Rational Filters for Passive Depth from Defocus*

Masahiro Watanabe[†] **and Shree K. Nayar**[‡]

[†]**Production Engineering Research Lab., Hitachi Ltd.,**
292 Yoshida-cho, Totsuka, Yokohama 244, Japan
E-mail: nabe@cs.columbia.edu

[‡]**Department of Computer Science, Columbia University,**
New York, NY 10027
E-mail: nayar@cs.columbia.edu

**Address for Correspondence:* Shree K. Nayar, Department of Computer Science, Columbia University, New York, NY 10027.

Abstract

A fundamental problem in depth from defocus is the measurement of relative defocus between images. The performance of previously proposed focus operators are inevitably sensitive to the frequency spectra of local scene textures. As a result, focus operators such as the Laplacian of Gaussian result in poor depth estimates. An alternative is to use large filter banks that densely sample the frequency space. Though this approach can result in better depth accuracy, it sacrifices the computational efficiency that depth from defocus offers over stereo and structure from motion. We propose a class of broadband operators that, when used together, provide invariance to scene texture and produce accurate and dense depth maps. Since the operators are broadband, a small number of them are sufficient for depth estimation of scenes with complex textural properties. In addition, a depth confidence measure is derived that can be computed from the outputs of the operators. This confidence measure permits further refinement of computed depth maps. Experiments are conducted on both synthetic and real scenes to evaluate the performance of the proposed operators. The depth detection gain error is less than 1%, irrespective of texture frequency. Depth accuracy is found to be 0.5 $\sim$ 1.2% of the distance of the object from the imaging optics.

Index Terms: Passive depth from defocus, blur function, scene textures, normalized image ratio, broadband rational operators, texture invariance, depth confidence measure, depth estimation, real-time performance.

1 Introduction

A pertinent problem in computational vision is the recovery of three-dimensional scene structure from two-dimensional images. Of all problems studied in vision, the above has, by far, attracted the most attention. This has resulted in a variety of sensors and algorithms [Jarvis-1983, Besl-1988] that can be broadly classified into two categories: active and passive. Active techniques produce relatively reliable depth maps, and have been applied to many industrial applications. However, when the environment cannot be controlled, as in the case of distant objects in outdoor scenes, active methods prove impractical. As a consequence, passive techniques are always desirable.

Passive sensing methods, such as stereo and structure from motion, rely on algorithms that establish local correspondences between two or more images. From the resulting disparity estimates or motion vectors, the depths of points in the scene are computed. The process of determining correspondence is widely acknowledged as being computationally expensive. In addition, the above techniques suffer from the occlusion or missing part problems; it is not possible to compute depths of scene points that are visible in only one of the images. Alternative passive techniques are based on focus analysis. Depth from focus uses a sequence of images taken by changing the focus setting of the imaging optics in small steps. For each pixel, the focus setting that maximizes image contrast is determined. This, in turn, can be used to compute the depth of the corresponding scene point [Horn-1968, Jarvis-1983, Krotkov-1987, Darrell and Wohn-1988, Nayar and Nakagawa-1994].

In contrast, depth from defocus uses only two images with different optical settings [Pentland-1987, Subbarao-1988, Ens and Lawrence-1991, Bove, Jr.-1993, Subbarao and Surya-1994, Nayar *et al.*-1995, Xiong and Shafer-1995]. The relative defocus in the two images can, in principle, be used to determine three-dimensional structure. The focus level in the two images can be varied by changing the focus setting of the lens, by moving the image sensor with respect to the lens, or by changing the aperture size. Depth from defocus is not confronted with the abovementioned missing part and correspondence problems. This makes it an attractive prospect for structure estimation.

Despite these merits, at this point in time, fast, accurate, and dense depth from defocus has only been demonstrated using active illumination that constrains the dominant frequencies of the scene texture [Nayar *et al.*-1995, Watanabe *et al.*-1995]. Past investigations of *passive* depth from defocus indicate that it can prove computationally expensive to obtain a reliable depth map. This is because the frequency characteristics of scene textures are, to a large extent, unpredictable. Furthermore, the texture itself can vary dramatically over the image. Since the response of the defocus (blur) function varies with texture frequency, a single broadband filter that produces an aggregate estimate of defocus for an unknown texture cannot lead to accurate depth estimates. The obvious solution is to use an enormous bank of narrow-band filters and compute depth in a least-squares sense using all dominant frequencies of the texture [Xiong and Shafer-1995, Gokstorp-1994]. This requires one to forego computational efficiency. To worsen matters, a depth map of high spatial resolution can be obtained only if all the filters in the bank have small kernel sizes. The *uncertainty relation* [Bracewell-1965] tells us that the frequency resolution of the filter bank reduces proportional to the inverse of the kernel size used. In short, one cannot design a filter with narrow enough response if the support area of the filter kernel is small.

Xiong and Shafer [Xiong and Shafer-1995] proposed an attractive way to cope with this problem. They used *moment filters* to compensate for the frequency spectrum of the texture within the passband of each of the narrowband filters. This approach results in accurate depth estimates but requires the use of four additional filters for each of the tuned filters in the filter bank. This translates to five times as many convolutions as is needed for any typical filter bank method. Xiong and Shafer [Xiong and Shafer-1995] use 240 convolutions in total, which makes their approach computationally expensive.

Ens and Lawrence [Ens and Lawrence-1991] have proposed a method based on a spatial-domain analysis of two blurred images. They estimate the convolution matrix, which is convolved with one of the two images to produce the other image. The matrix corresponds to the relative blur between the two images. Once the matrix is computed, it can be mapped to depth estimates. This method produces accurate depth maps. However, the iterative nature of the convolution matrix estimation makes it computationally expensive.

Subbarao and Surya [Subbarao and Surya-1994] proposed the *S-Transform* and applied it to depth from defocus. They modeled the image as a third-order polynomial in spatial domain, and arrived at a simple and elegant expression [Subbarao and Surya-1994]:

$$i_2(x,y) - i_1(x,y) \;=\; \frac{1}{4}\,({\sigma_2}^2 - {\sigma_1}^2)\,\nabla^2\,(\frac{i_2(x,y)+i_1(x,y)}{2})\,, \tag{1}$$

where, i_1 and i_2 are the far and near focused images, respectively. The blur circle diameters in images i_1 and i_2 are expressed by their second central moments ${\sigma_2}^2$ and ${\sigma_1}^2$, respectively. Since an additional relation between σ_2 and σ_1 can be obtained from the focus settings used for the two images, σ_2 and σ_1 can be solved for and mapped to a depth estimate. As we see no terms that depend on scene frequency in equation (1), this can be considered to be a sort of texture-frequency invariant depth from defocus method. It produces reasonable depth estimates for large planar surfaces in the scene. However, it does not yield depth maps with high spatial resolution that are needed when depth variations in the scene are significant. We argue that this requires a more detailed analysis of image formation as well as the design of novel filters based on frequency analysis.

In this paper, we propose a small set of filters, or operators, for passive depth from defocus. These operators, when used in conjunction, yield invariance to texture frequency while computing depth. The underlying idea is to precisely model relative image blur in frequency domain and express this model as a rational function of two linear combinations of basis functions. This rational expression leads us to a texture-invariant set of operators. The outputs of the operators are used as coefficients in a depth recovery equation that is solved to get a depth estimate. The attractive feature of this approach is that it uses only a small number of broadband linear operators with small kernel supports. Consequently, depth maps are computed not only with high efficiency and accuracy but also with high spatial resolution. Since our operators are derived using a rational expression to model relative image blur, they are referred to as *rational operators*. Rational operators are general, in that, they can be derived for any blur model.

The paper is structured as follows. First, the concept of a texture invariant operator set is described. Next, all the operations needed for depth from defocus are discussed, including the use of prefiltering and coefficient smoothing. An efficient algorithm for obtaining a confidence measure from the operator outputs is outlined. These confidence measures are effectively used for further refinement of computed depth maps. In our specific implementation of rational operators, we have used three basis functions to model the relative blurring function. This has resulted in a set of three rational operators with kernel sizes of 7×7. This operator set has been used to compute depth maps for both synthetic scenes and real scenes. The experimental results are analyzed to quantify the performance of the proposed depth from defocus approach.

2 Depth From Defocus

2.1 Principle

Fundamental to depth from defocus is the relationship between focused and defocused images[Born and Wolf-1965]. Figure 1 shows the basic image formation geometry. All light rays that are radiated by object point P and pass the aperture A are refracted by the lens to converge at point Q on the image plane. The relationship between the object distance d, focal length of the lens f, and the image distance d_i is given by the lens law:

$$\frac{1}{d} + \frac{1}{d_i} = \frac{1}{f}. \tag{2}$$

Each point on the object plane is projected onto a single point on the image plane, causing a clear or *focused* image i_f to be formed. If, however, the sensor plane does not coincide with the image plane and is displaced from it, the energy received from P by the lens is distributed over a patch on the sensor plane. The result is a blurred image of P.

It is clear that a single image does not include sufficient information for depth estimation, as two different scenes defocused to different degrees could produce identical images. A solution to the depth estimation problem is achieved by using two images, i_1 and i_2, separated by a known physical distance $2e$ [Ens and Lawrence-1991, Subbarao and Surya-1994]. The distance γ of the image i_1 from the lens should also be known. Given the above described setting, the problem is reduced to analyzing the relative blurring of each scene point in the two images and computing the position of its focused image. A restriction here is that the images of all of the scene points must lie between the *far-focused* sensor plane i_1 and the *near-focused* sensor plane i_2. For ease of description, we introduce the *normalized depth* α, which equals -1 at i_1 and 1 at i_2. Then, using $d_i = \gamma + (1+\alpha)e$ in the lens law (2), we obtain the depth d of the scene point.

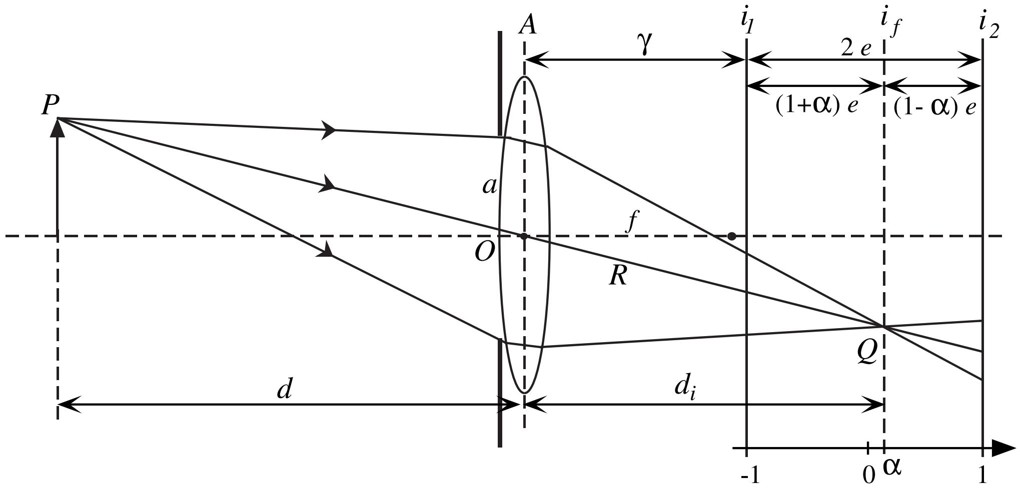


Figure 1: Image formation and depth from defocus. The two images, i_1 and i_2, include all the information required to recover scene structure between the focused planes in the scene corresponding to the two images.

2.2 Defocus Function

Precise modeling of the defocus function is critical to accurate depth estimation. The defocus function is described in detail in previous works [Born and Wolf-1965, Horn-1986]. In Figure 1, $(1 \pm \alpha)e$ is the distance between the focused image of a scene point and its defocused image formed on the sensor plane. The light energy radiated by the scene point and collected by the imaging optics is uniformly distributed on the sensor plane over a circular patch with a radius of $(1 \pm \alpha)e\, a/d_i$[1]. This distribution, also called the *pillbox*, is the defocus function:

$$h(x, y) = h(x, y; (1 \pm \alpha)e, F_e) = \frac{4{F_e}^2}{\pi(1 \pm \alpha)^2 e^2} \Pi(\frac{F_e}{(1 \pm \alpha)e}\sqrt{x^2 + y^2}) \tag{3}$$

where, $+$ is used for image i_1, $-$ is used for image i_2, and $\Pi(r)$ is the rectangular function which takes the value 1 for $|r| < \frac{1}{2}$ and 0 otherwise. F_e is the effective *F-number* of the optics. In the optical system shown in Figure 1, F_e equals $d_i/2a$. In order to eliminate magnification differences between the near and far focused images, we have used *telecentric optics*, which is described in Appendix 7.1.1 and detailed in [Watanabe and Nayar-1995b]. In the telecentric case, F_e equals $f/2a'$.

In Fourier domain, the defocus function in (3) is:

$$H(u, v) = H(u, v; (1 \pm \alpha)e, F_e) = \frac{2F_e}{\pi(1 \pm \alpha)e\sqrt{u^2 + v^2}} J_1(\frac{\pi(1 \pm \alpha)e}{F_e}\sqrt{u^2 + v^2}) \tag{4}$$

where, J_1 is the first-order Bessel function of the first kind, and u and v denote spatial frequency parameters in the x and y directions, respectively[2]. As is evident from the above expression, defocus serves as a low-pass filter. The bandwidth of the filter decreases as the radius of the blur circle increases, i.e. as the plane of focus gets farther from the sensor plane. Figure 2 illustrates this effect. Figure 2(a) shows the image $i_f(x, y)$ formed at the focused plane and its Fourier spectrum $I_f(u, v)$. When the sensor plane is displaced by a distance $(1 - \alpha)e$, the defocused image i_2 is the convolution of the focused image $i_f(x, y)$ with the pillbox $h_2(x, y)$, as shown in Figure 2(b). The effect of defocus in spatial

[1] This geometric model is valid as far as the image is not exactly focused, in which case, a wave optics model is needed to describe the point spread function. Further, it is assumed that lens induced aberrations are small compared to the radius of the blur circle [Born and Wolf-1965].

[2] In the past, most investigators have used the Gaussian model instead of the pillbox model for the blur function. This is mainly to facilitate mathematical manipulations; the Fourier transform of a Gaussian function is also a Gaussian which can be converted into a quadratic function by using the logarithm. As we will see, in our approach to depth from defocus, any form of blur function can be used.

and frequency domains can be written as:

$$\begin{aligned} i_2(x,y) &= i_f(x,y) * h(x,y;(1-\alpha)e, F_e)\,, \\ I_2(u,v) &= I_f(u,v) \cdot H(u,v;(1-\alpha)e, F_e)\,. \end{aligned} \tag{5}$$

Since α can vary from point to point in the image, strictly speaking, we have a *space-variant* system that cannot be expressed as a convolution. Therefore, equation (5) does not hold in a rigorous sense. However, if we assume that α is constant in a small patch around each pixel, equation (5) remains valid within the small patch. Hereon, when we use the terms Fourier transform or spectrum, they are assumed to be those of a small image patch. For the assumption that α variation in a patch is small to be valid, the patch itself must be small. In practice, to realize this requirement, one is forced to use broadband filters; the kernel size of a linear filter is inversely proportional to the bandwidth of the filter.

Figure 2(c) is similar to (b) except that the sensor lies at the distance $(1+\alpha)e$ from the focused plane to produce the defocused image i_1. Again:

$$\begin{aligned} i_1(x,y) &= i_f(x,y) * h(x,y;(1+\alpha)e, F_e)\,, \\ I_1(u,v) &= I_f(u,v) \cdot H(u,v;(1+\alpha)e, F_e)\,. \end{aligned} \tag{6}$$

Note that in the spectrum plots we have used the polar coordinates (f_r, f_θ) for spatial frequencies, rather than Cartesian coordinates (u, v). This is because the defocus function is usually rotationally symmetric. This symmetry allows us to express the defocus spectrum using a single parameter, namely, the radial frequency $f_r = \sqrt{u^2+v^2}$. We see in Figure 2 that, since the image in (c) is defocused more than the one in (b), the low-pass response of $H_1(u,v)$ is greater than that of $H_2(u,v)$.

2.3 Depth from Two Images

We now introduce the *normalized ratio*, $\frac{M}{P}(u,v;\alpha)$, where, $M(u,v) = I_2(u,v) - I_1(u,v)$ and $P(u,v) = I_2(u,v) + I_1(u,v)$. Equivalently, in the spatial domain, we have $m(x,y) = i_2(x,y) - i_1(x,y)$ and $p(x,y) = i_2(x,y) + i_1(x,y)$. Since the spectrum $I_f(u,v)$ of the focused image, which appears in equations (5) and (6), gets cancelled, the above normalized ratio is simply:

$$\frac{M(u,v;\alpha)}{P(u,v;\alpha)} = \frac{H(u,v;(1-\alpha)e,F_e) - H(u,v;(1+\alpha)e,F_e)}{H(u,v;(1-\alpha)e,F_e) + H(u,v;(1+\alpha)e,F_e)}\,, \tag{7}$$

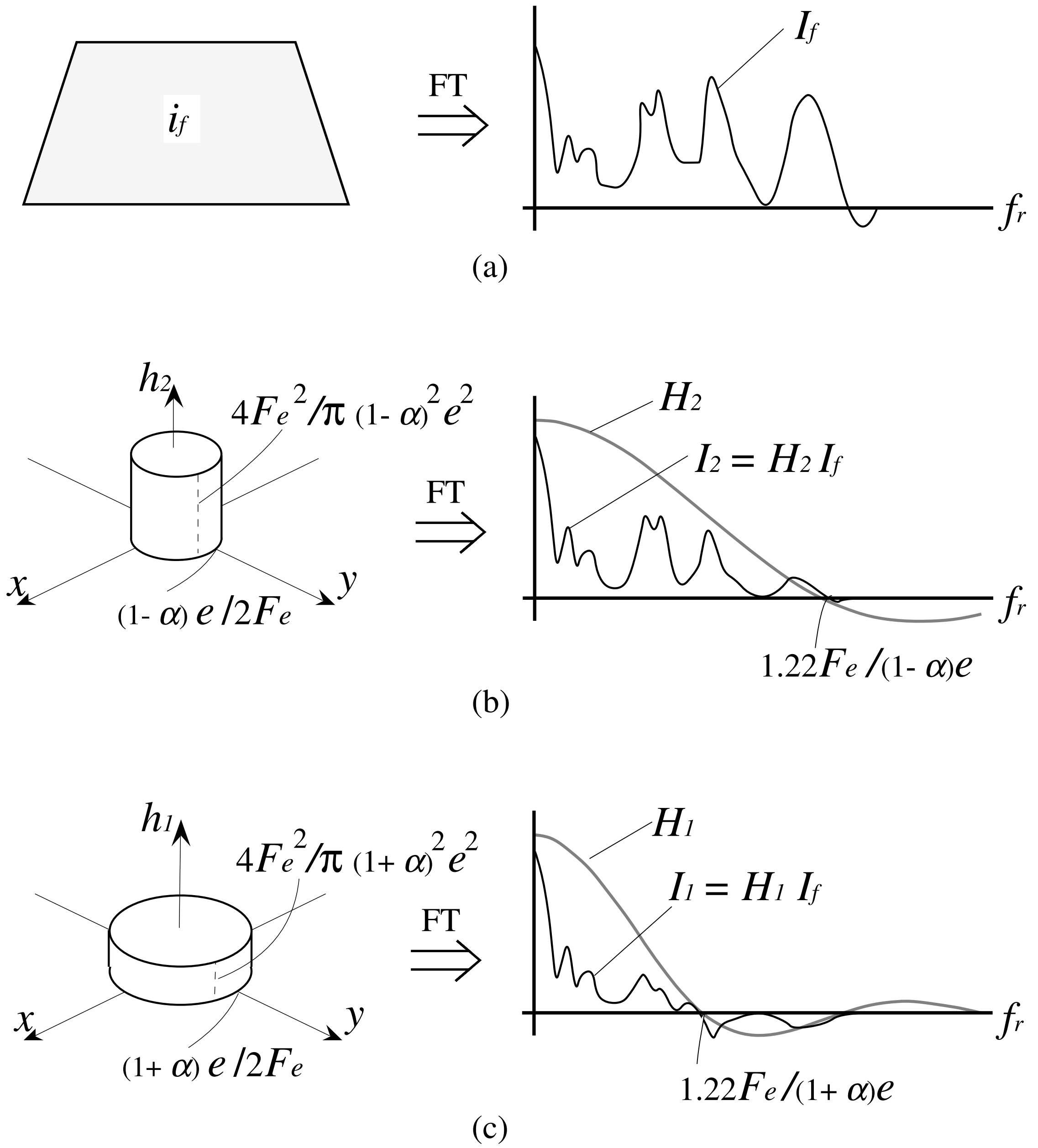


Figure 2: The effect of blurring on the near and far focused images. (a) Focused image i_f and its Fourier spectrum. (b) Pillbox defocus model h_2 and the Fourier spectrum I_2 of the blurred image. (c) Pillbox defocus model h_1 and the Fourier spectrum I_1 of the image for larger blurring. $f_r = \sqrt{u^2 + v^2}$ is the radial frequency.

Figure 3 shows the relationship between the normalized image ratio M/P and the normalized depth α for several spatial frequencies. It is seen that M/P is a monotonic function of α for $-1 \leq \alpha \leq 1$, provided the radial frequency $f_r = \sqrt{u^2 + v^2}$ is not too large. As a rule of thumb, this frequency range equals the width of the main lobe of the defocus function H when it is maximally defocused, i.e. when the distance between the focused image i_f and the sensor plane is $2e$. From the zero-crossing of the defocus function in Figure 2, the highest frequency below which the normalized image ratio M/P is monotonic is found to be:

$$f_r \leq 0.61 \, \frac{F_e}{e} \, . \tag{8}$$

For any given frequency within the above bound, since M/P is a monotonic function of α, M/P can be unambiguously mapped to a depth estimate β, as shown in Figure 3.

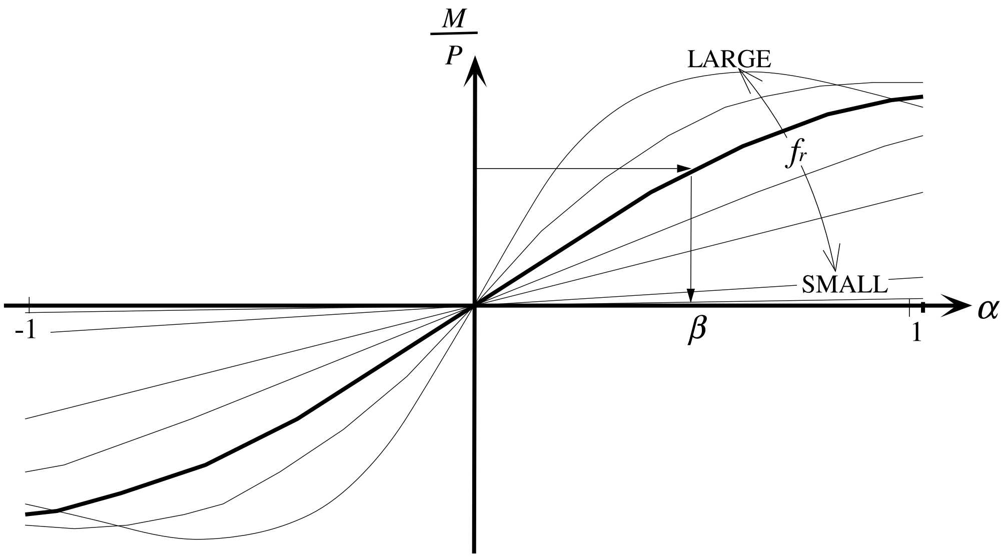


Figure 3: Relation between the normalized image ratio M/P and the defocus parameter α. An upper frequency bound can be determined, below which, M/P is a monotonic function of the defocus parameter α. For any given frequency within this bound, M/P can be unambiguously mapped to a depth estimate β.

Besides serving a critical role in our development, Figure 3 also gives us new way of viewing previous approaches to depth from defocus: If one can by some method determine the amplitudes, I_1 and I_2, of the spectra of the two defocused images at a predefined radial frequency $f_{r0} = \sqrt{{u_0}^2 + {v_0}^2}$, a unique depth estimate can be obtained. This is the basic idea that most of the previous work is based on [Pentland-1987, Gokstorp-1994, Xiong and Shafer-1995], although the ratio used in the past is simply I_1/I_2 rather than the normalized ratio M/P introduced here.

Magnitudes of the two image spectra, at a predefined frequency, can be determined using linear operators (convolution). However, this is not a trivial problem. The image texture is unknown and can include unpredictable dominant frequencies and hence it is not possible to fix *a priori* the frequency of interest. This problem may be resolved by using a large bank of narrowband filters that densely samples the frequency space to estimate powers at a large number of individual frequencies. However, important trade-offs emerge while implementing narrowband linear operators [Gokstorp-1994, Xiong and Shafer-1995]. First, such an approach is clearly inefficient from a computational perspective. Furthermore, the uncertainty relation [Bracewell-1965] tells us that, when we apply frequency analysis to a small image area, the frequency resolution reduces proportional to the inverse of the area used. To obtain a dense depth map, one must estimate $H_1 I$ and $H_2 I$ using a very small area around each pixel. A narrow filter in spatial domain corresponds to a broadband filter in frequency domain. As a result, any operator output is inevitably an average of the local image spectrum over a band of frequencies. Since the response of the defocus function H depends on the local depth α, and is not uniform within the pass-band of the operator, the output of the operator is, at best, an approximate focus measure and can result in large errors in depth.

Given that all linear operators, however carefully designed, end up having a pass-band, it would be desirable to have a set of broadband operators that together provide focus measures that are invariant to texture. Further, if the operators are broadband, a small number of them could cover the entire frequency space and avoid the use of an extensive filter bank. The result would be efficient, robust, and high-resolution depth estimation. In the next section, we describe a method to accomplish this.

3 Rational Operator Set

3.1 Modeling Relative Defocus using a Rational Expression

We have established the monotonic response of the normalized image ratio M/P to the normalized depth (or defocus) α over all frequencies (see equation (7) and Figure 3). Our objective here is to model this relation in closed form. In doing so, we would like the model to be precise and yet lead us to a small number of linear operators for depth recovery. To this end, we model the function M/P by a rational expression of two linear combinations of basis functions:

$$\frac{M(u,v;\alpha)}{P(u,v;\alpha)} = \frac{\sum_{i=1}^{n_P} G_{Pi}(u,v)\, b_{Pi}(\alpha)}{\sum_{i=1}^{n_M} G_{Mi}(u,v)\, b_{Mi}(\alpha)} + \varepsilon(u,v,\alpha)\,, \tag{9}$$

where, $b_{Pi}(\alpha)$ $(i = 1..n_P)$ and $b_{Mi}(\alpha)$ $(i = 1..n_M)$ are the basis functions, $G_{Pi}(u,v)$ and $G_{Mi}(u,v)$ are the coefficients which are functions of frequency (u,v), and $\varepsilon(u,v,\alpha)$ is the residual error of the fit of the model to the function M/P. If the model is accurate, the residual error is negligible, and it becomes possible to use the model to map the normalized image ratio M/P to the normalized depth α. The above expression can be rewritten as:

$$\frac{M(u,v;\alpha)}{P(u,v;\alpha)} = \frac{\sum_{i=1}^{n_P} G_{Pi}(u,v)\, b_{Pi}(\beta)}{\sum_{i=1}^{n_M} G_{Mi}(u,v)\, b_{Mi}(\beta)} = R(\beta;u,v)\,. \tag{10}$$

Here, α on the left hand side represents the *actual depth* of the scene point while β on the right is the *estimated depth*. A difference between the two can arise only when the residual error is non-zero. If the normalized ratio on the left side is given to us for any frequency (u,v), we can obtain the depth estimate β by solving equation (10).

The above model for the normalized image ratio is general. In principle, any basis that captures the monotonicity and structure of the normalized ratio can be used. To be specific in our discussion, we use the basis we have chosen in our implementation. Since the response of M/P to α is odd-symmetric and is almost linear for small radial frequencies f_r (see Figure 3), we could model the response using three basis functions that are powers of β:

$$n_P = 2,\quad n_M = 1,\quad b_{P1}(\beta) = \beta,\quad b_{P2}(\beta) = \beta^3,\quad b_{M1}(\beta) = 1. \tag{11}$$

Then, equation (10) becomes[3]:

$$\frac{M(u,v;\alpha)}{P(u,v;\alpha)} = \frac{G_{P1}(u,v)}{G_{M1}(u,v)}\beta + \frac{G_{P2}(u,v)}{G_{M1}(u,v)}\beta^3 = R(\beta;u,v)\,. \tag{12}$$

The term including β^3 can be seen as a small correction that compensates for the discrepancy of M/P from a linear model. From the previous section, we know that the blurring model completely determines M/P for any given depth α and frequency (u,v).

[3]We found that replacing $b_{P2}(\alpha)$ by $(\alpha - \frac{1}{a}\tanh a\alpha)$ gives us a slightly better fit when the defocus model is the pillbox function. Yet, to reduce the computational cost of solving equation (10) for depth β, we have chosen this simple polynomial model.

The above polynomial model, $R(\beta; u, v)$, can therefore be fit to the theoretical M/P in equation (7) by assuming β to be α. This gives us the unknown ratios G_{P1}/G_{M1} and G_{P2}/G_{M1} as functions of frequency (u, v). In the case of a rotationally symmetric blurring model, such as the pillbox function, these ratios reduce to functions of just the radial frequency f_r.

Now, if we fix any one of the coefficient functions, say, $G_{P1}(u, v)$, all the other coefficients can be determined from the ratios[4]. Therefore, it is possible to determine all the coefficient functions that ensure that the above polynomial model accurately fits the normalized image ratio M/P given by equation (7). Figure 4 shows an example set (based on an arbitrary selection of $G_{P1}(u, v)$) of the coefficient functions, G_{P1}, G_{P2} and G_{M1}, for the case of the pillbox blur model.

In the general form of the rational expression in equation (9), the coefficients of the rational expression can only be determined up to a multiplicative constant at each frequency. Therefore, we have:

$$\begin{aligned} G_{Pi}(u,v) &= \mathcal{G}_{Pi}(u,v)\,\mu(u,v)\,, \\ G_{Mi}(u,v) &= \mathcal{G}_{Mi}(u,v)\,\mu(u,v)\,. \end{aligned} \tag{13}$$

Here, $\mu(u, v)$ is the unknown scaling function of all the coefficient functions and $\mathcal{G}_{Pi}(u, v)$ and $\mathcal{G}_{Mi}(u, v)$ represent the structures of the ratios obtained by fitting $R(\beta; u, v)$ to $\frac{M}{P}(u, v, \alpha)$. The frequency response of the unknown scaling function $\mu(u, v)$ is needed to determine all the coefficient functions without ambiguity. How this can be accomplished for the general rational expression will be described in section 4.1.

We now examine how well the polynomial model fits the plots in Figure 3 of the normalized ratio $\frac{M}{P}(u, v, \alpha)$. More precisely, we are interested in knowing how well the model can used to estimate depth. To this end, for each frequency, we select a "true" depth value α and find the corresponding ratio M/P using the analytical expression in (7). This ratio is then plugged into the polynomial model of (12) to calculate the depth estimate β using the Newton-Raphson method. This process is repeated for all frequencies.

Let us rewrite equation (12) as:

$$p_0(u, v; \alpha) = p_1(u, v)\beta + p_3(u, v)\beta^3\,. \tag{14}$$

As the third-order term can be considered to be a small correction, the following initial

[4]In practice, $G_{P1}(u, v)$ cannot be selected arbitrarily. There are other restrictions that need to be considered. The exact selection procedure is discussed later in section 4.1.

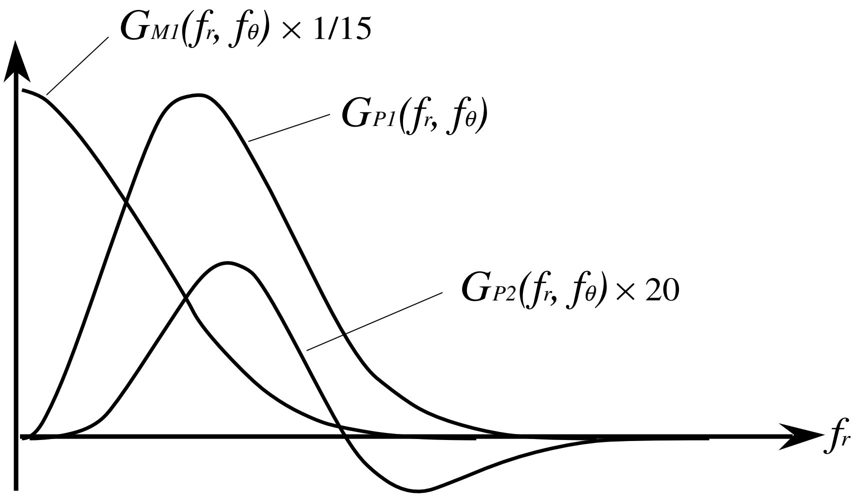


Figure 4: An example set of the coefficient functions obtained by fitting the polynomial model to the normalized image ratio M/P. Here, $G_{P1}(u, v)$ was chosen and the remaining two functions determined from the fit.

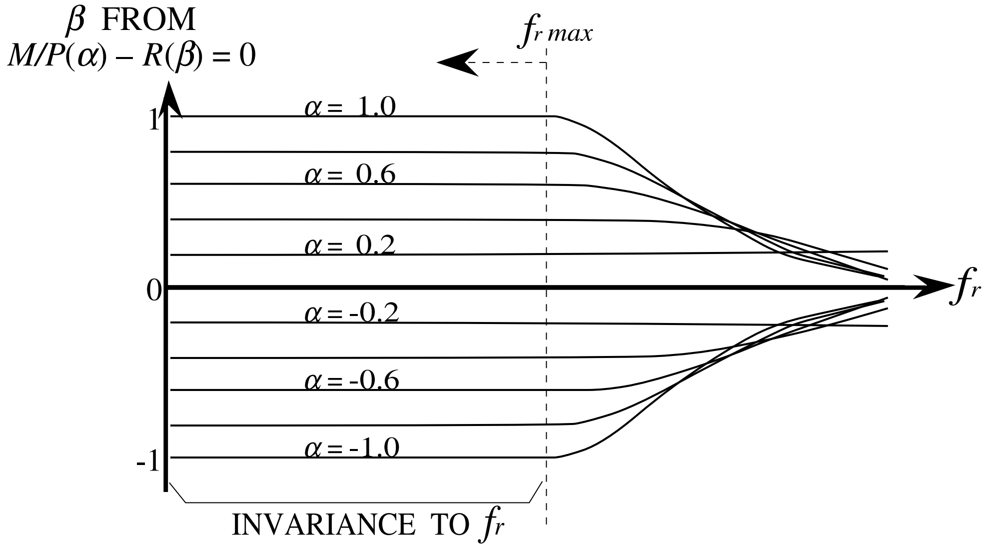


Figure 5: Depth β, estimated using the polynomial model in equation (12), is plotted as a function of spatial frequency for different values of actual depth α. We see that the estimated depth equals the actual depth and is invariant to frequencies within the upper bound $f_{r\,max}$ given by equation (17).

value can be provided to the Newton-Raphson method:

$$\beta_0(u,v) \;=\; \frac{p_0(u,v;\alpha)}{p_1(u,v)}\,. \tag{15}$$

Then, the solution after one iteration is:

$$\begin{aligned}\beta(u,v) \;&=\; \beta_0(u,v) - \frac{-p_0(u,v;\alpha) + p_1(u,v)\beta_0 + p_3(u,v){\beta_0}^3}{p_1(u,v) + 3p_3(u,v){\beta_0}^2}\\ &=\; \beta_0(u,v) - \frac{p_3(u,v){\beta_0}^3}{p_1(u,v) + 3p_3(u,v){\beta_0}^2}\end{aligned} \tag{16}$$

Figure 5 shows that the estimated depth β is, for all practical purposes, equal to the actual depth, indicating that the polynomial model is indeed accurate. Further, the estimated depth is invariant (insensitive) to texture frequency as far as the radial frequency f_r is below $f_{r\,max}$. Above this frequency limit $f_{r\,max}$, the response of $\frac{M}{P}(u,v;\alpha)$ to α, shown in Figure 3, becomes non-monotonic within the region $-1 \leq \alpha \leq 1$ and hence an accurate depth estimate is not obtainable. In practice, any image can be convolved using a passband filter to ensure that all frequencies above $f_{r\,max}$ are removed. The rule of thumb used to determine $f_{r\,max}$ is given by equation (8). However, for the pillbox blur model, we have found via numerical simulation that $f_{r\,max}$ is in fact 1.2 times larger[5] than the limit given by equation (8).

$$f_{r\,max} \;=\; 1.2 \cdot 0.61\,\frac{F_e}{e} \;=\; 0.73\,\frac{F_e}{e}\,. \tag{17}$$

This is a valuable side-effect of introducing the normalized image ratio M/P; we can utilize 20% more frequency spectrum information than conventional methods which use the ratio I_1/I_2.

3.2 Rational Operator Set

We have introduced a rational expression model for the normalized ratio M/P and shown that the solution of equation (10) gives us robust depth estimates for all frequencies within a permissible range. Thus far, this robustness was demonstrated for individual frequencies. In this section, we show how the rational model can be used to design a small set of broadband operators that can handle arbitrary textures.

[5]This number can be increased from 1.2 to 1.3 if a larger number of Newton-Raphson iterations are used. However, depth results in this additional range are not numerically stable in the presence of noise since the response curves of M/P tend to flatten out. Hence, we use only one iteration.

Taking cross-products in equation (10), we get:

$$\sum_{i=1}^{n_M} M(u,v;\alpha)\, G_{Mi}(u,v)\, b_{Mi}(\beta) \;=\; \sum_{i=1}^{n_P} P(u,v;\alpha)\, G_{Pi}(u,v)\, b_{Pi}(\beta)\,. \tag{18}$$

By integrating over the entire frequency space, we get:

$$\sum_{i=1}^{n_M} c_{Mi}(\alpha)\, b_{Mi}(\beta) \;=\; \sum_{i=1}^{n_M} c_{Pi}(\alpha)\, b_{Pi}(\beta)\,, \tag{19}$$

where:

$$\begin{aligned} c_{Mi}(\alpha) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} M(u,v;\alpha)\, G_{Mi}(u,v)\, du\, dv\,, \\ c_{Pi}(\alpha) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(u,v;\alpha)\; G_{Pi}(u,v)\; du\, dv\,. \end{aligned} \tag{20}$$

Here, we invoke the power theorem [Bracewell-1965]:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)\, G(u,v)\, du\, dv \;=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, g(-x,-y)\, dx\, dy\,, \tag{21}$$

where, $F(u,v)$ and $G(u,v)$ are the Fourier transforms of functions $f(x,y)$ and $g(x,y)$, respectively. Since we are conducting a spatial-frequency analysis, that is, we are analyzing the frequency content in a small area centered around each pixel, the right hand side of equation (20) is nothing but a convolution. This implies that $c_{Mi}(\alpha)$ and $c_{Pi}(\alpha)$ are actually functions of (x,y) and can be determined by convolutions as:

$$\begin{aligned} c_{Mi}(x,y;\alpha) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} m(x',y';\alpha)\, g_{Mi}(x-x',y-y')\, dx'\, dy'\,, \\ c_{Pi}(x,y;\alpha) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x',y';\alpha)\; g_{Pi}(x-x',y-y')\; dx'\, dy'\,, \end{aligned} \tag{22}$$

where, $g_{Mi}(x,y)$ and $g_{Pi}(x,y)$ are the inverse Fourier transforms of $G_{Mi}(u,v)$ and $G_{Pi}(u,v)$, respectively. In short, all the coefficients needed to compute depth using the polynomial in equation (19) can be determined by convolving the difference image $m(x,y)$ and the summed image $p(x,y)$ with linear operators that are spatial domain equivalents of the coefficient functions. We refer these as *rational operators*. The outputs of these operators at each pixel (x,y) are plugged into equation (19) to determine depth $\beta(x,y)$.

As an example, if we use the model in equation (12), the depth recovery equation (19) becomes:

$$c_{M1}(x,y;\alpha) \;=\; c_{P1}(x,y;\alpha)\,\beta \;+\; c_{P2}(x,y;\alpha)\,\beta^3\,. \tag{23}$$

By substituting equation (22), we have:

$$g_{M1}(x,y) * m(x,y;\alpha) \;=\; g_{P1}(x,y) * p(x,y;\alpha)\,\beta \;+\; g_{P2}(x,y) * p(x,y;\alpha)\,\beta^3\,. \tag{24}$$

Again, the above rational operators are nothing but inverse Fourier transforms of the coefficient functions shown in Figure 4. We see that, though the operators are all broadband (see Figure 4), the above recovery equation is independent of scene texture and provides an efficient means of computing precise depth estimates.

4 Implementation of Rational Operators

The previous section described the theory underlying rational operators. In this section, we discuss various design and implementation issues that must be addressed to ensure that the rational operators produce accurate depth from defocus. In particular, we describe the design procedure used to optimize rational operator kernels, the estimation of a depth confidence measure, prefiltering of images prior to application of the rational operators, and the post-processing of the outputs of the operators.

Since the rational expression model of equation (9) is too general, we focus on the simpler model of equation (12) which we used in our experiments. However, the procedures described here can be applied to other forms of the rational model.

4.1 Design of Rational Operators Kernels

Since our rational operators are broadband linear filters, we can implement them with small convolution kernels. This is beneficial for two reasons; (a) low computational cost and (b) high spatial resolution. However, as we shall see, the design problem itself is not trivial.

Note that, after deriving the operators, the functions $\mathcal{G}_{Pi}(u,v)$ and $\mathcal{G}_{Mi}(u,v)$ in equation (13) must have a ratio that equals the one obtained by fitting the polynomial model to the normalized image ratio. Any discrepancy in this ratio would naturally cause depth estimation errors. Fortunately, the base form function $\mu(u,v)$ of equation (13) remains at our discretion and can be adjusted to minimize such discrepancies. This does not imply that $\mu(u,v)$ will be selected arbitrarily, but rather that it will be given a convenient initial form that can be optimized. Clearly, the effect of discrepancies in the ratio would vary with frequency and hence depend on the textural properties of the scene. The design of the operator kernels is therefore done by minimizing an objective function that represents ratio errors over all frequencies. The relation between depth estimation error and ratio error is derived in appendix 7.2. We argue in the appendix that, for the depth error to be kept at a minimum, the ratio errors must satisfy the following condition:

$$
\begin{aligned}
\sigma_{\mathcal{G}_{M1}}(u,v) &= \kappa \frac{\mathcal{G}_{M1}(u,v)}{P(u,v;\alpha)\ G_{P1}(u,v)} \\
\sigma_{\mathcal{G}_{P2}}(u,v) &= \kappa \frac{1}{P(u,v;\alpha)\, G_{P1}(u,v)} ,
\end{aligned}
\tag{25}
$$

$\sigma_{\mathcal{G}_{M1}}(u,v)$ and $\sigma_{\mathcal{G}_{P2}}(u,v)$ determine the weighting functions to be used in the minimization of errors in $\mathcal{G}_{M1}(u,v)$ and $\mathcal{G}_{P2}(u,v)$. Here, κ is a constant and in the derivation of these expressions we have set $\mu(u,v)$ equal to $G_{P1}(u,v)$, i.e. $\mathcal{G}_{P1}(u,v)=1$. Therefore, from equation (13) we have $\mu(u,v) = G_{P1}(u,v)$, $G_{M1}(u,v) = \mathcal{G}_{M1}(u,v)G_{P1}(u,v)$ and $G_{P2}(u,v) = \mathcal{G}_{P2}(u,v)G_{P1}(u,v)$.

Now, we are in a position to formulate our objective function for operator design as follows:

$$
\begin{aligned}
\chi^2 &= \sum_{(u_i,v_i)\neq(0,0)} \left[\left(\frac{\mathcal{G}'_{M1}(u_i,v_i) - \mathcal{G}_{M1}(u_i,v_i)}{\sigma_{\mathcal{G}_{M1}}(u_i,v_i)} \right)^2 + \left(\frac{\mathcal{G}'_{P2}(u_i,v_i) - \mathcal{G}_{P2}(u_i,v_i)}{\sigma_{\mathcal{G}_{P1}}(u_i,v_i)} \right)^2 \right] + \\
&\qquad \left(\frac{G'_{M1}(0,0) - G_{M1}(0,0)}{\sigma_{G_{M1\,o}}} \right)^2 ,
\end{aligned}
\tag{26}
$$

where, $\mathcal{G}'_{M1}(u,v)$ and $\mathcal{G}'_{P2}(u,v)$ are the actual ratios of the designed discrete kernels, $\mathcal{G}_{M1}(u,v)$ and $\mathcal{G}_{P2}(u,v)$ are the ratios obtained in the previous section by fitting the polynomial model to the normalized image ratio, and $\sigma_{G_{M1\,o}}$ is a constant used to ensure that the minimization of χ^2 does not produce the trivial result of zero-valued operators. $G'_{M1}(0,0)$ is the actural DC response of the designed discrete kernel g_{M1}, and $G_{M1}(0,0)$ is its initial value. In the above summation, the discrete frequency samples (u_i,v_i) should be sufficiently dense. When the kernel size is $n \times n$, the frequency samples should be at least $2n \times 2n$ in order to avoid the Gibbs phenomenon [Oppenheim and Schafer-1989]. In our optimization, we use 32×32 sample points for 7×7 kernels. Since χ^2 is non-linear, its minimization is done using the Levenberg-Marquardt algorithm [Press *et al.*-1992].

We still need to define $P(u,v;\alpha)$ in equation (25), which is dependent on the unknown texture of the image. However, since $P(u,v;\alpha)$ is only used to fix the weighting functions in equation (25), a rough approximation suffices. To this end, we assume the distribution of the image spectrum to be:

$$
|I(f_r,f_\theta)| \propto 1/{f_r}^n . \tag{27}
$$

In our optimization we have used $n = 1.5$, which corresponds to Brownian motion[6]. Though $P(u,v;\alpha)$ changes with α, we can use the approximation $P(u,v;\alpha) = I(u,v)$.

[6]If we denote fractal dimension [Peitgen and Saupe-1988] by D_h, in the two dimensional case the

The last issue concerns the base form function $\mu(u,v) = G_{P1}(u,v)$ in equation (25). An initial selection can be made for this function that will be refined by the optimization of χ^2. As $\mathcal{G}_{M1}(u,v) = G_{M1}(u,v)/G_{P1}(u,v)$ is infinity[7] when $|(u,v)| \rightarrow 0$, $G_{P1}(0,0)$ must be 0 in order to realize $G_{M1}(u,v) = \mathcal{G}_{M1}(u,v)G_{P1}(u,v)$ using a finite kernel. Also, $G_{P1}(u,v)$ must be smooth (without rapid fluctuations) to obtain rational operators with small kernels. In our implementation, we have imposed rotational symmetry as an added constraint and used the Laplacian of Gaussian to initialize $G_{P1}(u,v)$:

$$G_{P1}(f_r) = (\frac{f_r}{f_{peak}})^2 \exp(1 - (\frac{f_r}{f_{peak}})^2)\,, \tag{28}$$

where, f_{peak} is the radial frequency at which G_{P1} is maximum. This frequency is set to $0.4 f_{Nyquist}$ in our optimization. Once again, the above function is only used for initialization and is further refined by the optimization of χ^2. An example set of discrete rational operators obtained from the optimization of χ^2 will be presented shortly.

4.2 Prefiltering

We now discuss prefiltering that needs to be applied to the input images $i_1(x,y)$ and $i_2(x,y)$, or, $p(x,y)$ and $m(x,y)$. The purpose is to remove the DC component and very high frequencies before applying the rational filters. The DC component is harmful because a small change in the illumination, between the two images, $i_1(x,y)$ and $i_2(x,y)$, can cause an unanticipated bias in the image $m(x,y)$. Such a bias would propagate errors to the coefficient image $c_{M1}(x,y)$ since the G_{M1} operator applied to $m(x,y)$ is essentially a low-pass filter. This, in turn, would cause depth errors. At the other end of the spectrum, radial frequencies greater than $f_{r\,max}$ (see equation (17)) are also harmful as they violate the monotonicity property of M/P, which is needed for rational operators to work. Therefore, such high frequencies must also be removed.

Although it is possible to embed the desired prefilter within the rational filters (given that prefiltering can be done using linear operators), we have chosen to use a separate prefilter for the following reason. Since the prefilter attempts to cut low and high frequencies, it tends to have a large kernel. Embedding such a prefilter in the rational operators would require the operators also to have large kernels, thus, resulting in low spatial resolution as well as unnecessary additional computations.

relation $n = 4 - D_h$ holds true. $D_h = 3, n = 1$ corresponds to the case of extreme fractal, $D_h = 2.5, n = 1.5$ corresponds to Brownian motion and $D_h = 2, n = 2$ corresponds to a smooth image. Finally, $n = 0$ corresponds to white noise (completely random image).

[7] In equation (12), M/P is zero when $|(u,v)| \rightarrow 0$. Since α can be non-zero, $1/\mathcal{G}_{M1}(u,v) = G_{P1}(u,v)/G_{M1}(u,v)$ must be zero for equation (12) to be valid.

As with the rational operators, the design of the prefilter can be posed as the optimization of an objective function. Let us define the desirable frequency response of the prefilter as $f(u, v)$. For reasons stated earlier, this frequency response must cut both the DC component and high frequencies. In addition, the frequency response should be smooth and rotationally symmetric to ensure a small kernel size. A function with these desired properties is again the Laplacian of Gaussian given by the right hand side of equation (28), but using $f_{peak} = 0.4 f_{r\,max}$. We define the objective function as:

$$\chi_p{}^2 = \sum_{(u_i, v_i) \in passband} \left(\frac{f'(u_i, v_i) - f(u_i, v_i)}{\sigma_{pass}} \right)^2 + \sum_{(u_i, v_i) \in stopband} \left(\frac{f'(u_i, v_i) - f(u_i, v_i)}{\sigma_{stop}} \right)^2 \qquad (29)$$

where, $f'(u, v)$ is the frequency response of the designed prefilter kernel. σ_{pass} and σ_{stop} represent the weights assigned to the passband and the stopband regions of the prefilter, respectively. The stopband is $(u_i, v_i) = (0, 0)$ and $\sqrt{u_i{}^2 + v_i{}^2} > f_{r\,max}$. The Levenberg-Marquardt algorithm [Press *et al.*-1992] is used to determine the prefilter kernel that minimizes $\chi_p{}^2$.

4.3 An Example Set of Discrete Rational Filters

Figures 6 and 7 show the kernels and their frequency responses for the rational operators and the prefilter, derived with kernel size set to 7×7 and $e/F_e = 2.307$ pixels. In order to make the operators uniformly sensitive to textures in all directions, we imposed the constraint that the kernels must be symmetric with respect to the x and y axes as well as the lines $y = x$ and $y = -x$. These constraints reduce the number of degrees of freedom (DOF) in the kernel design problem. In the case of a 7×7 kernel, the DOF is reduced to 10. This further reduces to 6 for a 6×6 or a 5×5 kernel. This DOF of 6 is too small to design operators with the desired frequency responses. Therefore, the smallest kernel size was chosen to be $k_s = 7$. Note that the passband response of the prefilter in Figure 7 can be further refined if its kernel size is increased.

The final design issue pertains to the maximum frequency $f_{r\,max}$. Since the discrete Fourier transform of a kernel of size k_s has the minimum discrete frequency period of $1/k_s$, it is difficult to obtain precisely any response in the frequency region below $1/k_s$. Further, the spectrum in this region is going to be suppressed by the prefilter as it is close to the DC component. Therefore, the maximum frequency $f_{r\,max}$ must be well above $1/k_s$. We express this condition as $f_{r\,max} \geq 2\frac{1}{k_s}$. Using equation (17), we obtain:

$$\frac{2e}{F_e} \leq 0.73 k_s \,. \qquad (30)$$

This condition can be interpreted as follows: The maximum blur circle diameter $2e/F_e$ must be smaller than 73% of the kernel size k_s. This is also intuitively reasonable as the kernel should be larger than the blur circle as it seeks to measure blur[8].

4.4 Coefficient Image Smoothing

By applying the prefilter and the rational operators in Figure 6 to the images $m(x,y)$ and $p(x,y)$, we obtain coefficients that can plugged into equation (23) to compute depth β. However, a problem can arise in solving for depth. If $c_{P1}(x,y) = g_{P1}(x,y) * p(x,y;\alpha)$ in equation (24) is close to zero, the depth estimate becomes unstable as is evident from the solution step in equation (15). Since the frequency response of $g_{P1}(x,y)$ cuts the DC component (Figure 5 (a)), zero-crossings are usually common in the coefficient image $c_{P1}(x,y;\alpha)$. It is also obvious that, for image areas with weak texture[9], $c_{P1}(x,y;\alpha)$ approaches zero.

To solve this problem, we apply a smoothing operator to the coefficient image. This enables us to avoid unstable depth estimates at zero-crossings in the coefficient image, which otherwise must be removed by some ad hoc post-filtering. To optimize this smoothing operation, so as to minimize depth errors, we need an analytic model of depth error. Using the depth recovery equation (23), we get:

$$dc_{M1} \;=\; dc_{P1}\,\beta + c_{P1}\,d\beta + dc_{P2}\,\beta^3 + 3c_{P2}\,\beta^2\,d\beta\,. \tag{31}$$

Here, we have dropped the parameter (x,y) for brevity. Solving for $d\beta$, we get:

$$d\beta \;=\; \frac{dc_{M1} - dc_{P1}\,\beta - dc_{P2}\,\beta^3}{c_{P1} + 3c_{P2}\,\beta^2}\,. \tag{32}$$

As c_{P2} is only a small correction factor, the following approximation can be made:

$$d\beta \;=\; \frac{dc_{M1} - dc_{P1}\,\beta - dc_{P2}\,\beta^3}{c_{P1}}\,. \tag{33}$$

We denote the standard deviations (errors) of c_{M1}, c_{P1} and c_{P2} by $\sigma_{C_{M1}}$, $\sigma_{C_{M1}}$ and $\sigma_{C_{M1}}$, respectively. To simplify matters, it is assumed that the errors are independent of each other. Then, we get [Hoel-1971]:

$${\sigma_\beta}^2 \;=\; \frac{{\sigma_{c_{M1}}}^2 + \beta^2\,{\sigma_{c_{P1}}}^2 + \beta^6\,{\sigma_{c_{P2}}}^2}{{c_{P1}}^2}\,. \tag{34}$$

[8] Since the above conditions related to kernel size are rough, we suggest that the linearity of depth estimation be checked (using synthetic images) to find the best kernel size k_s. Such an evaluation is reported in the experimental section.

[9] Weak texture is equivalent to low spectrum power in the high frequency region.

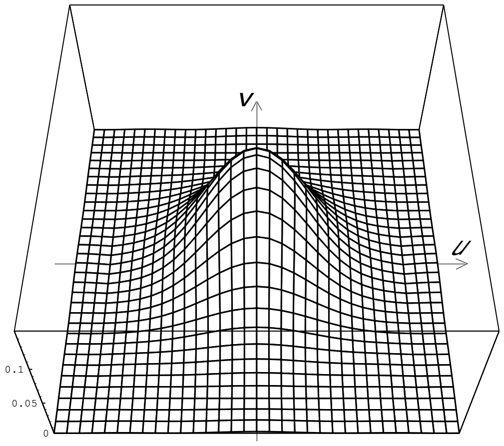
$$
g_{M1} = \begin{pmatrix}
-0.00133 & 0.0453 & 0.1799 & 0.297 & 0.1799 & 0.0453 & -0.00133 \\
0.0453 & 0.4009 & 0.8685 & 1.093 & 0.8685 & 0.4009 & 0.0453 \\
0.1799 & 0.8685 & 2.957 & 4.077 & 2.957 & 0.8685 & 0.1799 \\
0.297 & 1.093 & 4.077 & 6.005 & 4.077 & 1.093 & 0.297 \\
0.1799 & 0.8685 & 2.957 & 4.077 & 2.957 & 0.8685 & 0.1799 \\
0.0453 & 0.4009 & 0.8685 & 1.093 & 0.8685 & 0.4009 & 0.0453 \\
-0.00133 & 0.0453 & 0.1799 & 0.297 & 0.1799 & 0.0453 & -0.00133
\end{pmatrix}
$$

$$
g_{P1} = \begin{pmatrix}
-0.03983 & -0.09189 & -0.198 & -0.259 & -0.198 & -0.09189 & -0.03983 \\
-0.09189 & -0.3276 & -0.4702 & -0.4256 & -0.4702 & -0.3276 & -0.09189 \\
-0.198 & -0.4702 & 0.3354 & 1.393 & 0.3354 & -0.4702 & -0.198 \\
-0.259 & -0.4256 & 1.393 & 3.385 & 1.393 & -0.4256 & -0.259 \\
-0.198 & -0.4702 & 0.3354 & 1.393 & 0.3354 & -0.4702 & -0.198 \\
-0.09189 & -0.3276 & -0.4702 & -0.4256 & -0.4702 & -0.3276 & -0.09189 \\
-0.03983 & -0.09189 & -0.198 & -0.259 & -0.198 & -0.09189 & -0.03983
\end{pmatrix}
$$

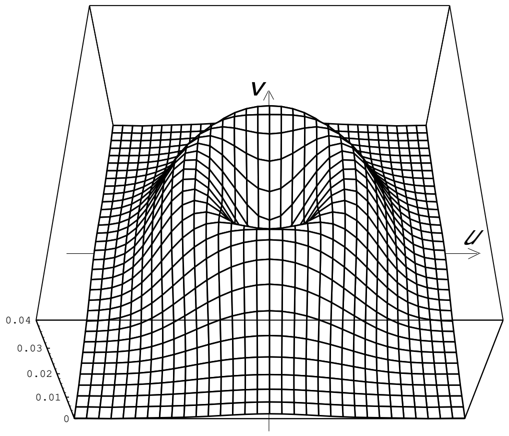
$$
g_{P2} = \begin{pmatrix}
0.05685 & -0.02031 & -0.06835 & -0.06135 & -0.06835 & -0.02031 & 0.05685 \\
-0.02031 & -0.06831 & 0.05922 & 0.1454 & 0.05922 & -0.06831 & -0.02031 \\
-0.06835 & 0.05922 & 0.1762 & -0.01998 & 0.1762 & 0.05922 & -0.06835 \\
-0.06135 & 0.1454 & -0.01998 & -0.698 & -0.01998 & 0.1454 & -0.06135 \\
-0.06835 & 0.05922 & 0.1762 & -0.01998 & 0.1762 & 0.05922 & -0.06835 \\
-0.02031 & -0.06831 & 0.05922 & 0.1454 & 0.05922 & -0.06831 & -0.02031 \\
0.05685 & -0.02031 & -0.06835 & -0.06135 & -0.06835 & -0.02031 & 0.05685
\end{pmatrix}
$$

$$
prefilter = \begin{pmatrix}
-0.143 & -0.1986 & -0.1056 & -0.07133 & -0.1056 & -0.1986 & -0.143 \\
-0.1986 & -0.1927 & 0.01795 & 0.07296 & 0.01795 & -0.1927 & -0.1986 \\
-0.1056 & 0.01795 & 0.2843 & 0.4601 & 0.2843 & 0.01795 & -0.1056 \\
-0.07133 & 0.07296 & 0.4601 & 0.6449 & 0.4601 & 0.07296 & -0.07133 \\
-0.1056 & 0.01795 & 0.2843 & 0.4601 & 0.2843 & 0.01795 & -0.1056 \\
-0.1986 & -0.1927 & 0.01795 & 0.07296 & 0.01795 & -0.1927 & -0.1986 \\
-0.143 & -0.1986 & -0.1056 & -0.07133 & -0.1056 & -0.1986 & -0.143
\end{pmatrix}
$$

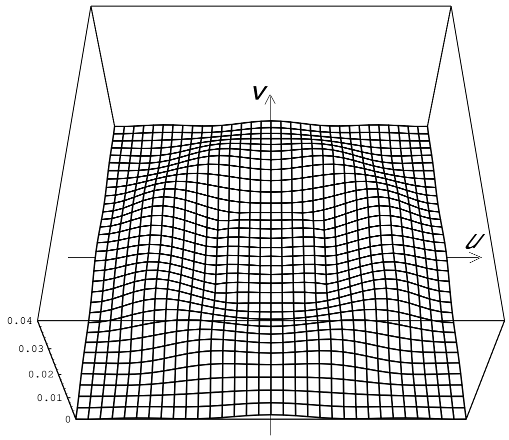
Figure 6: Rational operator kernels derived using kernel size of 7×7 and $e/F_e = 2.307$ pixels. Regardless of scene texture, passive depth from defocus can be accomplished using this small operator set.



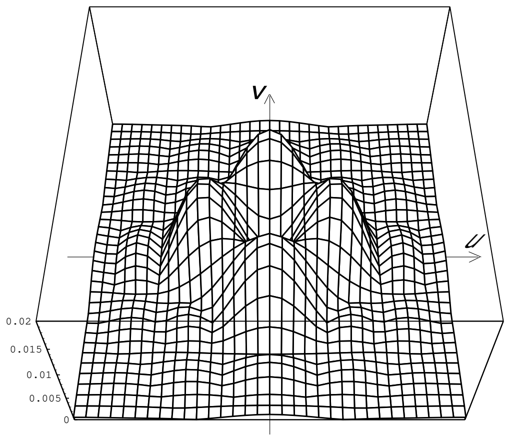
(a) g_{M1}



(b) g_{P1}



(c) g_{P2}



(d) prefilter

Figure 7: Frequency responses of the rational operators shown in Figure 6.

This expression is useful as it gives us an estimate of depth error. The inverse of this estimate, $1/\sigma_\beta{}^2$, can be viewed as a *depth confidence* measure and be used to combine adjacent depth estimates in a maximum likelihood sense to obtain more accurate depth estimates. Also, when one wishes to apply depth from defocus at different scales using a *pyramid framework* [Jolion and Rosenfeld-1994, Burt and Adelson-1983, Darrell and Wohn-1988, Gokstorp-1994], the above confidence measure can be used to combine depth values at different levels of the pyramid.

In equation (34), $\sigma_{c_{M1}}$, $\sigma_{c_{P1}}$ and $\sigma_{c_{P2}}$ are constants because they are defined by the readout noise of the image sensor used and the frequency responses of the rational operators. On the other hand, β can be assumed to be locally constant, since depth can be expected to vary smoothly at most points in the image. These facts lead us to:

$$\sigma_\beta{}^2 \propto \frac{1}{c_{P1}{}^2}. \tag{35}$$

With the above error model in place, we can develop a method for coefficient image smoothing. If we multiply equation (23) by $c_{P1}(x,y;\alpha)$, and sum up depth values in the neighborhood R of each pixel, we get:

$$\sum_{(x,y)\in R} c_{P1}(x,y;\alpha)\, c_{M1}(x,y;\alpha) = \sum_{(x,y)\in R} c_{P1}{}^2(x,y;\alpha)\,\beta_a + \sum_{(x,y)\in R} c_{P1}(x,y;\alpha)\, c_{P2}(x,y;\alpha)\,\beta_a{}^3. \tag{36}$$

where, β_a is the depth estimate after coefficient smoothing. Since the last terms in equations (36) and (23) are small corrections, β_a can be approximated by:

$$\begin{aligned}\beta_a &\simeq \frac{\sum_{(x,y)\in R} (c_{P1}(x,y;\alpha)\, c_{M1}(x,y;\alpha))}{\sum_{(x,y)\in R} c_{P1}{}^2(x,y;\alpha)} = \frac{\sum_{(x,y)\in R} (c_{P1}{}^2(x,y;\alpha)\,\frac{c_{M1}(x,y;\alpha))}{c_{P1}(x,y;\alpha)})}{\sum_{(x,y)\in R} c_{P1}{}^2(x,y;\alpha)} \\ &\simeq \frac{\sum_{(x,y)\in R} (c_{P1}{}^2(x,y;\alpha)\,\beta(x,y;\alpha))}{\sum_{(x,y)\in R} c_{P1}{}^2(x,y;\alpha)}.\end{aligned} \tag{37}$$

From equation (35), we see that β_a is the weighted average of raw depth estimates β in the neighborhood R, where the weights are $1/\sigma_\beta{}^2(x,y;\alpha)$.

From statistics [Hoel-1971] we know that the optimal weighted average of independent variables X_i ($i = 1..N$) whose variances are $\sigma_i{}^2$, is obtained by weighting the X_i with $1/\sigma_i{}^2$. Therefore, the above weighted average of depth estimates can be viewed as optimal. The variance $\sigma_a{}^2$ of the resulting depth estimate β_a is given by:

$$\frac{1}{\sigma_a{}^2} = \sum_{i=1..N} \frac{1}{\sigma_i{}^2}. \tag{38}$$

Hence, the coefficient smoothing of equation (36) is optimal, in that, it minimizes[10] the error in estimated depth β_a. In addition, the resulting smoothed coefficient $c_{P1}(x,y;\alpha)$ is proportional to the inverse of the variance of β_a, i.e. $1/\sigma_{\beta_a}{}^2$, which is clear from equations (35), (37) and (38). Therefore, the smoothed coefficient $c_{P1}(x,y;\alpha)$ can be used as a confidence measure to post-process computed depth maps.

[10]Another method to cope with zero-crossings in the c_{P1} coefficient image is based on the Hilbert transform [Bracewell-1965, Oppenheim and Schafer-1989]. This approach is detailed in [Watanabe and Nayar-1995a].

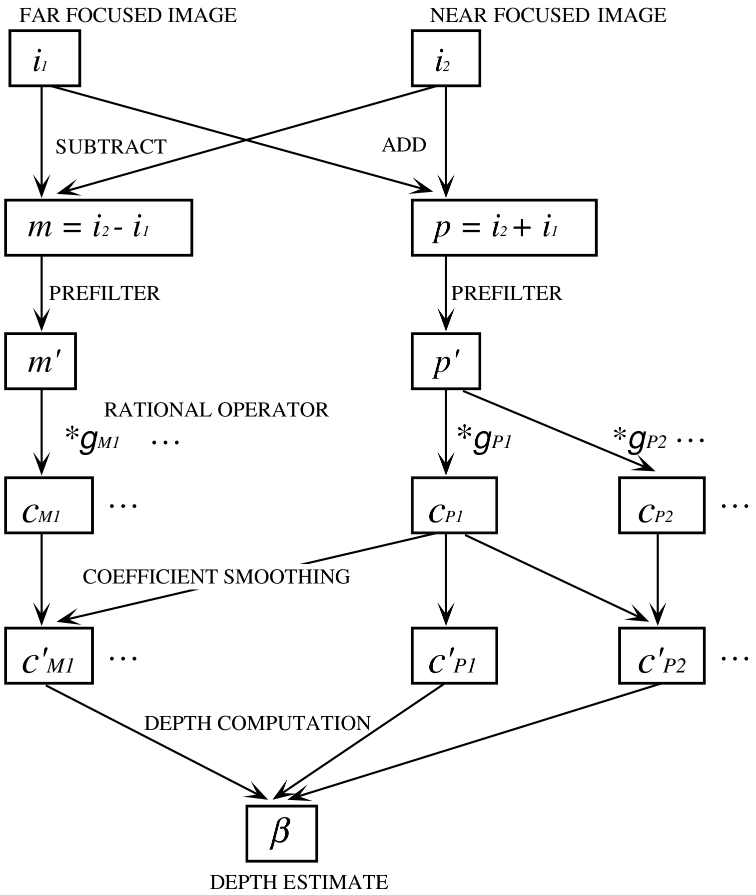


Figure 8: The flow of the depth from defocus algorithm. Using Datacube's MV200 pipeline processor, the entire algorithm can be executed in as little as 0.16 msec to obtain a 512×480 depth map.

4.5 Algorithm

Figure 8 illustrates the flow of the depth from defocus algorithm we have implemented. The far and near focused images are first added and subtracted to produce $p(x, y)$ and $m(x, y)$, respectively. Then they are convolved with the prefilter and subsequently with the three rational operators. The resulting coefficient images are then smoothed by local averaging. The final step is the computation of depth from the coefficients using a single iteration of the Newton-Raphson method using equations (15) and (16). Alternatively, depth computation can be achieved using a precomputed two-dimensional look-up table. The look-up table is configured to take $c'_{M1}(x, y)/c'_{P1}(x, y)$ and $c'_{P2}(x, y)/c'_{P1}(x, y)$ as inputs and provides depth $\beta(x, y)$ as output. In summary, a depth map is generated with as few as 5 two-dimensional convolutions, simple smoothing of the coefficient images, and a straightforward depth computation step.

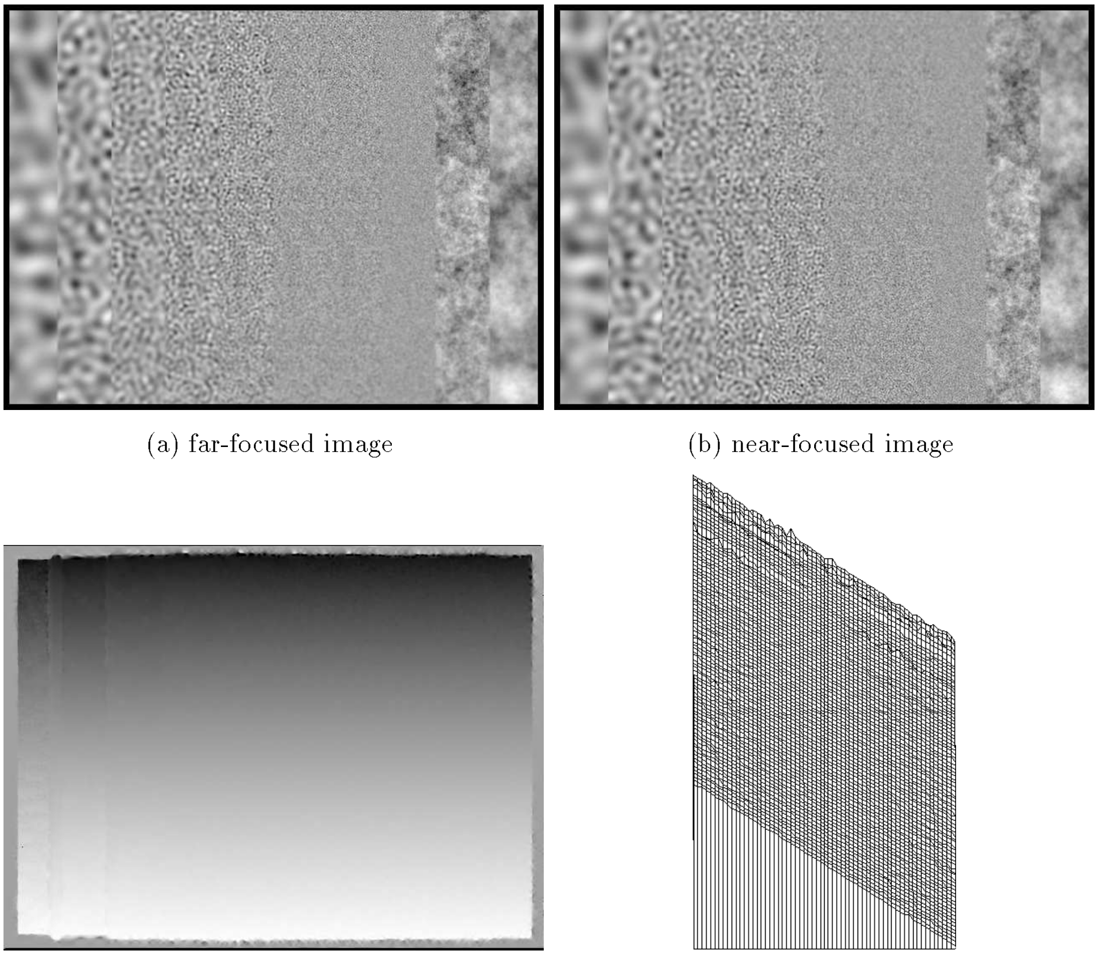
The above operations can be executed efficiently using a pipelined image processor. If one uses Datacube's MV200 pipeline processor, all the computations can be realized using as few as 10 pipelines. The entire depth from defocus algorithm can then be executed in 0.16 msec for an image size of 512×480. The efficiency of the algorithm, which comes from the use of the rational operator set, is far superior to any existing depth from defocus algorithm that attempts to compute accurate depth estimates [Xiong and Shafer-1993, Gokstorp-1994].

5 Experiments

5.1 Experiments with Synthetic Images

We first illustrate the linearity of depth estimation and its invariance to texture frequency using synthetic images. The synthetic images shown in Figure 9 correspond to a planar surface that is inclined away from the sensor such that its normalized depth value is 0 at the top and 255 at the bottom. The plane includes 10 vertical strips with different textural properties. The left 7 strips have textures with narrow power spectra whose central frequencies are 0.015, 0.03, 0.08, 0.13, 0.18, 0.25 and 0.35, from left to right. The 8^{th} strip is white noise. The next two strips are fractals with dimensions of 3 and 2.5, respectively [Peitgen and Saupe-1988]. The near and far focused images were generated using the pillbox blur model. The defocus condition used was $e/F_e = 2.307$ pixels. In all our experiments, the digital images used are of size 640×480. The depth map estimated using the 7×7 rational operators and 5×5 coefficient smoothing is shown as a gray-coded image in Figure 9(c) and a wireframe in Figure 9(d). As is evident, the proposed algorithm produces high accuracy despite the significant texture variations between the vertical strips.

Figure 10 summarizes quantitative results obtained from the above experiment. The figure includes plots of (a) the gradient of the estimated depth map, (b) RMS (root mean square) error (σ) in computed depth, and (c) the averaged confidence value. Each point (square) in the plots corresponds to one of the strips in the image, and is numbered from left to right (see numbers next to the squares). Note that the gradient of the estimated depth map is nothing but the depth detection gain. Figure 10(a) shows that the gain is invariant except for the left three strips. The slight gain error in the left three strips is because the ratio $G_{M1}(u,v)/G_{P1}(u,v)$ is high for low frequencies. As a result,



(a) far-focused image

(b) near-focused image

(c) gray-coded depth map

(d) wireframe of depth map

Figure 9: Depth from defocus applied to synthetic images of an the inclined plane is accurately recovered despite the significant texture variations.

$G_{P1}(u,v)$ is small in the low frequency region and a small error in $G_{P1}(u,v)$ causes a large error in the ratio. The low values of $G_{P1}(u,v)$ for low-frequency textures is reflected by the extremely low confidence values for the corresponding strips. However, as such low frequencies are cut by the prefilter, depth errors are suppressed if there exist other frequency components. When one wants to utilize low frequencies, a pyramid [Jolion and Rosenfeld-1994, Darrell and Wohn-1988] can be constructed and the rational filters can applied to each level of the pyramid. Depth maps computed at different levels of the pyramid can be combined in a maximum-likelihood sense using confidence measures which are easily computed along with the coefficient image using equation (34). Figure 10(b) and (c) show a rough agreement between the confidence measure plot and the function $1/\sigma^2$.

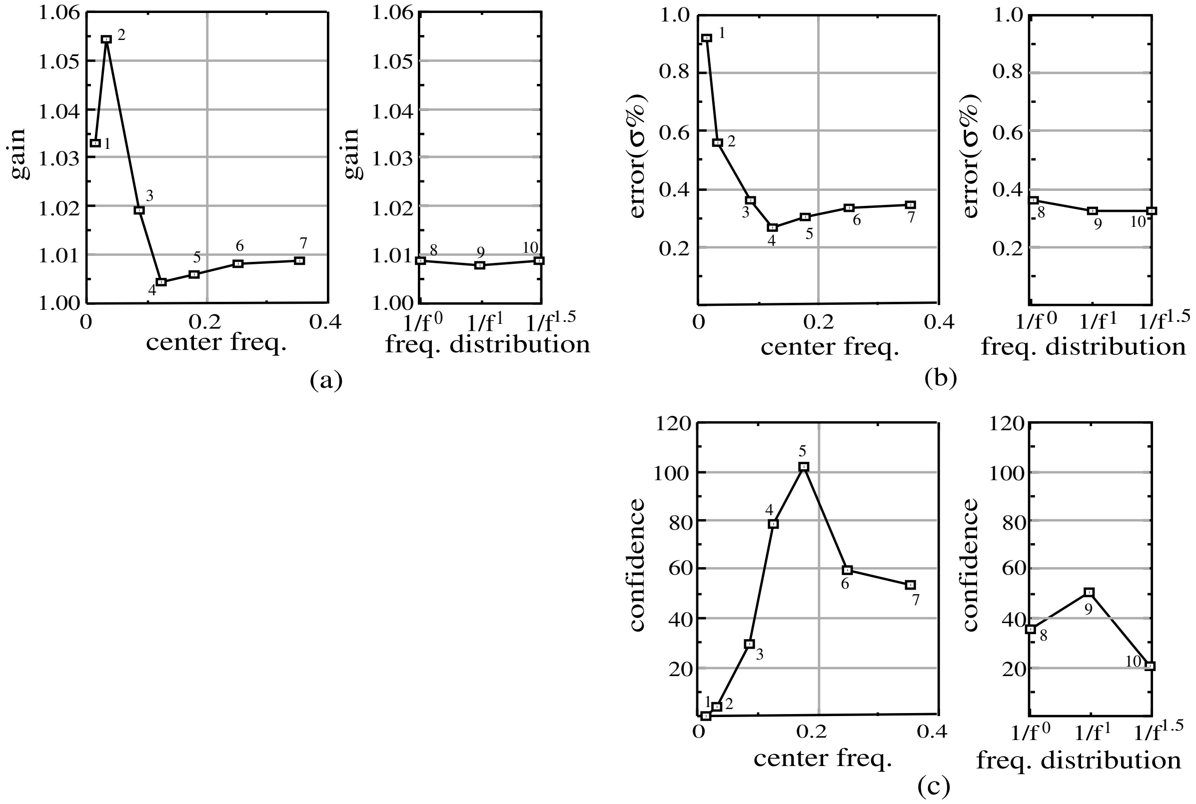
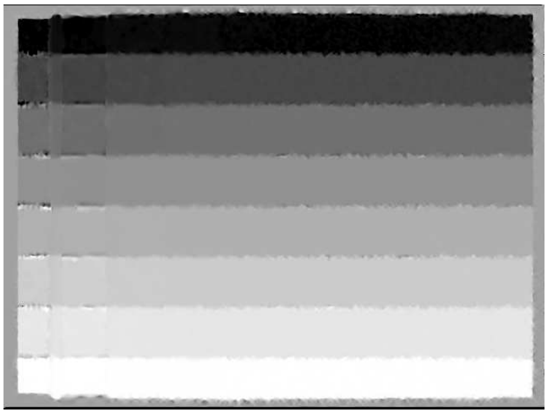
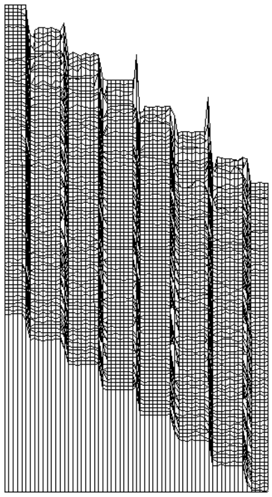


Figure 10: Analysis of depth errors for the textured inclined plane shown in Figure 9. Each point (square) in the plots corresponds to a single texture strip on the inclined plane (numbered 1~10.) (a) The gradient of the computed depth map which corresponds to the depth detection gain. The invariance of depth estimation to image texture is evident. (b) The RMS error (σ) in computed depth. (c) The depth confidence value which is seen to be in rough agreement with $1/\sigma^2$.

In Figure 11, the synthetic images were generated assuming a staircase like three-dimensional structure. The steps of the staircase have textures that are the same as those used in Figure 9. The computed depth map is again very accurate. The depth discontinuities are sensed with sharpness preserved, demonstrating the high spatial resolution of the proposed algorithm. Spikes in the two left strips are again due to extremely low depth confidence values in these areas. In the case of natural textures with enough texture contrast, such low confidence values are unlikely as other frequencies in the texture will provide sufficient information for robust depth estimation.



(a) gray-coded depth map



(a) wire frame plot of the depth map

Figure 11: Depth from defocus applied to synthetic images of a staircase. The textures of the stairs are the same as those of the strips in Figure 9. The depth discontinuities are estimated with high accuracy reflecting high spatial resolution produced by the proposed algorithm.

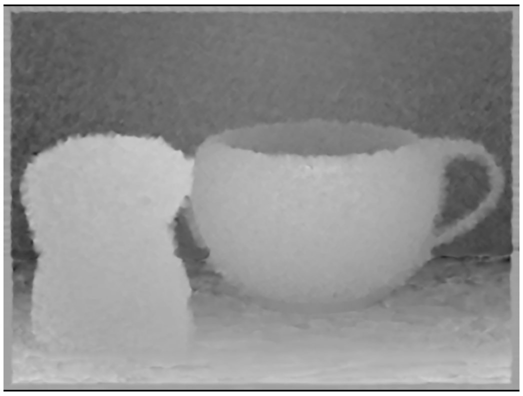
5.2 Experiments on Real Images

Images of real scenes were taken using a SONY XC-77 monochrome camera. The lens used is a Cosmicar B1214D-2 with f=25mm. The lens was converted into a *telecentric lens* by using an additional aperture to make its magnification invariant to defocus (see appendix and [Watanabe and Nayar-1995b]). As a result of telecentricity, image shifts between the far and near focused images are lower than 1/10 of a pixel. The lens aperture was set to F/8.3. The far-focused image i_1 was taken with the lens focused at 869mm from the camera, and the near-focused image i_2 with the lens focused at 529mm. These two distances were chosen so that all scene points lie between them. The above focus settings result in a maximum blur circle radius of $e/F_e = 2.307$ pixels. For each of the two focus settings, 256 images were averaged over 8.5 sec to get images with high signal-to-noise ratio.

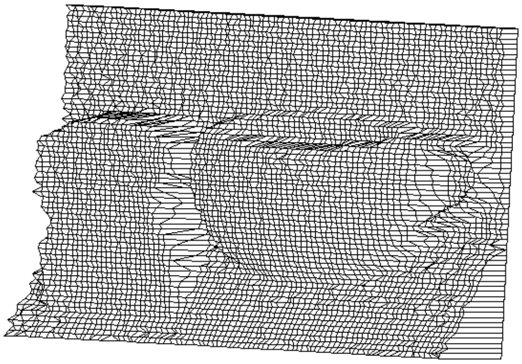
Figure 12 shows results obtained for a scene that includes a variety of textures. Figure 12(a) and (b) are the far-focused and near-focused images, respectively. Figure 12(c) and (d) are the computed depth map and its wireframe plot. Depth maps of all the curved and planar surfaces are detected with high fidelity and high resolution without any post-filtering. After 9×9 median filtering, we get an even better depth map as shown in Figure 12(e).

(a) far-focused image

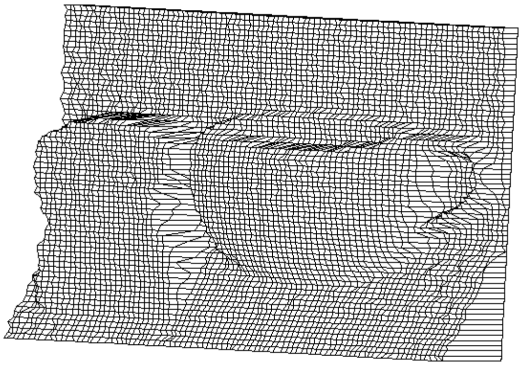
(b) near-focused image



(c) gray-coded depth map without post-filtering



(d) wireframe plot of (c)



(e) wireframe plot after 9×9 median filtering

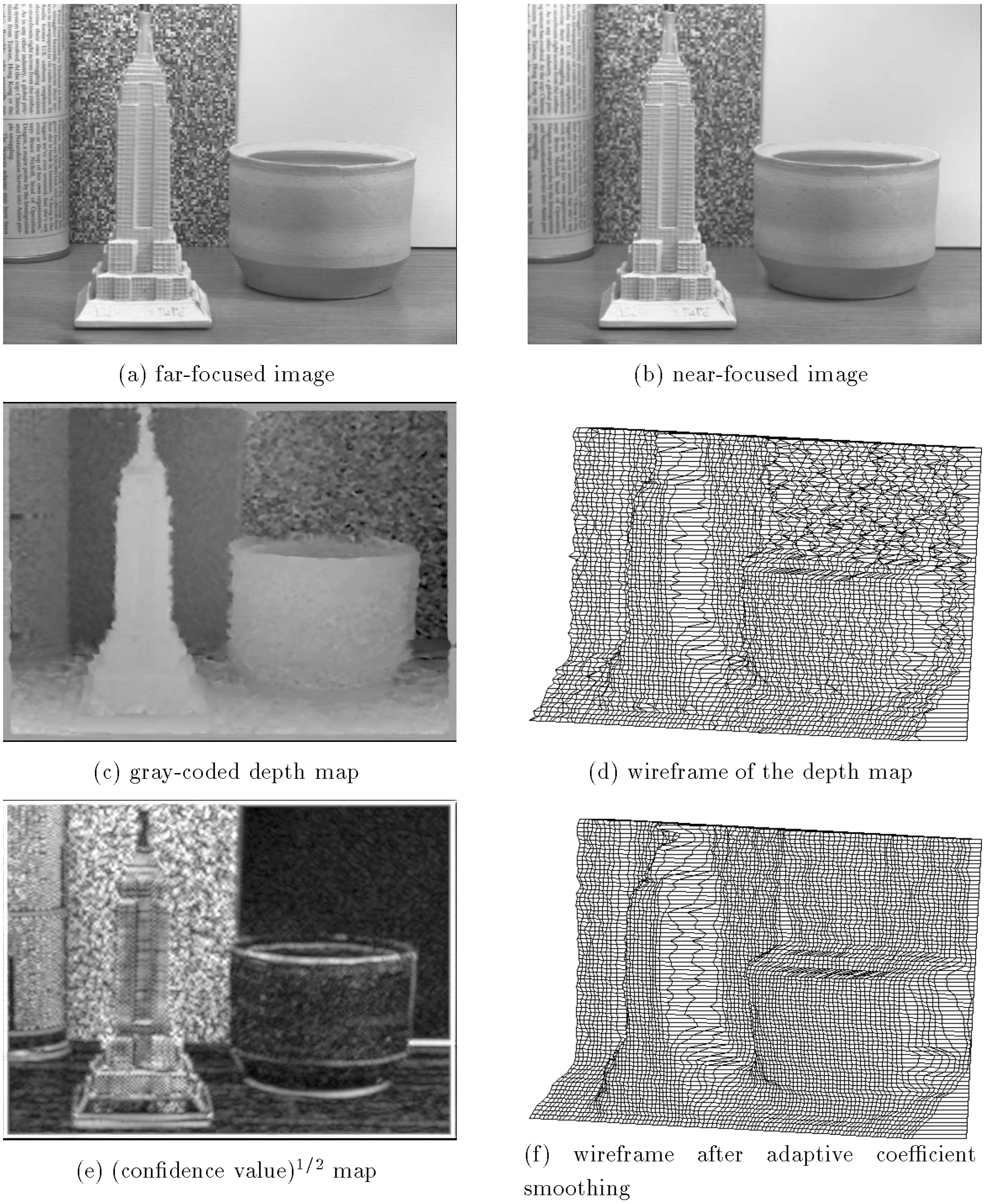
Figure 12: The depth from defocus algorithm applied to a real scene with complex textures.

Figure 13 shows results for a scene which includes areas with extremely weak textures, such as, the white background and the clay cup. Figure 13(a) and (b) are the far-focused and near-focused images, respectively. Figure 13(c) and (d) are the computed depth map and its wireframe plot. All image areas, except the white background area, produce accurate depth estimates. The depth confidence value σ in the textured background is 0.5% of the object distance.[11] The error on the table surface is 1.0% relative to object distance. Even the white background area has a reasonable depth map despite the fact that its texture is very weak. We see that the confidence map in Figure 13(e) reflects this lack of texture. This has motivated us to develop a modified algorithm, called *adaptive coefficient smoothing*, that repeatedly averages the coefficients computed by the rational operators until the confidence value reaches a certain acceptable level. Figure 13(f) shows the depth map computed using this algorithm.

The last experiment seeks to quantify the accuracy of depth estimation. The target used is a plane paper similar to the textured background in the scene in Figure 12. This plane is moved in steps of 25mm and a depth map of the plane is computed for each position. Since the estimated depth β is measured on the image side, it is mapped to the object side using the lens law of equation (2). The optical settings and processing conditions are the same as those used in the previous experiments. The plot in Figure 14 illustrates that the algorithm has excellent depth estimation linearity. The RMS error of a line fit to the measured depths is 4.2 mm. The slight curvature of the plot is probably due to errors in optical settings, such as, focal length and aperture.

Depth values for a 50×50 area were used to estimate the RMS depth error for each position of the planar surface. In Figure 14 the RMS errors are plotted as $\pm\sigma$ error bars. The RMS error relative to object distance is seen to vary with object distance. It is $0.4\% \sim 0.8\%$ for close objects and $0.8\% \sim 1.2\%$ for objects farther than 880 mm. This is partly because of the mapping from the depth measured on the image side to depth on the object side. The other reason is that the error in estimated depth σ_β is larger for a scene point with larger β, as seen from equation (34). Note that this RMS error depends on the coefficient smoothing and post-filtering stages. We found empirically that the error has a Gaussian-like distribution. Using this distribution, one can show that the error reduces by a factor of 1/8 if the depth map is convolved with an 8×8 averaging filter.

[11] This definition of error is often used to quantify the performance of range sensors.



(a) far-focused image

(b) near-focused image

(c) gray-coded depth map

(d) wireframe of the depth map

(e) (confidence value)$^{1/2}$ map

(f) wireframe after adaptive coefficient smoothing

Figure 13: Depth from defocus applied to a scene that includes very weak texture (white background). The larger errors in the region of weak texture is reflected by the confidence map. An adaptive coefficient smoothing algorithm uses the confidence map to refine depth estimates in regions with weak texture.

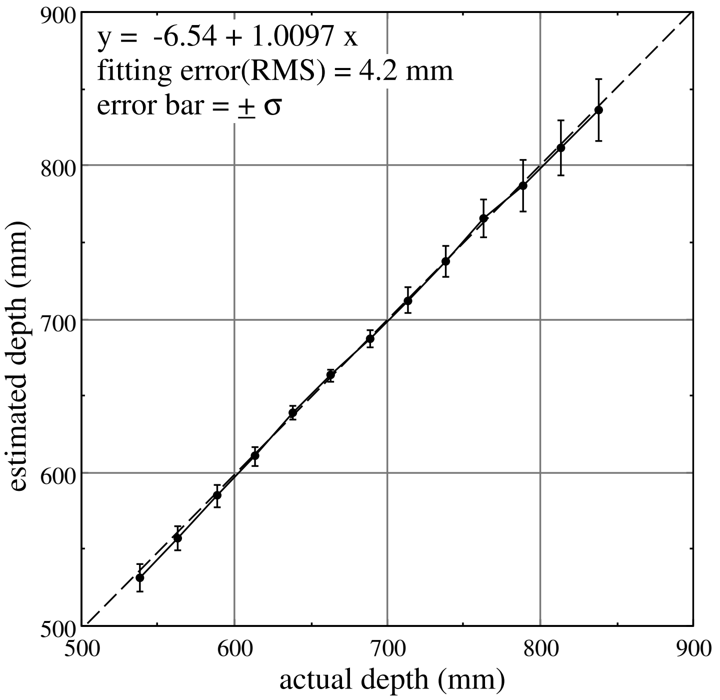


Figure 14: Depth estimation linearity for a textured plane. The plane is moved in increments of 25mm, away from the lens. All plotted distances are measured from the lens. The RMS error relative to object distance is 0.4% $\sim$ 1.2%.

6 Conclusions

We proposed the class of rational operators for passive depth from defocus. Though the operators are broadband, when used together, they provide invariance to scene texture. Since they are broadband, a small number of operators are sufficient to cover the entire frequency spectrum. Hence, rational operators can replace large filter banks that are expensive from a computational perspective. This advantage comes without the need to sacrifice depth estimation accuracy and resolution. We have detailed the procedure used to design rational operators. As an example, we constructed 7$\times$7 operators using a polynomial model for the normalized image ratio. However, the notion of rational operators is more general and represents a complete class of filters. The design procedure described here can be used to construct operators based on other rational models for the normalized image ratio. Further, rational operators can be derived for any desired blur function.

In addition to the rational operators, we discussed a wide range of issues that are pertinent to depth from defocus. In particular, detailed analyses and techniques were provided for prefiltering near and far focused images as well as post-processing the outputs of the rational operators. The operator outputs have also been used to derive a depth confidence measure. This measure can be used to enhance computed depth maps. The proposed depth from defocus algorithm requires only a total of 5 convolutions. We tested the algorithm using both synthetic scenes and real scenes to evaluate performance. We found the depth detection gain error to be less than 1%, regardless of texture frequency. Depth accuracy was found to be $0.5 \sim 1.2\%$ of object distance from the sensor.

These results have several natural extensions. (a) Since some scene areas are expect to have very low texture frequency, it would be meaningful to embed the proposed scheme in a pyramid-based processing framework. Image areas with dominant low frequencies will have higher frequencies at higher levels of the pyramid. The proposed algorithm can be applied to all levels of the pyramid and the resulting depth maps can be combined using the depth confidence measures. (b) Given the efficiency of the algorithm, it is worth implementing a real-time version using a pipeline image processing architecture such as the Datacube MV200. We estimate that such an algorithm would result in at least 6 depth maps per second of 512×480 resolution. (c) In our present implementation, we have varied the position of the image sensor to change the focus setting. Alternatively, the aperture size can be varied. Rational operators can be derived for such an optical setup using the basis functions $b_{P1}(\alpha) = \alpha^2$, $b_{P2}(\alpha) = \alpha^4$ and $b_{M1}(\alpha) = 1$ (see [Watanabe and Nayar-1995a]). (d) Finally, it would be worthwhile applying the algorithm to outdoor scenes with large structures.

7 Appendix

7.1 Problem of Image Registration

For the rational operators to give accurate results, the far-focused image i_1 and near-focused image i_2 need to be precisely registered (within 0.1 pixel) with respect to one another. However, in most conventional lenses, magnification varies with focus setting and hence misregistration is introduced. Further, in our experiments, we have mechanically changed the focus setting and, in the process, introduced some translation between the two images. If the lens aberrations are small, the misregistration is decomposed into two factors — a global magnification change and a global translation. Of the two factors, magnification change proves much more harmful. This change can be corrected using image warping techniques [Darrell and Wohn-1988, Wolberg-1990]. However, this generally introduces undesirable effects such as smoothing and aliasing since warping is based on spatial interpolation and resampling techniques. We have used an optical solution to the problem that is described in the following section and detailed in [Watanabe and Nayar-1995b].

7.1.1 Telecentric Optics

In the imaging system shown in Figure 1, the effective image location of point P moves along the *principal ray* R as the sensor plane is displaced. This causes a shift in image coordinates of the image of P. This variation in image magnification with defocus manifests as a correspondence like problem in depth from defocus, as corresponding points in images i_1 and i_2 are needed to estimate blurring.

We approach the problem from an optical perspective rather than a computational one. Consider the image formation model shown in Figure 15. The only modification made with respect to the model in Figure 1 is the use of the external aperture A'. The aperture is placed at the *front-focal plane*, i.e. a focal length in front of the *principal point* O of the lens. This simple addition solves the problem of magnification variation with distance α of the sensor plane from the lens. Simple geometrical analysis reveals that a ray of light R' from any scene point that passes through the center O' of aperture A' emerges parallel to the optical axis on the image side of the lens [Kingslake-1983]. As a result, despite blurring, the effective image coordinates of point P in both images i_1 and i_2 are the same as the coordinate of its focused image Q on i_f. Given an off-the-shelf lens, such an aperture is easily appended to the casing of the lens. The resulting optical system is called a *telecentric lens*. While the nominal and effective *F-numbers* of the classical optics in Figure 1 are f/a and d_i/a, respectively, they are both equal to f/a' in the telecentric case. The magnification change can be reduced to an order of less than 0.03%, i.e. 0.1 pixel for a 640×480 image. A detailed discussion on telecentricity and its implementation can be found in [Watanabe and Nayar-1995b]. We recently used this idea to develop a real-time active depth from defocus sensor [Nayar *et al.*-1995, Watanabe *et al.*-1995].

7.1.2 Translation Correction

We have seen in the previous section how magnification changes between the far-focused and near-focused images can be avoided. When the focus setting is changed, translations may also be introduced. Translation correction can be done using image processing without introducing any harmful image artifacts. However, the processing must be carefully

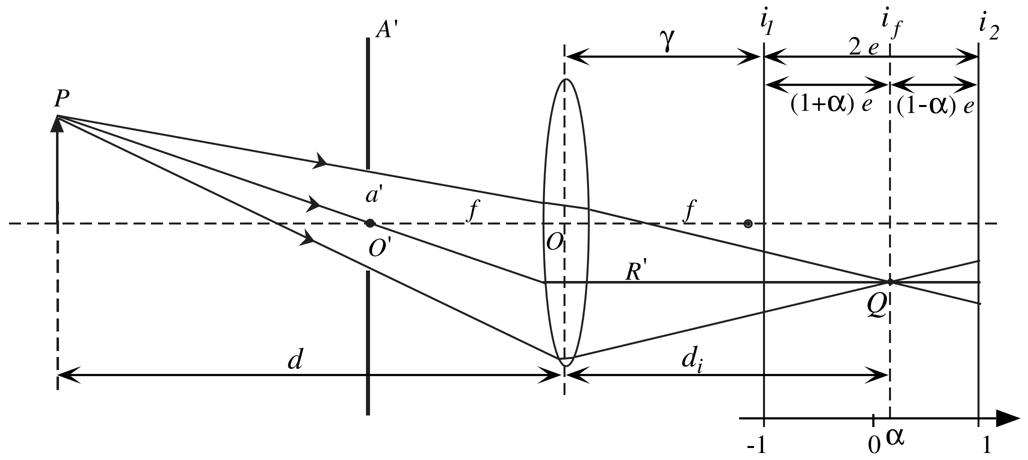


Figure 15: A constant-magnification imaging system for depth from defocus is achieved by simply placing an aperture at the front-focal plane of the optics. The resulting telecentric optics avoids the need for registering the far-focused and near-focused images [Watanabe and Nayar-1995b].

implemented since we seek 0.1 pixel registration between the two images. The procedure we use is briefly described here and is detailed in [Watanabe and Nayar-1995b]. We use FFT-phase based local shift detection to estimate shift vectors with sub-pixel accuracy. We divide the Fourier spectra of corresponding local areas of the two images. Then we fit a plane to the phases of the ratio of the spectra. The gradient of the fitted plane is nothing but the relative shift between the two images. Once we get shift vectors at several positions in the image, similarity transform is used to model the shift vector field. By fitting the vectors to the similarity model, we can estimate the global translation and any residual magnification changes, separately [Watanabe and Nayar-1995b]. The residual magnification is corrected by tuning the aperture position of the telecentric optics. The translation is corrected by shifting both images in opposite directions. As we need sub-pixel accuracy, we interpolate the image and resample it to generate the registered images. The interpolating function is the Lanczos4 windowed sinc function [Wolberg-1990]. Since the translation correction remains constant over the entire image, a single shift invariant convolution achieves the desired shift. Though this convolution distorts the image spectrum, since both images undergo the same amount of shift, the distortion is the same for both images. This common distortion is eliminated when the normalized image ratio M/P is computed before the application of the rational filters. After the above translation correction, we found the maximum registration error in our experiments to be as small as 0.02 pixels.

7.2 Operator Response and Depth Error

The deviation of the ratio functions $\mathcal{G}_{Pi}(u,v)$ or $\mathcal{G}_{Mi}(u,v)$ after filter design, to those obtained from fitting the polynomial model to the normalized image ratio, varies with frequency (u,v), and hence depends on the texture of the scene. For the filter design described in section 4.1, we need a relation between the above ratio error and the depth estimation error. Starting with equation (12), we get:

$$M(u,v;\alpha)\frac{G_{M1}(u,v)}{G_{P1}(u,v)} = P(u,v;\alpha)\beta_f(u,v) + P(u,v;\alpha)\frac{G_{P2}(u,v)}{G_{P1}(u,v)}\beta_f(u,v)^3 . \qquad (39)$$

Here, $\beta_f(u,v)$ is the depth estimated at a single frequency (u,v). Since $\mu(u,v)$ in the ratio condition (13) has not been fixed, we can define $G_{P1}(u,v) = \mu(u,v)$, $G_{M1}(u,v) = \mathcal{G}_{M1}(u,v)\mu(u,v)$ and $G_{P2}(u,v) = \mathcal{G}_{P2}(u,v)\mu(u,v)$. Then, equation (39) becomes:

$$M(u,v;\alpha)\,\mathcal{G}_{M1}(u,v) = P(u,v;\alpha)\,\beta_f(u,v) + P(u,v;\alpha)\,\mathcal{G}_{P2}(u,v)\,\beta_f(u,v)^3 . \qquad (40)$$

By differentiation we get:

$$\begin{aligned} M(u,v;\alpha)\,d\mathcal{G}_{M1}(u,v) &= P(u,v;\alpha)\,d\beta_f(u,v) + P(u,v;\alpha)\,\beta_f(u,v)^3\,d\mathcal{G}_{P2}(u,v) \\ &\quad + 3\,P(u,v;\alpha)\,\mathcal{G}_{P2}(u,v)\,\beta_f(u,v)^2\,d\beta_f(u,v), \end{aligned} \qquad (41)$$

where, $M(u,v;\alpha)$ and $P(u,v;\alpha)$ can be treated as constants since we wish to find the error in $\beta_f(u,v)$ caused by errors in $\mathcal{G}_{M1}(u,v)$ and $\mathcal{G}_{P2}(u,v)$. Solving for $d\beta_f(u,v)$, we get:

$$d\beta_f(u,v) = \frac{M(u,v;\alpha)\,d\mathcal{G}_{M1}(u,v) - P(u,v;\alpha)\,\beta_f(u,v)^3\,d\mathcal{G}_{P2}(u,v)}{P(u,v;\alpha)(1 + 3\,\mathcal{G}_{P2}(u,v)\,\beta_f(u,v)^2)} \qquad (42)$$

Since $\mathcal{G}_{P2}(u,v)$ is a small correction factor, it can be approximated by:

$$\begin{aligned} d\beta_f(u,v) &\simeq \frac{M(u,v;\alpha)d\mathcal{G}_{M1}(u,v) - P(u,v;\alpha)\beta_f(u,v)^3 d\mathcal{G}_{P2}(u,v)}{P(u,v;\alpha)} \\ &= \frac{M(u,v;\alpha)}{P(u,v;\alpha)}d\mathcal{G}_{M1}(u,v) - \beta_f(u,v)^3 d\mathcal{G}_{P2}(u,v) \\ &\simeq \frac{G_{P1}(u,v;\alpha)\beta_f(u,v)}{G_{M1}(u,v;\alpha)}d\mathcal{G}_{M1}(u,v) - \beta_f(u,v)^3 d\mathcal{G}_{P2}(u,v) \\ &= \frac{\beta_f(u,v)}{\mathcal{G}_{M1}(u,v)}d\mathcal{G}_{M1}(u,v) - \beta_f(u,v)^3 d\mathcal{G}_{P2}(u,v). \end{aligned} \qquad (43)$$

From equations (23) and (20), since $c_{P2} \ll c_{P1}$ (c_{P2} represents a small correction), the depth estimate β can be approximated by integrating over all frequencies:

$$\begin{aligned}
\beta &\simeq \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} M(u,v;\alpha)\, G_{M1}(u,v)\, du\, dv}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(u,v;\alpha)\ G_{P1}(u,v)\ du\, dv} \\
&= \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(u,v;\alpha) G_{P1}(u,v) \frac{M(u,v;\alpha) G_{M1}(u,v)}{P(u,v;\alpha) G_{P1}(u,v)}\, du\, dv}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(u,v;\alpha) G_{P1}(u,v)\, du\, dv} \\
&\simeq \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(u,v;\alpha) G_{P1}(u,v) \beta_f(u,v)\, du\, dv}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(u,v;\alpha) G_{P1}(u,v)\, du\, dv}.
\end{aligned} \tag{44}$$

Hence, the error in β caused by the error in $\beta_f(u,v)$ is:

$$d\beta \simeq \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(u,v;\alpha) G_{P1}(u,v) d\beta_f(u,v)\, du\, dv}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(u,v;\alpha) G_{P1}(u,v)\, du\, dv}, \tag{45}$$

Combining this expression with equation (43), we have:

$$d\beta \simeq \frac{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Big(\frac{P(u,v;\alpha) G_{P1}(u,v) \beta_f(u,v)}{\mathcal{G}_{M1}(u,v)} d\mathcal{G}_{M1}(u,v) - P(u,v;\alpha) G_{P1}(u,v) \beta_f(u,v)^3 d\mathcal{G}_{P2}(u,v) \Big)\, du\, dv}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(u,v;\alpha) G_{P1}(u,v)\, du\, dv}. \tag{46}$$

What are the optimal values of $d\mathcal{G}_{M1}(u,v)$ and $d\mathcal{G}_{P2}(u,v)$ that would minimize the depth error $d\beta$? This question is not trivial as $d\mathcal{G}_{M1}(u,v)$ and $d\mathcal{G}_{P2}(u,v)$ influence each other in a complex way. To avoid either of the two terms in the integrand in the numerator from taking on a disproportionately large value, we have decided to assume both terms to be constant of value κ. This gives us the following bounds on $d\mathcal{G}_{M1}(u,v)$ and $d\mathcal{G}_{P2}(u,v)$:

$$\begin{aligned}
\sigma_{\mathcal{G}_{M1}}(u,v) &= \kappa \frac{\mathcal{G}_{M1}(u,v)}{P(u,v;\alpha)\, G_{P1}(u,v)} \\
\sigma_{\mathcal{G}_{P2}}(u,v) &= \kappa \frac{1}{P(u,v;\alpha) G_{P1}(u,v)},
\end{aligned} \tag{47}$$

where, the $|\beta_f(u,v)|$ was set to 1 as this represents the worst case, i.e. largest normalized depth error.

Acknowledgements

This research was conducted at the Center for Research in Intelligent Systems, Department of Computer Science, Columbia University. It was supported in part by the Production Engineering Research Laboratory, Hitachi, and in part by the David and Lucile Packard Fellowship. The authors thank Yasuo Nakagawa of Hitachi Ltd. for his support and encouragement of this work.

References

[Besl, 1988] P. J. Besl. Range imaging sensors. Technical Report GMR-6090, General Motors Research Laboratories, March 1988.

[Born and Wolf, 1965] M. Born and E. Wolf. *Principles of Optics.* London:Permagon, 1965.

[Bove, Jr., 1993] V. M. Bove, Jr. Entropy-based depth from focus. *Journal of Optical Society of America A*, 10:561–566, April 1993.

[Bracewell, 1965] R. N. Bracewell. *The Fourier Transform and Its Applications.* McGraw Hill, 1965.

[Burt and Adelson, 1983] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Trans. on Communications*, COM-31(4):532–540, April 1983.

[Darrell and Wohn, 1988] T. Darrell and K. Wohn. Pyramid based depth from focus. *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 504–509, June 1988.

[Ens and Lawrence, 1991] J. Ens and P. Lawrence. A matrix based method for determining depth from focus. *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 600–609, June 1991.

[Gokstorp, 1994] M. Gokstorp. Computing depth from out-of-focus blur using a local frequency representation. *Proc. on Intl. Conf. on Patt. Recog.*, October 1994.

[Hoel, 1971] P. G. Hoel. *Introduction to Mathematical Statistics.* John Wiley & Sons, New York, 1971.

[Horn, 1968] B. K. P. Horn. Focusing. Memo 160, AI Lab., Massachusetts Institute of Technology, Cambridge, MA, USA, 1968.

[Horn, 1986] B. K. P. Horn. *Robot Vision.* The MIT Press, 1986.

[Jarvis, 1983] R. A. Jarvis. A perspective on range finding techniques for computer vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 5(2):122–139, March 1983.

[Jolion and Rosenfeld, 1994] J. M. Jolion and A. Rosenfeld. *A Pyramid Framework for Early Vision.* Kluwer Academic Publishers, Boston, MA, 1994.

[Kingslake, 1983] R. Kingslake. *Optical System Design.* Academic Press, 1983.

[Krotkov, 1987] E. Krotkov. Focusing. *Intl. Journal of Computer Vision*, 1:223–237, 1987.

[Nayar and Nakagawa, 1994] S. K. Nayar and Y. Nakagawa. Shape from focus: An effective approach for rough surfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(8):824–831, August 1994.

[Nayar *et al.*, 1995] S. K. Nayar, M. Watanabe and M. Noguchi. Real-time focus range sensor. *Proc. of Intl. Conf. on Computer Vision*, pages 995–1001, June 1995.

[Oppenheim and Schafer, 1989] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing.* Prentice Hall, Englewood Cliffs, NJ, 1989.

[Peitgen and Saupe, 1988] H. O. Peitgen and D. Saupe, editors. *The Science of Fractal Images.* Springer-Verlag, New York, NY, 1988.

[Pentland, 1987] A. Pentland. A new sense for depth of field. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(4):523–531, July 1987.

[Press *et al.*, 1992] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C.* Cambridge University Press, 1992.

[Subbarao and Surya, 1994] M. Subbarao and G. Surya. Depth from defocus: A spatial domain approach. *International Journal of Computer Vision*, 13(3):271–294, 1994.

[Subbarao, 1988] M. Subbarao. Parallel depth recovery by changing camera parameters. *Proc. of Intl. Conf. on Computer Vision*, pages 149–155, December 1988.

[Watanabe and Nayar, 1995a] M. Watanabe and S. K. Nayar. Minimal operator set for texture invariant depth from defocus. Technical Report CUCS-031-95, Dept. of Computer Science, Columbia University, New York, NY, USA, October 1995.

[Watanabe and Nayar, 1995b] M. Watanabe and S. K. Nayar. Telecentric optics for constant-magnification imaging. Technical Report CUCS-026-95, Dept. of Computer Science, Columbia University, New York, NY, USA, September 1995.

[Watanabe *et al.*, 1995] M. Watanabe, S. K. Nayar and M. Noguchi. Real-time computation of depth from defocus. *Proc. of SPIE: Three-Dimensional and Unconventional Imaging for Industrial Inspection and Metrology*, 2599:A–03, November 1995.

[Wolberg, 1990] G. Wolberg. *Digital Image Warping.* IEEE Compter Society Press, Los Alamitos, CA, 1990.

[Xiong and Shafer, 1993] Y. Xiong and S. A. Shafer. Depth from focusing and defocusing. *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 68–73, June 1993. Also, Technical Report CMU-RI-TR-93-07, Pittsburgh, PA, USA, March, 1993.

[Xiong and Shafer, 1995] Y. Xiong and S. A. Shafer. Moment filters for high precision computation of focus and stereo. *Proc. of Intl. Conf. on Robotics and Automation*, pages 108–113, August 1995. Also, Technical Report CMU-RI-TR-94-28, Pittsburgh, PA, USA, September, 1994.