i+1}(x) - f_0 m_i(x)). \qquad (159)
\end{aligned}$$

The following recursive property of the moment filters in the frequency domain is obviously true from Eq. 4,

$$(f-f_0)M_i(f) = M_{i+1}(f). \qquad (160)$$

B.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. 13 becomes a function of the location, i.e.

$$w_i(x_0) = \int_{-\infty}^{+\infty} \xi(x) m_i(x - x_0) dx. \tag{161}$$

We knew 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.

$$\begin{aligned} \frac{d}{dx_0}\phi(x_0) &= \frac{R_0 I_0' - R_0' I_0}{R_0^2 + I_0^2} \\ &= \frac{\mathrm{Im}(w_0^* w_0')}{\| w_0 \|^2}, \end{aligned} \tag{162}$$

where Im means the imaginary part.

From Eq. 161 and Eq. 17, we can compute the spatial derivative of w_0 as,

$$\begin{aligned} w_0' &= -\int_{-\infty}^{+\infty} \xi(x) m_0'(x - x_0) dx \\ &= -j \int_{-\infty}^{+\infty} \xi(x)(m_1(x - x_0) - f_0 m_0(x - x_0)) dx \\ &= j(f_0 w_0 - w_1). \end{aligned} \tag{163}$$

Replacing Eq. 163 into Eq. 162, we then have

$$\frac{d}{dx_0}\phi(x_0) = f_0 - \frac{\mathrm{Re}(w_1 w_0^*)}{\| w_0 \|^2} = f_0 - \mathrm{Re}\left(\frac{w_1}{w_0}\right), \tag{164}$$

where Re means the real part.

The stability criterion of the Gabor filter output is usually formulated as a threshold [6, 30],

$$T = \| \frac{1}{a}\frac{da}{dx_0} \|^2 + \| \frac{d\phi}{dx_0} - f_0 \|^2, \tag{165}$$

where a is the amplitude.

Because

$$\begin{aligned} \frac{1}{a}\frac{da}{dx_0} &= \frac{R_0 R_0' + I_0 I_0'}{R_0^2 + I_0^2} \\ &= \frac{\mathrm{Re}(w_0^* w_0')}{\| w_0 \|^2} \\ &= -\frac{\mathrm{Im}(w_1 w_0^*)}{\| w_0 \|^2} \\ &= -\mathrm{Im}\left(\frac{w_1}{w_0}\right), \end{aligned} \tag{166}$$

we then have

$$T = \frac{\| w_1 \|^2}{\| w_0 \|^2}. \tag{167}$$

C Error Estimations in Moment Filter Approach

C.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:

$$W_i^2 = E_i^2 N_w^2, \tag{168}$$

where E_i^2 is the energy of the filter in Eq. 31.

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. 39 and Eq. 40, and taking the two error sources, i.e. white noise $W()$ and truncation error $T()$, into consideration, we have:

$$u = \frac{2}{f_0^2} \left(\ln(C_0(u) + W(C_0) + T(C_0)) - \ln(V_0 + vP_0 + W(V_0)) \right), \tag{169}$$

$$\frac{V_1 + vP_1 + W(V_1)}{V_0 + vP_0 + W(V_0)} v = \frac{C_1(u) + W(C_1) + T(C_1)}{C_0(u) + W(C_0) + T(C_0)}. \tag{170}$$

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. 168, 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. 39, and the second the estimated energy of truncated part.

From Eq. 169 and Eq. 170, assuming all the error sources are independent, after some manipulations, we have,

$$\begin{bmatrix} du \\ dv \end{bmatrix} = \begin{bmatrix} \frac{f_0^2}{2} - \frac{C_0'(u)}{C_0(u)} & \frac{P_0}{V_0 + vP_0} \\ (V_0 + vP_0)C_1'(u) - (V_1 + vP_1)C_0'(u) & C_1 P_0 - C_0 P_1 \end{bmatrix}^{-1}.$$

$$\left[\begin{array}{cccccc} \frac{1}{C_0(u)} & 0 & -\frac{1}{V_0+vP_0} & 0 & \frac{1}{C_0(u)} & 0 \\ V_1+vP_1 & -(V_0+vP_0) & -C_1(u) & C_0(u) & V_1+vP_1 & -(V_0+vP_0) \end{array}\right] \left[\begin{array}{c} W(C_0) \\ W(C_1) \\ W(V_0) \\ W(V_1) \\ T(C_0) \\ T(C_1) \end{array}\right].$$

Rewrite the above equation in a more concise form, we have

$$\left[\begin{array}{c} du \\ dv \end{array}\right] = \mathbf{AW}, \tag{171}$$

where $\mathbf{A}$ is a 2x6 complex matrix, and $\mathbf{W}$ is the 6x1 error vector.

C.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:

$$W_i^2 = E_i^2 N_w^2, \tag{172}$$

where E_i^2 is the energy of the ith moment filter.

Let

$$\begin{aligned} C_0(D) &= U_0 - jDU_1 - \frac{1}{2}D^2U_2, & (173) \\ C_1(D) &= U_1 - jDU_2 - \frac{1}{2}D^2U_3, & (174) \end{aligned}$$

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,

$$\begin{aligned} \| T(C_0) \|^2 &= \| e^{-jf_0D}(V_0+\mu P_0) - C_0 \|^2 + \frac{E_3^2}{E_0^2} \| \frac{D^3}{6}U_0 \|^2, & (175) \\ \| T(C_1) \|^2 &= \| e^{-jf_0D}(V_1+\mu P_1) - C_1 \|^2 + \frac{E_4^2}{E_0^2} \| \frac{D^3}{6}U_0 \|^2, & (176) \end{aligned}$$

where the first term is the residual in Eq. 73, and the second the estimated energy of 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 truncation error $T()$, as following,

$$\left[\begin{array}{c} dD \\ d\mu \end{array}\right] = \left[\begin{array}{cc} -jf_0 - \frac{C_0'(D)}{C_0(D)} & \frac{P_0}{V_0+\mu P_0} \\ (V_0+\mu P_0)C_1'(D) - (V_1+\mu P_1)C_0'(D) & C_1P_0 - C_0P_1 \end{array}\right]^{-1}.$$

$$\left[\begin{array}{cccccc} \frac{1}{C_0(D)} & 0 & -\frac{1}{V_0+\mu P_0} & 0 & \frac{1}{C_0(D)} & 0 \\ V_1+\mu P_1 & -(V_0+\mu P_0) & -C_1(D) & C_0(D) & V_1+\mu P_1 & -(V_0+\mu P_0) \end{array}\right] \left[\begin{array}{c} W(C_0) \\ W(C_1) \\ W(V_0) \\ W(V_1) \\ T(C_0) \\ T(C_1) \end{array}\right].$$

Rewrite the above equation into a more concise form, we have

$$\left[\begin{array}{c} dD \\ d\mu \end{array}\right] = \mathbf{AW}, \tag{177}$$

where $\mathbf{A}$ is a 2x6 complex matrix, and $\mathbf{W}$ is the 6x1 error vector.

D Properties of Hypergeometric Filters

In this appendix, we will prove some properties of the H filters, 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 properties of the confluent hypergeometric function $\Phi(a,c,x)$, which are useful to us. All the properties listed below are from Chapter 13 in [1]:

$$\Phi(a,c,x) = e^x \Phi(c-a,c,-x), \tag{178}$$

$$(1+a-c)\Phi(a,c,x) - a\Phi(a+1,c,x) + (c-1)\Phi(a,c-1,x) = 0, \tag{179}$$

$$c\Phi(a,c,x) - c\Phi(a-1,c,x) - x\Phi(a,c+1,x) = 0, \tag{180}$$

$$\frac{n!}{(\alpha+1)_n} L(n,\alpha,x) = \Phi(-n,\alpha+1,x). \tag{181}$$

From Eq. 84 and Eq. 86, using the relation between the Laguerre's polynomial and the confluent hypergeometric function in Eq. 181, we have

$$\begin{aligned} h_{2i}(x) + h_{-2i}(x) &= 2a_m \Phi(-i, \frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}} \\ &= 2a_m \frac{i!}{(1/2)_n} L(i, -\frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}}, \end{aligned} \tag{182}$$

$$\begin{aligned} h_{2i+1}(x) - h_{-(2i+1)}(x) &= j2b_m x \Phi(-i, \frac{3}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}} \\ &= j2b_m \frac{i!}{(3/2)_n} x L(i, \frac{1}{2}, \frac{x^2}{2\sigma^2}) e^{-\frac{x^2}{2\sigma^2}}, \end{aligned} \tag{183}$$

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. 77, Eq. 78, and Eq. 79, 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:

$$\begin{aligned} t &= \frac{x}{\sqrt{2}\sigma}, && (184) \\ h_m(t) &= \sqrt{\frac{2^m}{\pi, (m+1/2)\sigma}} e^{-t^2} \\ &\quad \left(\frac{1}{\sqrt{2}}, (\frac{m+1}{2}) \Phi(-\frac{m}{2}, \frac{1}{2}, t^2) + j\sqrt{2}, (1+\frac{m}{2}) \Phi(\frac{1-m}{2}, \frac{3}{2}, t^2) t \right) \\ &= \sqrt{\frac{2^m}{\pi, (m+1/2)\sigma}} \varphi_m(t), && (185) \end{aligned}$$

where

$$\begin{aligned} \varphi_m(t) &= e^{-t^2} \left(\frac{1}{\sqrt{2}}, (\frac{m+1}{2}) \Phi(-\frac{m}{2}, \frac{1}{2}, t^2) + j\sqrt{2}, (1+\frac{m}{2}) \Phi(\frac{1-m}{2}, \frac{3}{2}, t^2) t \right) \\ &= \frac{1}{\sqrt{2}}, (\frac{m+l}{2}) \Phi(\frac{1+m}{2}, \frac{1}{2}, -t^2) + j\sqrt{2}, (1+\frac{m}{2}) \Phi(1+\frac{m}{2}, \frac{3}{2}, -t^2) t. && (186) \end{aligned}$$

Then by using Eq. 179 and Eq. 180 and the property of the Gamma function

$$, (x+1) = x, (x), \qquad (187)$$

we have

$$\begin{aligned} jt\varphi_m(t) &= j\frac{1}{\sqrt{2}}, (\frac{1+m}{2}) t \Phi(\frac{1+m}{2}, \frac{1}{2}, -t^2) - \sqrt{2}, (\frac{2+m}{2}) t^2 \Phi(\frac{2+m}{2}, \frac{3}{2}, -t^2) \\ &= -j\sqrt{2}, (\frac{1+m}{2}) t \left(\frac{m}{2} \Phi(\frac{1+m}{2}, \frac{3}{2}, -t^2) - \frac{1+m}{2} \Phi(\frac{3+m}{2}, \frac{3}{2}, -t^2) \right) \\ &\quad + \frac{1}{\sqrt{2}}, (\frac{2+m}{2}) \left(\Phi(\frac{2+m}{2}, \frac{1}{2}, -t^2) - \Phi(\frac{m}{2}, \frac{1}{2}, -t^2) \right) \\ &= -\left(\frac{1}{\sqrt{2}}, (\frac{2+m}{2}) \Phi(\frac{n}{2}, \frac{1}{2}, -t^2) + j\sqrt{2}, (\frac{1+m}{2}) \frac{m}{2} \Phi(\frac{1+m}{2}, \frac{3}{2}, -t^2) t \right) \\ &\quad + \left(\frac{1}{\sqrt{2}}, (\frac{2+m}{2}) \Phi(\frac{2+m}{2}, \frac{1}{2}, -t^2) + j\sqrt{2}, (\frac{3+m}{2}) \Phi(\frac{3+m}{2}, \frac{3}{2}, -t^2) t \right) \\ &= -\frac{m}{2} \varphi_{m-1}(t) + \varphi_{m+1}(t). && (188) \end{aligned}$$

Replacing Eq. 188 into Eq. 185, we obtain

$$jth_m(t) = \sqrt{\frac{m+1/2}{2}} h_{m+1}(t) - \sqrt{\frac{m^2}{2m-1}} h_{m-1}(t). \qquad (189)$$

Then we replace t in Eq. 189 by Eq. 184. After some manipulations, we have

$$xh_m(x) = -j\sigma \left(\sqrt{m+\frac{1}{2}} h_{m+1}(x) - \sqrt{\frac{m^2}{m-1/2}} h_{m-1}(x) \right). \qquad (190)$$

Similarly

$$\begin{aligned} xh_0(x) &= -\frac{1}{2}j\sigma(h_1(x) - h_{-1}(x)), && (191) \\ xh_{-m}(x) &= j\sigma\left(\sqrt{m+\frac{1}{2}}h_{-(m+1)}(x) - \sqrt{\frac{m^2}{m-1/2}}h_{-(m-1)}(x)\right). && (192) \end{aligned}$$

About the recursive property of H filters about the differential operation, we have,

$$\begin{aligned} h'_m(x) &= \mathcal{F}^{-1}[\mathcal{F}[h'_m(x)]] \\ &= \mathcal{F}^{-1}[jfH_m(f)] \\ &= j\frac{c_m}{c_{m+1}}\mathcal{F}^{-1}[H_{m+1}(f)] \\ &= j\frac{c_m}{c_{m+1}}h_{m+1}(x). && (193) \end{aligned}$$

Similarly

$$\begin{aligned} h'_0(x) &= j\frac{c_0}{c_1}(h_1(x) - h_{-1}(x)), && (194) \\ h'_{-m}(x) &= -j\frac{c_m}{c_{m+1}}h_{-(m+1)}(x). && (195) \end{aligned}$$

References

[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] 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.

[4] V. Michael Bove, Jr. Discrete fourier transform based depth-from-focus. In *Proceedings OSA Topical Meeting on Image Understanding and Machine Vision*, 1989.

[5] Alan C. Bovik, Marianna Clark, and Wilson S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Trans. Patt. Recog. Machine Intell.*, 12(1):55–73, January 1990.

[6] 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.

[7] David J. Fleet and Allan D. Jepson. Computation of normal velocity from local phase information. In *Proc. Comput. Vision Patt. Recog.*, pages 379–386, 1989.

[8] David J. Fleet, Allan D. Jepson, and Michael Jenkin. Phase-based disparity measurement. *CVGIP: Image Understanding*, 53(2):198–210, 1991.

[9] D. Gabor. Theory of communication. *Journal of the IEE*, 93:429–457, 1946.

[10] David J. Heeger. Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, pages 279–302, January 1988.

[11] Anil K. Jain and Farshid Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[12] 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.

[13] 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.

[14] John Krumm and Steven A. Shafer. Shape from periodic texture using the spectrogram. In *Proc. Computer Vision Patt. Recog.*, pages 284–289, 1992.

[15] 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.

[16] 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.

[17] Jitendra Malik and Ruth Rosenholtz. A differential method for computing local shape-from-texture for planar and curved surfaces. In *Proc. Computer Vision Patt. Recog.*, pages 267–273, 1993.

[18] 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.

[19] 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.

[20] Arnold F. Nikiforov and Vasilii B. Uvarov. *Special Functions of Mathematical Physics*. Birkhauser Verlag Basel, 1988.

[21] Alex P. Pentland. A new sense for depth of field. *IEEE Transactions on PAMI*, 9(4):523–531, 1987.

[22] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1988.

[23] T. D. Sanger. Stereo disparity computation using Gabor filters. *Biological Cybernetics*, 59:405–418, 1988.

[24] Randall C. Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainty. *The International Journal of Robotics Research*, 5(4), 1986.

[25] Jerome Spanier and Keith B. Oldham. *An Atlas of Functions.* Washington: Hemisphere Pub. Corp., 1987.

[26] Murali Subbarao. Parallel depth recovery by changing camera parameters. In *2nd International Conference on Computer Vision*, pages 149–155, 1988.

[27] Juyang Weng. A theory of image matching. In *International Conference on Computer Vision*, pages 200–209, 1990.

[28] 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.

[29] Yalin Xiong and Steven A. Shafer. Depth from focusing and defocusing. In *Proc. Computer Vision Patt. Recog.*, pages 68–73, 1993.

[30] Yalin Xiong and Steven A. Shafer. Variable window Gabor filters and their use in focus and correspondence. In *Proc. Computer Vision and Pattern Recognition*, 1994.