

Moment Filters For High Precision Computation of Focus and Stereo

Yalin Xiong

Steven A. Shafer

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Many low level visual computation problems such as focus, stereo, 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 a new set of filters, moment filters. Due to their recursiveness in the Fourier and spatial domains, these filters allow the effects of finite-width windows and foreshortening to be explicitly analyzed and eliminated.

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 a new sets of filters, “moment” filters. Due to their recursiveness in the Fourier domain, the effects of finite-width windows can be explicitly analyzed and eliminated.

The most fundamental difference between finite- and infinite-width filters in the Fourier domain is that the bandwidth of finite-width filters must be non-zero while that of infinite-width filters is zero. Since a non-stationary transform in the Fourier domain can be an arbitrary curve within the passband, 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 approximate the curve by a polynomial, whose order can be arbitrarily high so that the approximation can be as accurate as a specific task requires. Therefore, the extraction of the parameter based upon the polynomial approximation is much more accurate.

We apply the technique of moment filters to the problems of focus and stereo. Traditionally, people estimate the parameter by comparing the outputs of convolutions of the two images with the same filter, usually a Gabor filter. Examples in focus are [5, 7, 10],

in stereo are [6, 8, 2]. Our moment filter approach provides a new insight into this traditional approach by showing that the output of the convolution of one image with a 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 a parameter using finite-width filters, it is usually assume that the parameter is constant within the width of filters. But another aspect of the non-stationarity is that the parameter actually changes within the effective width of the filters, which we refer to as the shift variance problem or foreshortening problem. The recursiveness of the moment filters in the spatial domain allows modeling this effect as a modification of the Gabor filter convolution by convolutions of the higher order moment filters. Applying this technique to the focus and stereo problem, we demonstrate that we can not only eliminate the negative effects on the estimation of the parameter caused by the foreshortening, but also estimate the the degree of foreshortening quantitatively.

2 Theory of Moment Filter

We define the i th order moment filter in the spatial domain as: ($i = 0, 1, 2, \dots$)

$$m_i(x) = p_i(x) \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} e^{-j f_0 x}, \quad (1)$$

in which

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

where n is an integer, j is $\sqrt{-1}$, and $L(n, \alpha, x)$ is the generalized Laguerre polynomial [4]. Figure 1 illustrates some of the moment filters in the spatial domain when $f_0 = 0$.

Apparently the zeroth order moment filter is the Gabor filter $G(x; f_0)$ with the peak frequency f_0 ,

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

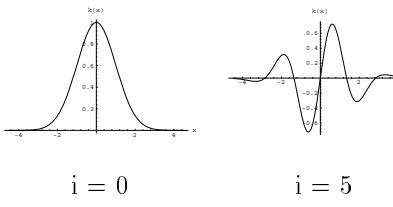


Figure 1: The Moment Filters in Spatial Domain

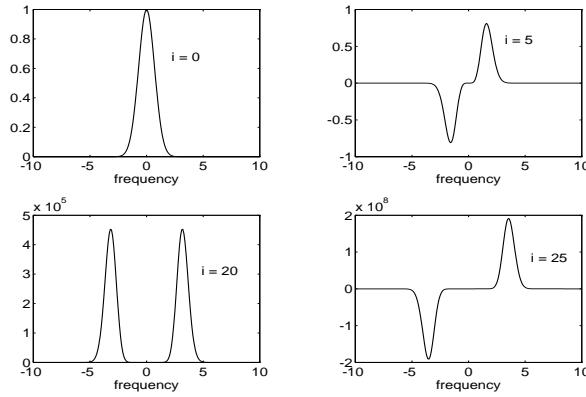


Figure 2: The Pass Band Moves as i Increases

In the Fourier domain, the moment filter is:

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

Figure 2 illustrates the pass bands of moment filters as i increases, assuming $f_0 = 0$.

Here are some important properties of moment filters given without proof. Interested readers may refer to [9].

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

$$(f - f_0) M_i(f) = M_{i+1}(f). \quad (5)$$

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

$$x m_i(x) = j(i m_{i-1}(x) - \sigma^2 m_{i+1}(x)). \quad (6)$$

3 Moment Filters For Focus

The depth from defocus method obtains 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 the approach is applicable to any model as we will see later. Let us denote two images as $i_0(x)$ and $i_1(x)$ in the spatial domain and $I_0(f)$ and $I_1(f)$ in the Fourier domain. The relations between these two images are:

$$i_1(x) = i_0(x) \otimes \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{x^2}{2\sigma_0^2}} \quad (7)$$

$$I_1(f) = I_0(f) e^{-f^2 \sigma_0^2/2} \quad (8)$$

The difference of focus between two images is characterized by σ_0 . If σ_0 is not a constant within the window, we can assume that it changes linearly in neighborhood of the window center $x = 0$, i.e.

$$\sigma_0 = s(1 + \mu x). \quad (9)$$

We can expand the second integral term in Eq. 7 as

$$\begin{aligned} & \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x-t)^2}{2\sigma_0^2}} \\ & \approx \frac{1}{\sqrt{2\pi}s} e^{-\frac{(x-t)^2}{2s^2}} + \mu x \frac{(x-t)^2 - s^2}{\sqrt{2\pi}s^3} e^{-\frac{(x-t)^2}{2s^2}} \end{aligned} \quad (10)$$

where we truncated terms of higher order of μx assuming μ is small.

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

$$U_i = \int i_0(x) m_i(-x) dx = \int I_0(f) M_i(f) df, \quad (11)$$

$$\begin{aligned} V_i &= \int i_1(x) m_i(-x) dx = \int I_1(f) M_i(f) df \\ &= \int I_0(f) e^{-f^2 \sigma_0^2/2} M_i(f) df, \end{aligned} \quad (12)$$

where the integrations are done from $-\infty$ to $+\infty$.

By Taylor expansion in the neighborhood of f_0 , we have

$$\begin{aligned} & e^{-f^2 \sigma_0^2/2} / e^{-f_0^2 \sigma_0^2/2} \\ & \approx \left(1 - f_0 \sigma_0^2 (f - f_0) - \frac{\sigma_0^2 - f_0^2 \sigma_0^4}{2} (f - f_0)^2 \right) \end{aligned} \quad (13)$$

where higher order expansions are truncated.

Replacing Eq. 13 into Eq. 12 and using the recursiveness of the moment filters in the Fourier domain, we obtain

$$V_0 = e^{-f_0^2 \sigma_0^2/2} (U_0 - f_0 \sigma_0^2 U_1 - \frac{\sigma_0^2 - f_0^2 \sigma_0^4}{2} U_2). \quad (14)$$

Therefore, we can compute σ_0^2 from this equation

$$\sigma_0^2 = \frac{2}{f_0^2} \left(\ln(U_0 - f_0 \sigma_0^2 U_1 - \frac{\sigma_0^2 - f_0^2 \sigma_0^4}{2} U_2) - \ln V_0 \right). \quad (15)$$

Comparing with the equation from Eq. 8, i.e.

$$\sigma_0^2 = \frac{2}{f_0^2} (\ln I_0(f_0) - \ln I_1(f_0)), \quad (16)$$

we can see that if we assume that the effects caused by window are negligible, this may result into substantial errors if either U_1 or U_2 happens to be not very small in comparison to U_0 . More details on solving Eq. 15 can be found in [9].

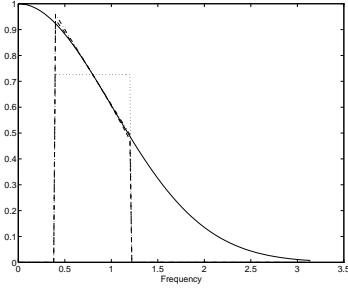


Figure 3: Polynomial Fitting Of The Transformation Curve

A more intuitive way of looking at the difference between Eq. 15 and Eq. 16 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 is 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. 16 is equivalent to using a constant to approximate the transformation curve within the passband. The first order approximation (dash-dot curve), i.e. Eq. 15 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. 15, is equivalent to using a quadratic curve to approximate the transformation curve within the passband. Generally, if we use up to the n th order moment filter in the Fourier domain, we are actually fitting the transformation curve by an n th order polynomial. Certainly, it will drastically decrease errors compared to the constant approximation.

When $\mu \neq 0$ in Eq. 9 and Eq. 10, we must consider the effect of the shift variance. We define

$$V_i = \int i_1(x) m_i(-x) dx, \quad (17)$$

$$\mathcal{V}_i = \int \left(\int i_0(t) \frac{1}{\sqrt{2\pi}s} e^{-\frac{(x-t)^2}{2s^2}} dt \right) m_i(-x) dx \quad (18)$$

In other words, \mathcal{V}_i represents the convolution of frontalized image, while V_i represents the convolution of the original real image $i_1(x)$. In order to use Eq. 15 to compute s accurately, we need to compute \mathcal{V}_i from V_i .

Replacing Eq. 10 into Eq. 17, and using the recursiveness of the moment filters in the spatial domain, we can represent the effects of the shift variance as

$$\begin{aligned} \mathcal{V}_i &\approx V_i - js^2\mu(\sigma^2(V_{i+3} + 2f_0V_{i+2} + f_0^2V_{i+1}) - \\ &\quad i(V_{i+1} + 2f_0V_i + f_0^2V_{i-1})) \\ &= V_i + s^2\mu P_i, \end{aligned} \quad (19)$$

i.e. when μ is small, we can represent the effects of the shift variance as a linear recombination of moment filter outputs.

Therefore, we can compute s and μ simultaneously by solving the following equations:

$$s^2 = \frac{2}{f_0^2} (\ln(U_0 + f_0s^2U_1 - \frac{1}{2}(s^2 - f_0^2s^4)U_2)$$

$$- \ln(V_0 + s^2\mu P_0)), \quad (20)$$

$$\frac{V_1 + s^2\mu P_1}{V_0 + s^2\mu P_0} \approx \frac{U_1 - f_0s^2U_2 - \frac{1}{2}s^2(1 - f_0^2s^2)U_3}{U_0 - f_0s^2U_1 - \frac{1}{2}s^2(1 - f_0^2s^2)U_2}. \quad (21)$$

More details of solving the equations can be found in [9].

4 Moment Filters For Stereo

The depth from stereo method obtains 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

$$i_1(x) = i_0(x + D_0), \quad (22)$$

$$I_1(f) = I_0(f) e^{j f D_0}. \quad (23)$$

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

$$D_0 = D + \mu x, \quad (24)$$

we then have

$$i_0(x + D_0) \approx i_0(x + D) + \mu x \frac{d}{dx} i_0(x + D), \quad (25)$$

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

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

$$e^{j f D_0} \approx e^{j f_0 D_0} (1 + j D_0(f - f_0) - \frac{D_0^2}{2}(f - f_0)^2), \quad (26)$$

where higher order terms are truncated.

Using exactly the same method as we did in the previous section, we have an equation to compute D_0 as

$$D_0 = \frac{j}{f_0} \left(\ln(U_0 + j D_0 U_1 - \frac{D_0^2}{2} U_2) - \ln V_0 \right). \quad (27)$$

Again, as we showed in the focus problem, the difference between Eq. 27 and Eq. 23 is that the moment filter approach uses a second order polynomial to approximate the transformation curve $e^{j f D_0}$ in the neighborhood of f_0 , while the traditional approach uses a constant $e^{j f_0 D_0}$ to approximate the curve. As we can see, only when $U_1 = U_2 = 0$, is Eq. 27 the same as Eq. 23. Therefore, when the magnitude of U_1 or U_2 is not small, using Eq. 23 to compute D_0 will result into large errors.

It is not a surprise to learn that the stability constraint used by Fleet et al [2] is actually the magnitude ratio of U_1 and U_0 , i.e.

$$\left\| \frac{\partial \phi}{\partial x} + j \frac{1}{a} \frac{\partial a}{\partial x} \right\| = \frac{\| U_1 \|}{\| U_0 \|}, \quad (28)$$

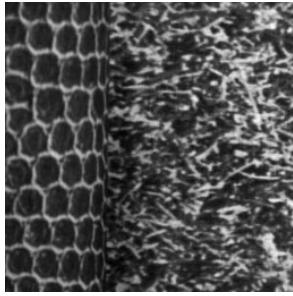


Figure 4: Synthetic Texture Image For Focus

where ϕ is the phase, and a is the magnitude. Eq. 28 also justifies the so-called phase singularity, which is a location where U_1 is so large compared with U_0 that Eq. 23 isn't even approximately valid. We can see that such a situation is exactly characterized by Eq. 27.

When $\mu \neq 0$ in Eq. 24 and Eq. 25, we must consider the effects of the shift variance. Using the same method as we did in the previous section, we can represent the effects of shift variance as

$$\begin{aligned} V_i &\approx V_i + \mu (i(f_0 V_{i-1} - V_i) - \sigma^2 (f_0 V_{i+1} - V_{i+2})) \\ &= V_i + \mu Q_i, \end{aligned} \quad (29)$$

where Q_i is used for simplicity of illustration.

Combining Eq. 27 and Eq. 29, we have the equations to solve D and μ

$$D = \frac{j}{f_0} \left(\ln(U_0 + jDU_1 - \frac{D^2}{2}U_2) - \ln(V_0 + \mu Q_0) \right), \quad (30)$$

$$\frac{V_1 + \mu Q_1}{V_0 + \mu Q_0} \approx \frac{U_1 + jDU_2 - \frac{1}{2}D^2 U_3}{U_0 + jDU_1 - \frac{1}{2}D^2 U_2}. \quad (31)$$

More details on numerically solving those equations can be found in [9].

5 Experiments

5.1 Focus

We implemented both algorithms under the variable window scheme. More details of implementation can be found in [9]. The performance tests were done on both synthetic and real images. In experiments on synthetic images, the original image is shown in Figure 4. We uniformly blurred the original texture image by a Gaussian function with parameter $\sigma_0 = 1.0$ (in pixel-width) to get the first synthetic image pair. And we non-uniformly blurred the original image by a Gaussian function with parameter σ_0 increasing linearly as the column increases, to get the second synthetic image pair.

We compare three different algorithms, namely, Subbarao's algorithm (SA), the moment filter algorithm without slope estimation (MFF1), and the moment filter algorithm with slope estimation (MFF2). Table 1 shows the root mean square errors of the synthetic image pairs using different algorithms. For the first pair, MFF1 and MFF2 show almost no error at

	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

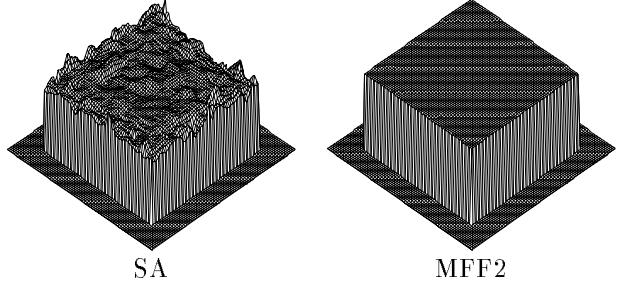


Figure 5: Computed Blurring Difference of The First Pair

all, while SA has a significant amount of error. For the second pair, the RMS error of SA is 4 times larger than that of MFF1, and 27 times larger than that of MFF2! We can see that the main source of error in Subbarao's algorithm arises from the truncation of information from moment filters. When the surface is slanted, i.e. foreshortening happens, the error increases as the surface slant increases. The MFF2 algorithm can correct the errors caused by the foreshortening. Figure 5 and Figure 6 show the results as elevation maps. And Figure 7 shows the slope computed for the second pair by MFF2.

A real image pair is shown in Figure 8. These two pictures were taken under different aperture and exposure time settings. The toy house is about two meters away from the camera, and the size of the house itself is about 2.5 inches wide, 3.0 inches long, and 4.5 inches high. Figure 9 shows the blurring difference of the house image pair computed using Subbarao's algorithm and MFF1. We can see that the result from SA is so noisy that the shape of the house can not be observed with confidence. But the shape of the house from MFF1 is clearly visible though the discontinuity generates a lot of spikes. Finally Figure 10 shows the result using MFF2. We can see that the computed blurring difference is even more smooth. Additionally,

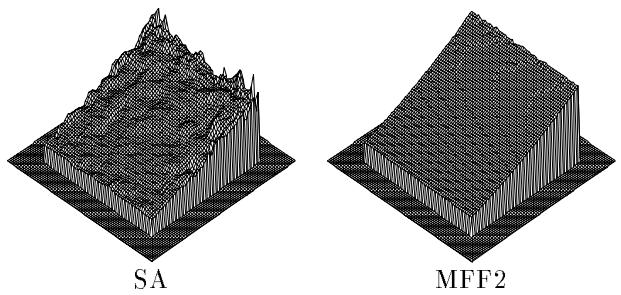


Figure 6: Computed Blurring Difference of The Second Pair

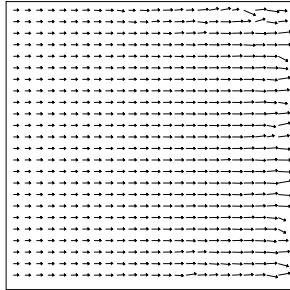


Figure 7: Computed Surface Slope of The Second Pair



Figure 8: Real Image Pair of A Toy House

most of the slope vectors have the correct direction despite the fact that the slope is very small ($\mu \approx 0.002$ in Eq. 9), and the discontinuity generates very large slope vectors which can be used potentially to locate the discontinuity in later processing stages.

5.2 Stereo

Though filter-based stereo algorithms may be able to alleviate the mismatch problem, our concern here is focused on the precision of sub-pixel registration. We will compare our algorithms with Kanade & Okutomi's adaptive window scheme [3]. 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 our algorithms under the fixed window scheme. More details of the implementation can be found in [9]. For Kanade & Okutomi's algorithm, the range of the window size is from 3 to 21,

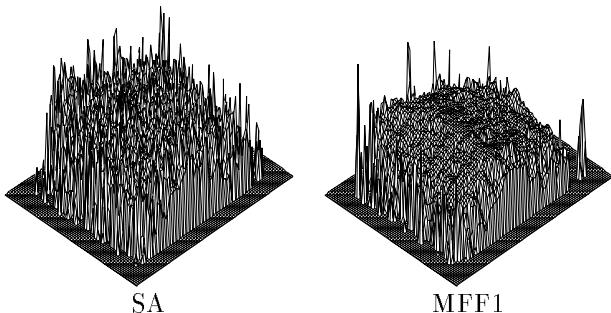


Figure 9: Blurring Difference of House From SA and MFF1

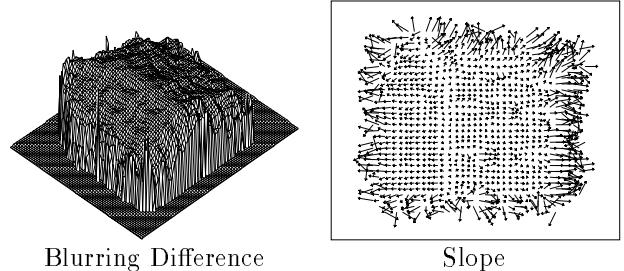


Figure 10: Blurring Difference and Slope of House From MFF2



Figure 11: The 20th Frame in Tree Sequence

and the iteration count 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 approach with slope correction as MFS2. The RMS errors we use below are in pixel widths.

We first tested the stereo algorithms on synthetic images. The synthetic image pair is the 20th and 21st frames of the translating tree sequence [1] as in Figure 11(b). 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. Figure 12 and Figure 13 show the disparity maps and slope computed by different algorithms. The RMS errors of KOA, MFS1 and MFS2 are 0.016, 0.012 and 0.010 pixel-width respectively.

We also tested these algorithms on a real image pair.

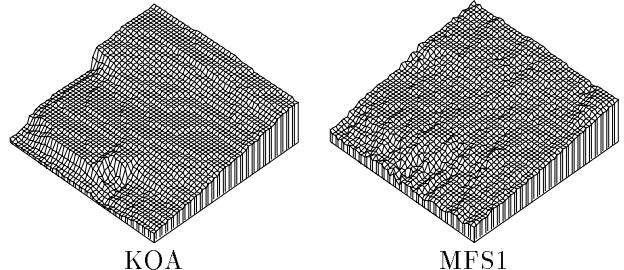


Figure 12: Disparity Maps of the Tree Pair from KOA and MFS1

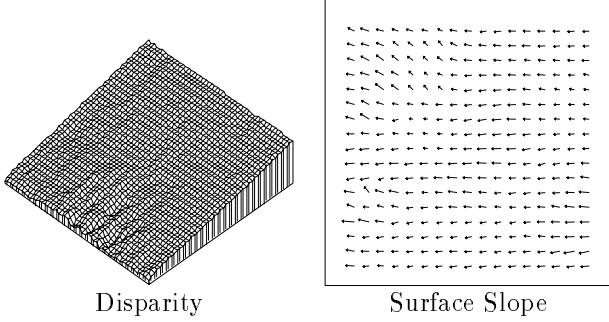


Figure 13: Disparity and Slope of the Tree Pair from MFS2

The targets were about 2.5 meters from the camera, and the camera was translated horizontally by 0.5 inch between the left and right views. Because there is no vergence, the disparity map of a planar surface should still be planar. We will fit planes 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 pixels near the boundaries so that the RMS errors we computed are not influenced by the results near depth discontinuities.

The pair of images are of a textured cube shown in Figure 14. There are three large planar surfaces, which we denote as top patch, left patch, and right patch in Figure 14. The difference between the largest and the smallest disparity is about 3.0 pixel widths. Table 2 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 remains almost constant when the surface slant increases which we attribute to the slope correction performed by MFS2, while the error of KOA increases obviously as the slope increases. Figure 15 shows the disparity and slope computed by MFS2. The slope estimation correctly indicates not only the direction of the slope, but also the magnitude of the slant.

	Top Patch	Left Patch	Right Patch
KOA	0.063	0.032	0.027
MFS1	0.039	0.024	0.020
MFS2	0.018	0.020	0.016

Table 2: RMS Errors of The Cube

Acknowledgment

The first author is in the Robotics Ph.D. Program in Carnegie Mellon University. This research was sponsored by the Department of the Army, Army Research Office under grant number DAAH04-94-G-0006.

References

- [1] 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.
- [2] David J. Fleet, Allan D. Jepson, and Michael Jenkin. Phase-based disparity measurement. *CVGIP: Image Understanding*, 53(2):198–210, 1991.
- [3] 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.
- [4] Arnold F. Nikiforov and Vasilii B. Uvarov. *Special Functions of Mathematical Physics*. Birkhauser Verlag Basel, 1988.
- [5] Alex P. Pentland. A new sense for depth of field. *IEEE Transactions on PAMI*, 9(4):523–531, 1987.
- [6] T. D. Sanger. Stereo disparity computation using Gabor filters. *Biological Cybernetics*, 59:405–418, 1988.
- [7] Murali Subbarao. Parallel depth recovery by changing camera parameters. In *2nd International Conference on Computer Vision*, pages 149–155, 1988.
- [8] Juyang Weng. Image matching using the windowed Fourier phase. *International Journal of Computer Vision*, 11(3):211–236, December 1993.
- [9] Yalin Xiong and Steven A. Shafer. Moment and hypergeometric filters for high precision computation of focus, stereo, and optical flow. Technical Report CMU-RI-TR-94-28, The Robotics Institute, Carnegie Mellon University, 1994.
- [10] Yalin Xiong and Steven A. Shafer. Variable window Gabor filters and their use in focus and correspondence. In *Proc. Computer Vision and Pattern Recognition*, pages 668–671, 1994.

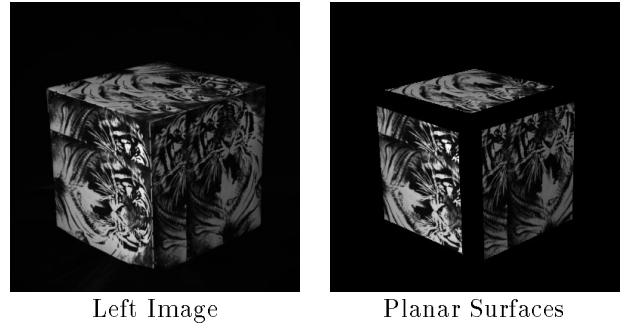


Figure 14: Stereo Images of A Textured Cube

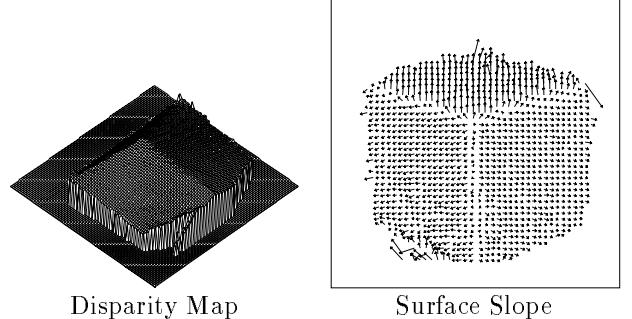


Figure 15: Disparity and Slope of The Cube From MFS2