Determining Distance from Defocused Images of Simple Objects

Muralidhara Subbarao

Department of Electrical Engineering
State University of New York at Stony Brook
Stony Brook, NY 11794-2350, USA

Abstract

New methods for determining the distance of a simple class of objects from a camera system are presented. The methods use the defocus or blur information in the images formed by an optical system such as a convex lens. Paraxial geometric optics forms the theoretical basis of these methods. The class of objects includes bright points, lines, step edges, blobs, stripes, and smooth edges. Only one defocused image is used. The methods are general in the sense that no restrictions are imposed on the form of the point spread function of the camera system. Computational methods are presented for determining the distance of objects, focal length of the camera system, and the size of the camera's aperture. Methods are also presented for finding the point spread function, line spread function, and the edge spread function of the camera system. The methods for determining distance have been implemented and verified experimentally. The experimental results suggest that useful depth information can be obtained from defocus information. Both experimental and theoretical error analyses are presented.

1. Introduction

The image of a scene formed by an optical system such as a convex lens contains information about the distance (or depth) of objects in the scene. Objects at a particular distance are focused whereas other objects are defocused or blurred by different degrees depending on their distance. The depth information inherent in defocus has recently drawn the attention of some researchers (Pentland, '87, '89; Grossman, 1987; Subbarao, 1987a,b,c, 1988, 1989; Subbarao and Natarajan, 1988; T. Hwang, J. J. Clark, and A. L. Yuille, 1989). For a camera with a lens of focal length f the relation between the position of a point close to the optical axis in the scene and the position of its focused image is given by the well known lens formula

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v} \,, \tag{1.1}$$

where u is the distance of the object, and v is the distance of the image (see Figure 1). Therefore, in the image formed by a camera, only objects at a certain distance are in focus; other objects are blurred by varying degrees depending on their distance. The distance of an object that is in focus can be recovered easily using the lens formula (e.g.: Horn, 1968; Tenenbaum, 1970; Jarvis, 1983; Schlag *et al*, 1983; Krotkov, 1986). Here we consider the more general problem of *finding the distance of an object that may or may not be in focus.*

2. Previous work

Pentland (1987) and Grossman (1987) both addressed the problem of recovering depth from blurred edges. Pentland modeled a blurred edge as the result of convolving a focused image with a *Gaussian point spread function.* He showed that if $C(x,y)$ is the laplacian of the observed image then the spread σ of the Gaussian is related to $C(x,y)$ by

$$\ln\left(\frac{b}{\sqrt{2\pi}\,\sigma^3}\right) - \frac{x^2}{2\sigma^2} = \ln\left|\frac{C(x,y)}{x}\right| \,, \tag{2.1}$$

where b is the magnitude (or ''height'') of the step edge and x,y the image coordinate system with its origin on the edge and x-axis perpendicular to the edge. In the above equation b and σ are the only unknowns. He solved for these unknowns by formulating the above equation as a linear regression in x^2. The depth was then computed from σ. Pentland applied his method to an image of a natural scene and showed that depth of edges could be classified as being small, medium, or large.

Grossman (1987) showed experimentally that useful depth information can be obtained from blurred edges. He considers, in addition to step-edges, edges of finite width (e.g.: ramp edge). Grossman however does not provide a theoretical justification for his method.

Subbarao and Gurumoorthy (1987) presented a new approach for finding the distance of step edges. The computational method of this approach was much simpler than that of Pentland's. The solution for the edge height b and the spread σ in equation (2.1) were given in closed-form. More importantly, the new method assumed only that the point spread function of the camera was rotationally symmetric. It was not restricted to the case where the point spread function was a Gaussian as in Pentland's method. The method was verified experimentally and shown to provide useful depth information.

3. New developments

In this paper we develop a formal theoretical frame-work for determining distance. The frame-work is based on paraxial geometric optics (cf. Hecht and Zajac, 1979; Born and Wolf, 1975; Gaskill, 1978). While the previous work deals only with step edges, here we present methods for finding the distance of a larger class of objects. The objects include points, blobs (e.g. spot patterns of finite size), lines, stripes (elongated bar-like patterns of finite width), step edges, and smoothly rising edges (e.g. ramp edges). Further, no restrictions are imposed on the form of the point spread function of the camera system. In particular, the point spread function is not restricted to be of a Gaussian form or a rotationally symmetric form, as is the case in previous work. In addition to describing computational methods for determining the distance of the objects, this paper also suggests methods for finding certain camera parameters such as the focal length and the aperture size of the camera system. Methods are also presented for finding the point spread function, line spread function, and the edge spread function of the camera system.

The results of experiments carried-out using an actual camera system are presented. These results verify our theoretical frame-work and the computational approach. This paper includes both experimental and theoretical error analyses. Such analyses are not found in the previous work that we know of.

4. Example

The basic framework of our approach can be illustrated with the example of the method of determining the distance of an object having a brightness step edge (see Figure

2). The defocused edge image of the object is modeled as the result of convolving the focused image of the object with a point spread function (Figure 2a). The point spread function depends both on the camera parameters and the distance of the object from the camera.

The derivative of the defocused image is computed along a direction perpendicular to the edge (i.e. along the gradient direction). The height of the edge and line spread function of the camera are computed from this derivative image (Figure 2b). The standard deviation (σ_{l0}) of the distribution of the line spread function is then computed (Figure 2c). This quantity is called the *spread parameter* of the line spread function. It is a measure of image blur or defocus. The distance of the object is then computed from this spread parameter (Figure 2d). This last step requires the knowledge of some camera constants which can be determined by measuring certain camera parameters. The camera parameters are: size and shape of the camera's aperture, focal length, and distance between the lens and the image detector. The camera constants can also be determined experimentally through a calibration procedure.

The calibration procedure involves the following steps. The edge image of the object is recorded for several known distances of the object from the camera system. For each image the spread parameter of the line spread function is computed. The camera constants are then determined from a knowledge of both the distance and the corresponding spread parameter.

Methods for determining the distance of other objects such as points, blobs, lines, and stripes are similar to the above method.

5. Notation, Definitions, and Preliminaries

In this section we will describe the notation used in the remaining sections and define terms and present results for subsequent usage.

5.1 Functions of one variable

Let ψ be any function of one variable, say x.

The zeroth moment of ψ is denoted by A_ψ and is defined as

$$A_\psi = \int_{-\infty}^{\infty} \psi(x)\, dx \; . \tag{5.1.1}$$

A_ψ defined above can be interpreted as the area under the curve $y = \psi(x)$ plotted on a graph sheet. It can also be interpreted as the mass of a rod of infinite length whose density as a function of position is given by $\psi(x)$.

The first normalized moment of ψ is denoted by $\bar{x}_\psi$ and is defined as

$$\bar{x}_\psi = \frac{1}{A_\psi} \int_{-\infty}^{\infty} x\, \psi(x)\, dx \; . \tag{5.1.2}$$

$\bar{x}_\psi$ defined above can be interpreted as the mean of the distribution of the function ψ. It can also be interpreted as the center of mass of a rod of infinite length whose density distribution is given by $\psi(x)$.

The second central moment of ψ is denoted by σ_ψ^2 and is defined as

$$\sigma_\psi^2 = \frac{1}{A_\psi} \int_{-\infty}^{\infty} (x - \bar{x}_\psi)^2\, \psi(x)\, dx \; . \tag{5.1.3}$$

σ_ψ as defined above can be interpreted as the standard deviation of the distribution of the function ψ. It can also be interpreted as the radius of gyration of a rod of infinite length about its center of mass whose density distribution is given by $\psi(x)$.

Fourier transforms of functions will be denoted by the respective capital letters.

The Fourier transform of ψ is denoted by Ψ and is defined as

$$\Psi(\omega) = \int_{-\infty}^{\infty} \psi(x)\, e^{-j\omega x}\, dx \; , \tag{5.1.4}$$

where $j = \sqrt{-1}$ and ω is the spatial Fourier frequency expressed in radians per unit distance.

The first derivative of $\Psi(\omega)$ with respect to ω is denoted by $\Psi'(\omega)$. It can be shown that

$$\Psi'(\omega) = -j \int_{-\infty}^{\infty} x\, \psi(x)\, e^{-j\omega x}\, dx \; . \tag{5.1.5}$$

The second derivative of $\Psi(\omega)$ with respect to ω is denoted by $\Psi''(\omega)$. It can be shown that

$$\Psi''(\omega) = -\int_{-\infty}^{\infty} x^2\, \psi(x)\, e^{-j\omega x}\, dx\ . \tag{5.1.6}$$

The three theorems below follow from equations (5.1.4), (5.1.5), and (5.1.6).

Theorem 5.1.1a: $A_\psi = \Psi(0)$ ●

Theorem 5.1.1b: If A_ψ=1 then $\bar{x}_\psi = j\ \Psi'(0)$ ●

Theorem 5.1.1c: If A_ψ=1, and $\bar{x}_\psi$=0, then $\Psi''(0) = -\sigma_\psi^2$ ●

Let $h(x)$, $w(x)$, and $z(x)$ be three functions. For these three functions, we define the quantities A_h, A_w, A_z, $\bar{x}_h$, $\bar{x}_w$, $\bar{x}_z$, σ_h, σ_w, σz, $H(\omega)$, $W(\omega)$, $Z(\omega)$, $H'(\omega)$, $W'(\omega)$, $Z'(\omega)$, and $H''(\omega)$, $W''(\omega)$, $Z''(\omega)$, as we have done above for function ψ. Now we state and prove an important theorem which will be used in determining the distance of defocused stripes and smoothly rising edges.

Theorem 5.1.2: Let $z(x) = h(x) * w(x)$, where '*' denotes convolution operation. Also, let A_h=1, A_w=1, $\bar{x}_h$=0, and $\bar{x}_w = 0$. Then the following results hold: A_z=1, $\bar{x}_z = 0$, and $\sigma_z^2 = \sigma_h^2 + \sigma_w^2$.

Proof: Convolution in the spatial domain is equivalent to multiplication in the Fourier domain. Therefore

$$Z(\omega) = H(\omega)\ W(\omega)\ . \tag{5.1.7}$$

Consequently,

$$A_z = Z(0) = H(0)\ W(0) = A_h \cdot A_w = 1 \cdot 1 = 1\ . \tag{5.1.8}$$

Similarly,

$$\bar{x}_z = Z'(0) = H(0)\ W'(0) + H'(0)\ W(0) = A_h \bar{x}_w + \bar{x}_h A_w = 0\ , \tag{5.1.9}$$

and

$$\sigma_z^2 = Z''(0) = H(0)\, W''(0) \;+ 2H'(0)W'(0) + H''(0)\, W(0) \qquad \textbf{(5.1.10)}$$

$$= A_h \sigma_w^2 + 2\overline{x}_h \overline{x}_w + \sigma_h^2 A_w$$

$$= \sigma_w^2 + \sigma_h^2 \;\bullet$$

The above theorem implies that, under some weak conditions, the variance of the distribution of the convolution of two functions is equal to the sum of the variances of the distributions of the two individual functions. We suspect that the above result is not new. It is probably known in probability theory and statistics. (It is known in probability theory that the probability density function of the summation of two independent zero-mean random variables is given by the convolution of the probability density functions of the two individual random variables.) A similar result holds for functions of two variables. It is stated in the next section.

5.2 Functions of two variables

We now define a set of terms and present results for functions of two variables that are analogous to those above for the case of functions of one variable.

Let γ be any function of two variables, say x and y.

The zeroth moment of γ is denoted by A_γ and is defined as

$$A_\gamma = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \gamma(x,y)\; dx\; dy. \qquad (5.2.1)$$

A_γ defined above can be interpreted as the volume under the surface $z = \gamma(x,y)$. It can also be interpreted as the mass of a plate of infinite area whose density as a function of position is given by $\gamma(x,y)$.

The first normalized moment of γ about the $y-$axis is denoted by $\overline{x}_\gamma$ and is defined as

$$\overline{x}_\gamma = \frac{1}{A_\gamma} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x\; \gamma(x,y)\; dx\; dy. \qquad (5.2.2)$$

The first normalized moment of γ about the x −axis is denoted by $\bar{y}_\gamma$ and is defined as

$$\bar{y}_\gamma = \frac{1}{A_\gamma} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y \, \gamma(x,y) \, dx \, dy. \tag{5.2.3}$$

The point $(\bar{x}_\gamma, \bar{y}_\gamma)$ as defined above can be interpreted as the center of mass of a plate of infinite area whose density distribution is given by $\gamma(x,y)$.

The second central moment of γ parallel to the y −axis is denoted by $\sigma^2_{x\gamma}$ and is defined as

$$\sigma^2_{x\gamma} = \frac{1}{A_\gamma} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x}_\gamma)^2 \, \gamma(x,y) \, dx \, dy. \tag{5.2.4}$$

The second central moment of γ parallel to the x −axis is denoted by $\sigma^2_{y\gamma}$ and is defined as

$$\sigma^2_{y\gamma} = \frac{1}{A_\gamma} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (y - \bar{y}_\gamma)^2 \, \gamma(x,y) \, dx \, dy. \tag{5.2.5}$$

The second central moment of γ is denoted by σ^2_γ and is defined as

$$\sigma^2_\gamma = \frac{1}{A_\gamma} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} [(x - \bar{x}_\gamma)^2 + (y - \bar{y}_\gamma)^2] \, \gamma(x,y) \, dx \, dy. \tag{5.2.6}$$

σ_γ as defined above can be interpreted as the standard deviation of the distribution of the function γ. It can also be interpreted as the radius of gyration of a plate of infinite area about its center of mass whose density distribution is given by $\gamma(x,y)$.

From the above definitions we have

$$\sigma^2_\gamma = \sigma^2_{x\gamma} + \sigma^2_{y\gamma} \, . \tag{5.2.7}$$

The Fourier transform of γ is denoted by Γ and is defined as

$$\Gamma(\omega, \nu) = \int_{-\infty}^{\infty} \gamma(x,y) \, e^{-j(\omega x + \nu y)} \, dx \, dy, \tag{5.2.8}$$

where and ω, ν are the spatial Fourier frequencies expressed in radians per unit distance.

The first partial derivatives of Γ with respect to ω and ν are denoted by Γ_{ω} and Γ_{ν} respectively. Similarly, the second partial derivatives will be denoted by $\Gamma_{\omega\omega}$, $\Gamma_{\omega\nu}$, and $\Gamma_{\nu\nu}$.

The six theorems stated below follow from the above definitions and notation.

Theorem 5.2.1a: $A_{\gamma} = \Gamma(0,0)$ ●

Theorem 5.2.1b: If A_{γ}=1 then $\bar{x}_{\gamma} = j\ \Gamma_{\omega}(0,0)$ ●

Theorem 5.2.1c: If A_{γ}=1 then $\bar{y}_{\gamma} = j\ \Gamma_{\nu}(0,0)$ ●

Theorem 5.2.1d: If A_{γ}=1, and $\bar{x}_{\gamma}$=0, then $\Gamma_{\omega\omega}(0,0) = -\ \sigma^2_{x\gamma}$ ●

Theorem 5.2.1e: If A_{γ}=1, and $\bar{y}_{\gamma}$=0, then $\Gamma_{\nu\nu}(0,0) = -\ \sigma^2_{y\gamma}$ ●

Theorem 5.2.1f: If A_{γ}=1, $\bar{x}_{\gamma}$=0, and $\bar{y}_{\gamma}$=0, then

$$\Gamma_{\omega\omega}(0,0) + \Gamma_{\nu\nu}(0,0) = -\ \sigma^2_{\gamma}\ \bullet$$

Theorem 5.2.2: If $z(x,y) = h(x,y) * w(x,y)$, A_h=A_w=1, $\bar{x}_h = \bar{x}_w = 0$, and $\bar{y}_h = \bar{y}_w = 0$, then A_z=1, $\bar{x}_z = 0$, $\bar{y}_z = 0$, and $\sigma^2_z = \sigma^2_h + \sigma^2_w$ ●

The *spread parameter* of a function is defined as the standard deviation of the distribution of the function. For example, σ_{ψ} is the spread parameter of the function ψ and σ_{γ} is the spread parameter of the function γ.

6. Point Spread Function

The *point spread function* of a camera system is the image brightness distribution produced by a point light source when the light flux incident on the camera from the point light source is one unit.

We will denote the point spread function by $h(x,y)$. Following the notation in section 5, we define the quantities A_h, $\bar{x}_h$, $\bar{y}_h$, σ_{xh}, σ_{yh}, and σ_h corresponding to the function h.

Since the definition of the point spread function specifies the incident light flux to be one unit, due to conservation of energy, we have

$$A_h = 1. \tag{6.1}$$

Let P be a point on a visible surface in the scene and p be its focused image (see Figure 1). The relation between the positions of P and p is given by the lens formula (1.1). If P is not in focus then it gives rise to a blurred image. According to geometric optics, the blurred image of P has the same shape as the lens aperture but scaled by a factor. The scaling factor depends on the amount of blur. More the blur or defocus, larger the scaling factor. When the image is in exact focus, according to geometric optics, the scaling factor is zero and therefore the image is a point of intense brightness. If the incident light energy on the camera system from P is b units, then the focused image can be described by $b\delta(x-\bar{x}_h, y-\bar{y}_h)$ where δ is the *Dirac Delta function.*

Without loss of generality, we will assume that the Cartesian coordinate system on the image plane is defined with its origin at $(\bar{x}_h, \bar{y}_h)$, i.e.

$$\bar{x}_h = 0\,, \quad \text{and} \tag{6.2}$$

$$\bar{y}_h = 0. \tag{6.3}$$

Therefore, $h(x,y)$ can be thought of as the response of the camera system to the input signal $\delta(x,y)$

6.1 Relation between σ_h and u: two examples

Usually camera systems have a circular aperture. In this case, the blurred image of a point on the image detector is circular in shape and is called the *blur circle.* Let r be the radius of this blur circle, R be the radius of the lens aperture, and s be the distance from the lens to the image detector plane (see Figure 1). Also let q be the scaling factor defined by

$$q = \frac{r}{R}\,. \tag{6.4}$$

In Figure 1, from similar triangles we have

$$\frac{r}{R} = \frac{s-v}{v} = s\left(\frac{1}{v} - \frac{1}{s}\right)\,. \tag{6.5}$$

Substituting for $1/v$ from equation (1.1) in the above equation, we obtain

$$q = s\left(\frac{1}{f} - \frac{1}{u} - \frac{1}{s}\right) . \tag{6.6}$$

Therefore

$$r = R\ q = Rs\left(\frac{1}{f} - \frac{1}{u} - \frac{1}{s}\right) . \tag{6.7}$$

Note that q and therefore r can be either positive or negative depending on whether $s \geq v$ or $s < v$. In the former case the image detector plane is *behind* the focused image of P and in the latter case it is *in front* of the focused image of P. However, in practice, the sign of r cannot be determined from a single image. This gives rise to a two-fold ambiguity in the determination of distance. This ambiguity can be avoided by setting the distance between the lens and the image detector to be equal to the focal length, i.e. $s = f$. In this case q is always negative. Therefore a unique solution is obtained for distance.

According to geometric optics, the intensity within the blur circle is approximately constant. Therefore, using equation (6.1) we get

$$h_1(x,y) = \begin{cases} \dfrac{1}{\pi r^2} & \text{if } x^2+y^2 \leq r^2 \\ 0 & \text{otherwise.} \end{cases} \tag{6.8}$$

But due to diffraction and non-idealities of lenses (Horn, 1986; Schreiber, 1986; Pentland, 1987; Subbarao, 1987a,b) an alternative model is suggested for the intensity distribution given by a two-dimensional Gaussian

$$h_2(x,y) = \frac{1}{2\pi\sigma^2}\ e^{-\frac{1}{2}\frac{x^2+y^2}{\sigma^2}} \tag{6.9}$$

where σ is the spread parameter such that

$$\sigma = k\ r \quad \text{for} \ \ k > 0 \ . \tag{6.10}$$

(k is a constant of proportionality characteristic of a given camera; it is determined experimentally at the beginning once and for all as a calibration step).

Suppose that the radius r in equation (6.7) is a constant over some region on the image plane because the distance u and the camera parameters $s, R,$ and f all remain the same. Then the camera acts as a linear shift-invariant system (cf. Rosenfeld and Kak, 1982). Therefore the observed image $g(x,y)$ is the result of convolving the corresponding focused image $f(x,y)$ with the camera's point spread function $h(x,y)$, i.e.

$$g(x,y) = h(x,y) * f(x,y) \tag{6.11}$$

where * denotes the convolution operation.

The point spread functions h_1 and h_2 defined above are only two specific examples used to clarify our method. In order to deal with other forms of point spread functions, we use the *spread parameter* σ_h to characterize them. As mentioned earlier, σ_h is the standard deviation of the distribution of the function h.

Using a polar coordinate system, it can be shown that the spread parameter σ_{h1} corresponding to h_1 is $r/\sqrt{2}$. Therefore, from equation (6.7) we have

$$\sigma_{h1} = m_1 u^{-1} + c_1 , \tag{6.12}$$

where

$$m_1 = -\frac{Rs}{\sqrt{2}} \quad \text{and} \quad c_1 = \frac{Rs}{\sqrt{2}} \left(\frac{1}{f} - \frac{1}{s} \right) . \tag{6.13}$$

We see above that for a given camera setting (i.e. for a given value of the camera parameters s, f, R) the spread parameter σ_{h1} depends linearly on inverse distance u^{-1}.

Using the results

$$\int_{-\infty}^{\infty} e^{-\frac{t^2}{2\sigma^2}} \, dt = \sqrt{2\pi}\, \sigma , \quad \text{and} \tag{6.14}$$

$$\int_{-\infty}^{\infty} t^2 \, e^{-\frac{t^2}{2\sigma^2}} \, dt = \sqrt{2\pi}\, \sigma^3 , \tag{6.15}$$

it can be shown that the spread parameter σ_{h2} of h_2 is $\sqrt{2}\,\sigma$. Therefore, from equations (6.10) and (6.7) we can write

$$\sigma_{h2} = m_2 u^{-1} + c_2 , \tag{6.16}$$

where

$$m_2 = -\sqrt{2}\ kRs \quad \text{and} \quad c_2 = \sqrt{2}\ kRs \left(\frac{1}{f} - \frac{1}{s} \right) . \tag{6.17}$$

We see above that for a given camera setting (i.e. for a given value of the camera parameters k, s, f, R) the spread parameter σ_{h2} depends linearly on inverse distance u^{-1}.

6.2 Relation between σ_h and u: the general case

We next proceed to show, based on geometric optics, that the spread parameter σ_h of the point spread function h is linearly related to inverse distance u^{-1} even when the camera's aperture is non-circular.

Let $P(x,y)$ be the pupil function defined in the plane of the camera's aperture (we shall use context to distinguish between the point source P in Figure 1 and the pupil function $P(x,y)$). $P(x,y)$ is defined to be 1 for points inside the aperture and 0 outside the aperture, i.e.

$$P(x,y) = \begin{cases} 1 & \text{if point } (x,y) \text{ is inside the aperture} \\ 0 & \text{otherwise.} \end{cases} \tag{6.18}$$

Following the notation in section 5, we define the quantities A_p, $\bar{x}_p$, $\bar{y}_p$, σ_{xp}, σ_{yp}, and σ_p for the pupil function $P(x,y)$. We now define a polar coordinate system (ρ,θ) in the aperture plane with its origin at $(\bar{x}_p,\bar{y}_p)$. We have

$$x - \bar{x}_p = \rho \cos\theta \quad \text{and} \quad y - \bar{y}_p = \rho \sin\theta . \tag{6.19}$$

In this polar coordinate system, let the boundary of the aperture be given by

$$\rho = R(\theta) . \tag{6.20}$$

Therefore the the pupil function is given by

$$P(\rho,\theta) = \begin{cases} 1 & \text{if } \rho < R(\theta) \\ 0 & \text{otherwise.} \end{cases} . \tag{6.21}$$

Let another polar coordinate system be defined on the image detector with its origin at $(\bar{x}_h,\bar{y}_h)$ and the x–axis along the same direction as that of the coordinate system in the aperture plane. In this coordinate system we have

$$x - \bar{x}_h = \rho \cos\theta \quad \text{and} \quad y - \bar{y}_h = \rho \sin\theta . \tag{6.22}$$

We noted earlier that the shape of the image of a point is a scaled replica of the shape of the aperture, and that the image brightness is a constant inside the region covered by the image. Therefore, in the new coordinate system, the boundary within which h is non-zero can be given by $|q|\,R(\theta+i\pi)$ where q is the scaling factor defined in equation (6.6) and

$$i = \begin{cases} 1 & \text{if } q \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{6.23}$$

When $i=1$, the shape of the image is the inverted (i.e. rotated by 180^0) replica of the aperture (see Figure 1).

Let A'_h be the area of the region within the boundary defined by $|q|\,R(\theta+i\pi)$. Then the image brightness within the region is $1/A'_h$. (Note A'_h is not to be confused with A_h defined by equation (5.2.1); $A_h = 1$ according to equation (6.1)).

The area A_p of the aperture is given by

$$A_p = \int_0^{2\pi}\int_0^{R(\theta)} \rho \, d\rho \, d\theta \;=\; \frac{1}{2}\int_0^{2\pi} R^2(\theta)\, d\theta \,. \tag{6.24}$$

The standard deviation σ_p of the pupil is given by

$$\sigma_p^2 = \frac{1}{A_p}\int_0^{2\pi}\int_0^{R(\theta)} \rho^3 \, d\rho \, d\theta \quad = \quad \frac{1}{4A_p}\int_0^{2\pi} R^4(\theta)\, d\theta \,. \tag{6.25}$$

Having noted the above relations, it is now easy to derive the following relations which relate the pupil function parameters to the point spread function parameters in terms of the scaling factor q:

$$A'_h = q^2 \, A_p \;, \tag{6.26}$$

$$h(\rho,\theta) = \begin{cases} \dfrac{1}{q^2 A_p} & \text{for } \rho < |q|\,R(\theta+i\pi) \\ 0 & \text{otherwise.} \end{cases} \tag{6.27}$$

$$\sigma_h = q \; \sigma_p \,. \tag{6.28}$$

$$\sigma_{xh} = q \; \sigma_{xp} \,. \tag{6.29}$$

$$\sigma_{yh} = q\ \sigma_{yp}\ . \tag{6.30}$$

Substituting for q from equation (6.6) in equation (6.28) we obtain

$$\sigma_h = m_h\ u^{-1} + c_h \tag{6.31}$$

where m_h and c_h are camera constants given by

$$m_h = -\ s\ \sigma_p\ , \quad \text{and} \quad c_h = s\left(\frac{1}{f} - \frac{1}{s}\right)\sigma_p\ . \tag{6.32}$$

For a given camera system, A_p and σ_p are fixed. Therefore, even for arbitrarily shaped aperture, we see above that σ_h is linearly related to inverse distance u^{-1}.

In actuality, the relation between σ_h and u^{-1} will be close to linear, but not exactly linear. The deviation from linearity is due to various effects such as diffraction, lens aberrations, noise, etc. However, in our method of determining distance, what is important is the fact that σ_h and u^{-1} are related. The fact that they may be linearly related is not crucial, but this information is useful in in the computational implementation of the method. We will describe this general method in section 13.5.

7. Line spread function

The *line spread function* of a camera system is the image brightness distribution produced on the image plane by a line light source.

In order to simplify the presentation of our theory, we assume that the focused image of the line source lies along the y–axis. A suitable translation and rotation of the coordinate system can always be performed to satisfy this condition. When this condition is satisfied, the line source can be represented by $\delta(x)$. Let $l(x)$ denote the line spread function. Then we have

$$l(x) = h(x,y) * \delta(x) \tag{7.1}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(\xi,\eta)\ \delta(x-\xi)\ d\xi d\eta$$

$$= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} h(\xi,\eta)\, \delta(x-\xi)\, d\xi \right] d\eta$$

$$= \int_{-\infty}^{\infty} h(x,\eta)\, d\eta\ .$$

The right hand side of the above expression is a *line integral* along a line parallel to the y-axis. For convenience we reexpress it as

$$l(x) = \int_{-\infty}^{\infty} h(x,y)\, dy. \tag{7.2}$$

The equations above imply that l is a function of only x and is independent of y irrespective of the form of $h(x,y)$. In the remaining part of this paper, whenever a function depends only on one of the coordinates, say x, we drop the other coordinate y from notation.

Following the notation described in section 5 for functions of one variable, we define A_l, $\overline{x}_l$, and σ_l corresponding to $l(x)$.

From equations (6.1) and (7.2) we obtain

$$A_l = 1. \tag{7.3}$$

From equations (5.1.2), (7.2), (7.3), (5.2.2), (6.2), (6.1) we obtain

$$\overline{x}_l = 0\ . \tag{7.4}$$

The line spread function corresponding to the point spread function $h_1(x,y)$ in equation (6.8) can be derived analytically using equation (7.2). This can be shown to be (see Figure 3)

$$l_1(x) = \begin{cases} \dfrac{2}{\pi r^2}\sqrt{r^2 - x^2} & \text{if } |x| \le r \\ 0 & \text{otherwise.} \end{cases} \tag{7.5}$$

Similarly we find that the line spread function corresponding to h_2 in equations (6.9), (6.10) is

$$l_2(x) = \frac{1}{\sqrt{2\pi}\,\sigma}\, e^{-\frac{1}{2}\frac{x^2}{\sigma^2}} \,. \qquad (7.6)$$

$l_2(x)$ above is the one-dimensional Gaussian line spread function.

We charactarize a general line spread function $l(x)$ by its *spread parameter* σ_l. It is equal to the standard deviation of the distribution of the function *l*. Substituting for ψ in equation (5.1.3) by *l*, substituting for γ in equation (5.2.4) by *h*, and using equations (7.2), (7.3), (6.1), we obtain

$$\sigma_l = \sigma_{xh} \,. \qquad (7.7)$$

From equations (6.29) and (7.7) we obtain

$$\sigma_l = q\sigma_{xp} \,. \qquad (7.8)$$

In the above equation, σ_{xp} can be determined from the pupil function. Therefore it is a known constant for a given camera system. Now, from equations (6.6), (7.8) we can write

$$\sigma_l = m_l\, u^{-1} + c_l \qquad (7.9)$$

where m_l and c_l are camera constants given by

$$m_l = -\, s\ \sigma_{xp} \ , \quad \text{and} \quad c_l = s\left(\frac{1}{f} - \frac{1}{s}\right)\sigma_{xp} \,. \qquad (7.10)$$

We see above that σ_l is linearly related to inverse distance u^{-1}. If the pupil function $P(x,y)$ is circularly symmetric about $(\bar{x}_p,\bar{y}_p)$, then σ_{xp} remains the same for all orientations of the x–axis. Therefore the constants m_l and c_l remain the same for all orientations of the line source.

When the pupil function is circularly symmetric, we note that $\sigma_{xp} = \sigma_{yp}$, and therefore, from equation (5.2.7) we get

$$\sigma_p = \sqrt{2}\ \sigma_{xp} \,. \qquad (7.11)$$

If the pupil function is circularly symmetric, then the point spread function is also circularly symmetric. In this case, it can be shown that

$$\sigma_h = \sqrt{2}\ \sigma_l \,. \qquad (7.12)$$

Above results can be verified for the two line spread functions in equations (7.5) and (7.6). We find that $\bar{x}_{l_1}$ and $\bar{x}_{l_2}$ corresponding to l_1 and l_2 respectively are both equal to zero (this can be easily verified from symmetry consideration). Therefore the spread parameters can be shown to be $\sigma_{l_1} = r/2$ and $\sigma_{l_2} = \sigma$. Since r and σ are linearly related to inverse distance u^{-1} (see equations (6.7) and (6.10)), $\sigma_{l_1}, \sigma_{l_2}$ are also linearly related to u^{-1}.

8. Edge Spread Function

Edge spread function is the image of a unit step edge.

A unit step edge along the y−axis can be described by the standard *unit step function* $u(x)$. Therefore, if $e(x)$ denotes the edge spread function corresponding to an edge along the y−axis, then we have

$$e(x) = h(x,y) * u(x) \ . \qquad (8.1)$$

$e(x)$ above gives the image brightness distribution produced by an edge whose brightness is zero to the left of y axis and one unit to the right of y axis.

It is shown in Appendix that the edge spread function $e_1(x)$ corresponding to $h_1(x,y)$ is

$$e_1(x) = \begin{cases} 1 & \text{if } x \geq r \\ 0 & \text{if } x \leq -r \\ \dfrac{1}{\pi r^2}\left[\pi r^2 - r^2 \, \mathrm{Cos}^{-1}\left(\dfrac{x}{r}\right) + x\sqrt{r^2 - x^2}\right] & \text{if } |x| < r \ . \end{cases} \qquad (8.2)$$

Similarly, the edge spread function $e_2(x)$ corresponding to $h_2(x,y)$ in equation (6.9) can be shown to be

$$e_2(x) = \frac{1}{\sqrt{2\pi}\,\sigma} \int_{-\infty}^{x} e^{-\frac{t^2}{2\sigma^2}} \, dt \ . \qquad (8.3)$$

$e_2(x)$ above is actually the one-dimensional cumulative normal distribution function.

9. Relation between edge spread and line spread functions

The unit step function $u(x)$ can be expressed as

$$u(x) = \int_{-\infty}^{x} \delta(t)\, dt \;. \tag{9.1}$$

Therefore we have

$$e(x) = h(x,y) * \int_{-\infty}^{x} \delta(t)\, dt \;. \tag{9.2}$$

Changing the order of integration and convolution, and noting that $l(x) = h(x,y) * \delta(x)$, we obtain

$$e(x) = \int_{-\infty}^{x} l(t)\, dt \;. \tag{9.3}$$

Denoting the derivative of $e(x)$ with respect to x by $e'(x)$ we obtain from the above equation

$$l(x) = e'(x) \;. \tag{9.4}$$

Therefore, given the edge spread function, the corresponding line spread function can be obtained from equation (9.4).

10. Determining Line Spread Function from Defocused Edges

Let $f(x,y)$ be a step edge along the y-axis on the image plane (see Figure 2a). Let a be the image intensity to the left of the y-axis and b be the height of the step. The image can be expressed as

$$f(x,y) = a + b\; u(x) \tag{10.1}$$

If $g(x,y)$ is the observed image then we have

$$g(x,y) = h(x,y) * f(x,y) \;. \tag{10.2}$$

Substituting for $f(x,y)$ from equation (10.1) in equation (10.2) and using equation (8.1) we obtain

$$g(x) = a + b\; e(x) \;. \tag{10.3}$$

In the above equation, we have dropped y from $g(x,y)$ for convenience. Now consider the derivative of g along the gradient direction, i.e. with respect to x. We will denote it by $g_x(x)$ (see Figure 2b). We have from equations (9.3), (9.4), (10.3),

$$g_x(x) = b\ l(x)\ . \tag{10.4}$$

From equations (10.4) and (7.3) we obtain

$$b = \int_{-\infty}^{\infty} g_x(x)\ dx\ . \tag{10.5}$$

We have derived above an equation to obtain the height b of the original step edge defined in equation (10.1). Having thus determined b, we obtain the line spread function $l(x)$ from equation (10.4) as (see Figure 2c)

$$l(x) = \frac{g_x(x)}{b}\ . \tag{10.6}$$

The spread parameter σ_l is then computed from $l(x)$. The distance is then obtained from equation (7.9) (or equation (11.1) below).

11. Determining Distance

In this section we will outline a method for determining the distance of a defocused step edge (see Figure 2a-d). The methods for determining the distance of point and line objects are similar to that of the method for edges presented here.

[1] Record the image $g(x)$ of the defocused edge described by equation (10.3).

[2] Compute the derivative of the recorded image perpendicular to the edge (i.e. along the gradient direction). For the edge described by equation (10.3), the derivative $g_x(x)$ of g with respect to x is computed.

[3] Compute the height b of the edge using equation (10.5).

[4] Compute the line spread function $l(x)$ using equation (10.6).

[5] Compute σ_l using equations (5.1.1), (5.1.2), and (5.1.3) (by substituting l for ψ in these equations).

[6] Compute the distance using the equation (see equation (7.9))

$$u^{-1} = \frac{\sigma_l - c_l}{m_l}\ . \tag{11.1}$$

In order to perform step [6] above, we need to know the values of m_l and c_l. There are two methods of determining these two quantities. The first method is by measuring the three camera parameters s, f, σ_{xp}. Having measured these parameters by some means, equation (7.10) is used to compute m_l and c_l. For most camera systems, the focal length f and the aperture radius R are provided by the manufacturer as part of the camera specifications. We have seen earlier that, for a circular aperture, $\sigma_{xp}=R/2$. Only the parameter s which is the distance between the lens and the image plane needs to be measured. There are many methods for doing this, the simplest being directly measuring the distance physically using a ruler.

The second method of determining m_l and c_l is through experiments. The minimal experiments include the following steps:

[1] Place an object with a step edge in front of the camera at a known distance, say u_1.

[2] Record the image of the edge and compute the corresponding spread parameter σ_l, say σ_{l1}.

[3] Move the edge object to a new position, say u_2, record the image, and compute the new spread parameter σ_l, say σ_{l2}. u_1 and u_2 should however satisfy a certain condition. Without loss of generality let us assume $u_2 > u_1$. Let u_0 be the distance to which the camera is focused. Then, u_1 and u_2 should be such that one of the following conditions should be satisfied

$$u_2 > u_1 \geq u_0 \ , \tag{11.2}$$

or

$$u_0 \geq u_2 > u_1 \ . \tag{11.3}$$

Satisfaction of the above condition can be checked easily by visual means. The object is moved gradually from u_1 to u_2 and for each position, it is checked to see that the change in image blur is monotonic, i.e. the blur should continuously increase or continuously decrease from position u_1 to u_2.

[4] Solve the following two simultaneous equations to obtain m_l and c_l:

$$\sigma_{l1} = m_l \, u_1^{-1} + c_l \quad \text{and} \quad \sigma_{l2} = m_l \, u_2^{-1} + c_l \ . \tag{11.4}$$

In practice, a more robust estimate of m_l and c_l can be made by recording the image of an edge for a large number of different positions and using a least squares technique to solve the system of linear equations.

Note that m_l and c_l need to be determined only once after one or more of the three camera parameters s, f, σ_{xp} are changed. It is not necessary to determine them each time the distance of a new object is to be determined. Therefore, if the camera parameters are fixed, then m_l and c_l need to be determined only once at the beginning. This constitutes the camera calibration step.

12. Obtaining the Point Spread Function

Finding the point spread function of a camera system directly by imaging a point light source is often impractical as it is difficult to realize an ideal point light source. In such situations, if the point spread function is circularly symmetric, then it can be obtained from the line spread function using the *inverse Abel Transform.* As we have seen earlier, the line spread function itself can be obtained by imaging a flat surface with a brightness step edge.

The inverse Abel transform essentially ‘‘back projects’’ the line spread function to give the point spread function. Let (ρ,θ) define a polar coordinate system with its origin at the center of symmetry of a circularly symmetric point spread function $h(\rho)$. The relation between $h(\rho)$ and the corresponding line spread function $l(x)$ is (cf. Horn ’86, page 143)

$$h(\rho) = -\frac{1}{\pi}\int_{\rho}^{\infty} \frac{l'(x)}{\sqrt{x^2-\rho^2}}\,dx = -\frac{1}{\pi}\int_{\rho}^{\infty} \sqrt{x^2-\rho^2}\;\frac{d}{dx}\left[\frac{l'(x)}{x}\right]\,dx\,, \tag{12.1}$$

where l' denotes the derivative of l with respect to x. The forward Abel transform is defined as

$$l(x) = 2\int_{x}^{\infty} \frac{\rho}{\sqrt{\rho^2-x^2}}\,h(\rho)\,d\rho. \tag{12.2}$$

One can verify the above two relations for the specific examples of the point spread functions (6.8), (6.9) and the line spread functions (7.5), (7.6) given earlier.

Although equation (12.1) can be used to recover the point spread function from the line spread function, in the experiments we conducted, we found the resolution of our images to be insufficient to do this. Note that equation (12.1) requires the derivative of the line spread function $l(x)$. This is essentially a second derivative of the original edge image g. The estimation of second derivative for the images in our experiments were very unreliable. However, this equation will be useful for better quality camera systems.

Our discussion in this section until now deals with a circularly symmetric point spread function. If the point spread function is non-symmetric, then it can still be obtained provided the line spread function is given for each possible orientation. Let L be the one-dimensional Fourier transform of the line spread function l for a line source through the origin and along a direction θ with respect to the x−axis. Also, let H be the two-dimensional Fourier transform of the point spread function h for a point source at the origin. Then it can be shown that L is equal to the cross-section of H along a line through the origin at an angle $\theta+\pi/2$ (cf. Rosenfeld and Kak, 1982; section 7.1.2). Therefore, if we know l for all possible θs then H can be determined. Taking the inverse Fourier transform of H gives the point spread function. For a circularly symmetric point spread function, the cross-section of H is the same for all lines through the origin. This suggests that the point spread function h can be determined given the line spread function at any one orientation.

13. Blobs, Stripes, and Generalized Edges

In real camera systems, even when an object such as a point source is in best possible focus, the recorded image is not a geometric point of infinitesimal dimensions, but is more like a ‘‘blob’’ having a small but a finite spread. Similarly, focused line sources appear as stripes of small width and focused step edges appear as smooth edges with a finite slope. The reason for this is that, the optical transfer function of any optical system has a finite cutoff frequency (cf. Goodman, 1968; equation (6-31)). Discrete image sampling and quantization also increase image blur.

Another reason for considering blobs, stripes, and smooth edges is that, often images are preprocessed by a smoothing filter to reduce noise. For example, edge-detection often involves smoothing by a filter such as a Gaussian. The effect of smoothing by a convolution filter $w(x,y)$ is to transform a point $\delta(x,y)$ into $w(x,y)$, a line $\delta(x)$ into $\int_{-\infty}^{\infty} w(x,y)\, dy$, and a unit edge $u(x)$ into $\int_{-\infty}^{x}\int_{-\infty}^{\infty} w(t,y)\, dt\, dy$. For example, smoothing may transform an underlying perfect step edge into a ramp edge having a nearly constant slope. In this section we see that such smoothing operations cause no loss of depth information in principle, and little loss of depth information in practice (provided smoothed image gray levels are stored on the computer as floating point numbers and not rounded off to the usual 8 bits).

The discussion in this section also applies to a class of blob objects, stripe objects, and smooth edge objects, which subtend a constant angle at the camera's lens irrespective of their distance from the camera system. As a counter example, the discussion here does not apply to a bright bar stripe whose absolute width is fixed, because, in this case, the width of the image of the bar stripe decreases as its distance from the camera increases. Therefore the angle subtended by the bar at the lens decreases with distance. (Note however that, if the width of the original bar object in the scene is known, then its width in the image gives depth information.)

13.1 Blobs

A *unit blob* $\alpha(x,y)$ is defined as an image pattern satisfying the three properties:

$$A_\alpha = 1 \, , \tag{13.1}$$

$$\overline{x}_\alpha = 0 \, , \quad \text{and} \quad \overline{y}_\alpha = 0 \, . \tag{13.2}$$

A blob can be thought of as a smeared point through the process $\delta(x,y) * \alpha(x,y)$, where $\alpha(x,y)$ has properties similar to that of a point spread function.

A blob of strength b is denoted by $f_\alpha(x,y)$ and is given by

$$f_\alpha(x,y) = b \; \alpha(x,y) \, . \tag{13.3}$$

The defocused image of the blob is

$$g_\alpha(x,y) = b \; z_\alpha(x,y) \tag{13.4}$$

where

$$z_\alpha(x,y) = \alpha(x,y) * h(x,y) \, . \tag{13.5}$$

From theorem (5.2.2) in section 5 we have $A_{z_\alpha} = 1$ and

$$\sigma^2_{z_\alpha} = \sigma^2_\alpha + \sigma^2_h \, . \tag{13.6}$$

Therefore,

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g_{\alpha}(x,y)\,dx\,dy = b\ . \tag{13.7}$$

From the above equations we see that σ_{z_α} can be computed from the observed image. Therefore, if we have prior knowledge of σ_α, then we can compute σ_h. The distance u can then be estimated as before from σ_h using equation (6.31). If $\alpha(x,y)$ is known, as for example is the case when it corresponds to an image smoothing filter, σ_α can be computed using equation (5.2.6). Even if the function $\alpha(x,y)$ is not known, we shall see later that σ_α can be estimated experimentally through camera calibration.

13.2 Stripes

A *unit stripe* $\beta(x)$ lying along the y−axis is defined as a non-negative function satisfying the properties

$$A_\beta = 1\ , \tag{13.8}$$

and

$$\bar{x}_\beta = 0\ . \tag{13.9}$$

A stripe can be thought of as a smeared line through the process $\delta(x) * \beta(x)$, or $\delta(x) * \alpha(x,y)$ where

$$\beta(x) = \int_{-\infty}^{\infty} \alpha(x,y)\,dy\ . \tag{13.10}$$

$\beta(x)$ has properties similar to that of a line spread function.

A stripe of strength b is denoted by $f_\beta(x)$ and is given by

$$f_\beta(x) = b\ \beta(x)\ . \tag{13.11}$$

The defocused image of a stripe is

$$g_\beta(x) = b\ z_\beta(x) \tag{13.12}$$

where

$$z_\beta(x) = \beta(x) * h(x,y)\ . \tag{13.13}$$

Since we can write $\beta(x) = \beta(x) * \delta(x)$, we have

$$z_\beta(x) = \left[\beta(x) * \delta(x)\right] * h(x,y) \tag{13.14}$$

$$= \beta(x) * \left[\delta(x) * h(x,y)\right]$$

$$= \beta(x) * l(x) \ .$$

From theorem (5.1.2) in section 5, we have $A_{z_\beta} = 1$, $\bar{x}_{z_\beta} = 0$, and

$$\sigma^2_{z_\beta} = \sigma^2_\beta + \sigma^2_l \ . \tag{13.15}$$

Therefore,

$$\int_{-\infty}^{\infty} g_\beta(x)\, dx = b \ . \tag{13.16}$$

From the above equations we see that σ_{z_β} can be computed from the observed image. Therefore, if we have prior knowledge of σ_β, then we can compute σ_l. The distance u can then be estimated as before from σ_l using equation (11.1). If $\beta(x)$ is known, as for example is the case when it corresponds to an image smoothing filter, σ_β can be computed using equation (5.1.3). Even if the function $\beta(x)$ is not known, we shall see later that σ_β can be estimated experimentally through camera calibration.

13.3 General Edges

A generalized unit edge along the $y-$ axis is denoted by $e_g(x)$ and is defined as

$$e_g(x) = \int_{-\infty}^{x} \beta(t)\, dt \tag{13.17}$$

where β is a unit stripe function defined earlier. $e_g(x)$ has properties similar to that of an edge spread function.

A generalized edge of height b along the $y-$ axis is denoted by $f_e(x)$ and is defined as

$$f_e(x) = a + b\ e_g(x) \ . \tag{13.18}$$

The defocused edge of a generalized edge is denoted by $g_g(x)$ and is given by $f_e(x) * h(x,y)$. It can be shown to be

$$g_g(x) = a + b\ z_e(x) \tag{13.19}$$

where

$$z_e(x) = \int_{-\infty}^{x} z_\beta(t)\ dt \tag{13.20}$$

and

$$z_\beta(x) = \beta(x) * l(x)\ . \tag{13.21}$$

If the derivative of $g_g(x)$ with respect to x is denoted by $g_{gx}(x)$, then it is easy to see that $g_{gx}(x) = g_\beta(x)$ in equation (13.12). Therefore, the distance can be found from g_{gx} by the same method as described earlier for $g_\beta(x)$.

13.4 Estimating σ_α and σ_β

We have seen above that the value of σ_α is needed to determine the distance of blob objects and the value of σ_β is needed to determine the distance of stripes and smooth edges. We will describe here a method of estimating σ_β through experiments for smooth edges. This method constitutes the camera calibration step. This calibration step needs to be done only once for each set of camera parameter values. It is not necessary to do this step each time the distance of a new object is to be determined. A method similar to that described here can be devised to determine σ_α for blob objects.

Equation (13.15) can be written as

$$\sigma_{z_\beta}^2(u^{-1}) = \sigma_\beta^2 + \sigma_l^2(u^{-1})\ . \tag{13.22}$$

The above equation makes explicit the fact that σ_{z_β} and σ_l are functions of inverse distance u^{-1}. In order to determine σ_β, we record images of an edge object for a large number of known values of u^{-1} at close intervals. For the position u^{-1} at which the lens formula (1.1) holds, the object is in focus according to paraxial geometric optics. At this position σ_l^2=0 (see equations (1.1), (6.6), and (7.8)) and at all other positions σ_l^2>0. Therefore $\sigma_{z_\beta}^2$ has the minimum value at this position and it is equal to σ_β^2. Denoting the value of σ_{z_β} corresponding to this minimum by $\sigma_{z_\beta}^{\min}$, we can write

$$\sigma_\beta = \sigma_{z_\beta}^{\min}\ . \tag{13.23}$$

σ_β can be estimated from the above equation. σ_l for any position is then computed from

$$\sigma_l = \sqrt{\sigma_{z_\beta}^2 - \sigma_\beta^2}\ . \tag{13.24}$$

Having computed σ_l, the distance is computed from equation (11.1).

13.5 A General Method

All along, until now, we have strived to derive a linear relation for computing distance. Such a relation can always be derived for a camera system which can be modeled adequately by paraxial geometric optics. Paraxial geometric optics is a good approximation to wave optics. If, however, a model based on paraxial geometric optics is not satisfactory for a camera system, and for some reason a linear relation cannot be derived, then an alternative method can be used. In this method, first we estimate the function $\sigma_{z_\beta}(u^{-1})$ through experiments at close intervals of u^{-1} and store it as a table of values in computer memory. This table is then ‘‘inverted’’ to make it indexable by the value of σ_{z_β} (see equation (13.15)) directly, and retrieve the corresponding value(s) of u^{-1}. For any computed value of σ_{z_β}, the distance is determined by simply referring to this table. The table can also be encoded in terms of a set of parameters by fitting piecewise polynomials, and using these parameters to compute distance. This method is general and can be used in most cases.

14. Determining camera parameters

In equation (6.6), note that the dependence of q on u and f are similar. q is linearly related to both u^{-1} and f^{-1}. Therefore, a method similar to that for finding u can be used for finding f. In this case, one has to keep u fixed and store σ_h or σ_l as a function of f^{-1}.

We can also obtain a method for determining σ_p if we can measure σ_h from the image, and know the factor q. q can be computed from equation (6.6) by measuring the values of s, f and u. σ_p is obtained directly from the computed value of σ_h using equation (6.28). Note that σ_p is a measure of the size of the camera's aperture. A similar method can be used to determine σ_{xp} from σ_l using equation (7.8). In the case of a camera with circular aperture, these methods are useful in determining the radius of the camera's aperture.

15. Noise

There are three sources of noise. The first is the signal noise due to the electronic hardware that converts light energy to analog electrical signals. The other two sources are quantization and discrete sampling. The effects of these three sources of noise on our method of depth recovery are discussed in the Appendix.

All quantities computed from the recorded images are affected by noise. For example, if $\hat{\sigma}_l$ denotes the computed value of σ_l, then we can write

$$\hat{\sigma}_l = \sigma_l + \sigma_{lN} \tag{15.1}$$

where σ_{lN} represents the unknown noise term. In our experiments, we study the effects of noise on the various computed quantities in terms of the standard deviation of the computed quantities. For example, we find in our experiments that the standard deviation of $\hat{\sigma}_l$ is usually about 0.1 pixel. See next section for more details.

16. Experiments and computational steps

The goal of our experiments was two fold: (i) to verify the applicability of our mathematical model to practical camera systems, and (ii) to test the usefulness of the method in practical applications.

Rectangular sheets of a black paper and a white paper were pasted adjacent to each other on a cardboard. This created a step edge corresponding to a step discontinuity in reflectance along a straight line. The cardboard was placed directly in front of a camera such that the edge was vertical and located near the center of the camera's field of view. The camera was focused to 24'' and nine images of the cardboard were acquired (see Figure 4) for distances 8'', 11'', 14'', 17'', 20'', 24'', 33'', 48'', and 100'' between the cardboard and the camera. The camera setting (i.e. camera parameters) remained the same for all images. The acquired images were trimmed to size 64×64 so that the edge was approximately in the middle of the trimmed images. The edge was perpendicular to the rows and extended from top row to the bottom row.

The camera used by us was a Panasonic WV-CD50, focal length 16 mm, aperture diameter 11.4 mm, pixel size 0.013 mm × 0.017 mm, and 8 bits/pixel.

In our experiments, since the edge on the cardboard was vertical, its image was along a column of the image matrix. Therefore the gray values of pixels in a column were approximately constant, but changed more or less monotonically along the rows.

Each of the 64×64 images were cut perpendicular to the edge into 16 equal strips of size 4 rows × 64 columns. The edge was located approximately at the middle column of these image strips. In each strip, the gray values of pixels in the same column (4 pixels) were averaged. We will denote the resulting average pixel values of a strip by $\hat{g}(j)$ for $j=0,1,2,...,63$.

For each image strip, the first derivative of $\hat{g}(j)$ was computed along the intensity gradient by simply taking the difference of gray values of adjacent pixels. The intensity gradient in our images was along the rows since the edge was perpendicular to the rows. Let $\hat{g}_x(j)$ represent the derivative. It is computed as

$$\hat{g}_x(j) = \hat{g}(j+1) - \hat{g}(j) \quad \text{for } j=0,1,2,...,62. \tag{16.1}$$

An initial estimate of the height $\hat{b}$ of the edge was computed by

$$\hat{b} = \sum_{j=0}^{62} \hat{g}_x(j). \tag{16.2}$$

The approximate position $\hat{\bar{j}}$ of the edge was then estimated by computing the first moment of $\hat{g}_x(j)$ as (see equation (5.1.2))

$$\hat{\bar{j}} = \frac{1}{\hat{b}} \sum_{j=0}^{62} j\, \hat{g}_x(j)\ . \tag{16.3}$$

The following ‘‘noise cleaning’’ step was included to reduce the effects of noise. $\hat{g}_x(j)$ was traversed on either side of position $\hat{\bar{j}}$ until a pixel was reached where either $\hat{g}_x(j)$ was zero or its sign changed. All pixels between this pixel (where, for the first time, $\hat{g}_x$ became zero or its sign changed) and the pixel at the row’s end were set to zero (see Figure 5). This step is summarized in the following piece of psuedo-C-code:

```
k = ĵ̄              /* k is the moving index */
while( ( k > 0 ) and ( ĝx(k) * ĝx(k-1) > 0 ) )
    { k = k - 1 }
```

```
j1 = k                    /* marker for start position */
k = k - 1
while( k ≥ 0 )
    { ĝx(k) = 0 ;  k = k - 1 }

k = ĵ
while( ( k < 62 ) and ( ĝx(k) * ĝx(k+1) > 0 ) )
    { k = k + 1 }
j2 = k                    /* marker for end position */
k = k + 1
while( k ≤ 62 )
    { ĝx(k) = 0 ;  k = k + 1 }
```

The above step sets to zero the non-zero values for $\hat{g}_x(j)$ resulting from noise at points far away from the location of the edge.

The $\hat{g}_x(j)$ obtained by noise cleaning is used to recompute a refined estimate of the height $\hat{b}$ of the step edge using, again, equation (16.2). Next the line spread function is computed from

$$\hat{l}(j) = \frac{\hat{g}_x(j)}{\hat{b}} \quad \text{for } j{=}0,1,2,...,62. \tag{16.4}$$

Note that $\hat{l}$ computed above is not exactly the line spread function, but corresponds to z_β in equations (13.14) and (13.21). However we shall use $\hat{l}$ as σ_β in equation (13.23) was relatively small in our experiments. The image for distance u=24″ was visually judged to be the best possible focused image. From this image, σ_β was estimated to be 0.977 pixel.

We recompute $\hat{\bar{j}}$ using the noise-cleaned $\hat{g}_x(j)$ from equation (16.3) to obtain a better estimate of the location of the edge. An estimate of the spread $\hat{\sigma}_l$ of the line spread function is computed from

$$\hat{\sigma}_l = \pm \sqrt{\sum_{j=0}^{62} (j - \hat{\bar{j}})^2 \, \hat{l}(j) - \sigma_\beta^2} \; . \tag{16.5}$$

The results of the experiments are given in Table 1. Each of the nine images had 16 strips of size 4×64. For each image strip, location $\hat{\bar{j}}$ of the edge, height $\hat{b}$ of the edge, and the spread parameter $\hat{\sigma}_l$ of the line spread function were computed. The mean value (μ) and the standard deviation (σ) of these quantities over the 16 strips are tabulated for each image. The standard deviations give a measure of the uncertainty in the computed quantities due to noise and discretization effects.

We see in Table 1 that, the location of the edge is close to the center of the image (32 pixels) for most images. The height of the edge ranged from about 25 gray levels to 50 gray levels and the spread parameter ranged from about 0 to 7 pixels. The standard deviation of the position of the edge is about 0.15 pixel, and, for edge height it is about 1.5 gray level. The standard deviation of the spread parameter is about 0.1 pixel. We see that all quantities are reasonable and the standard deviations are comparable to the measurement errors.

Figure 6 shows a plot of the mean value of $\hat{\sigma}_l$ as a function of inverse distance u^{-1}. We see that this function is very nearly linear on either side of the position corresponding to the focused distance which is $u = 24''$. Further, the slopes of the graph on either side of the focused position are almost exactly equal. This confirms the linear relation predicted by equation (7.9) and thus validates our theoretical model. The sign of $\hat{\sigma}_l$ in equation (16.5) cannot be determined without some *a priori* knowledge. It is because of this reason that the graph in Figure 6 has a ‘V’ shape. This implies a two-fold ambiguity in distance u for a given value of $\hat{\sigma}_l$ (see Figure 2d). As mentioned earlier, this ambiguity can be avoided, for example, by setting $s = f$. In this case the focused image is always behind the image detector and $\hat{\sigma}_l$ is always negative.

Figure 7 has been obtained by enclosing the graph of Figure 6 with a dark band whose vertical width at each point is approximately two times the standard deviation of $\hat{\sigma}_l$. This shows that, at positions close to the camera’s focused distance, $\hat{\sigma}_l$ is very small, but the error $\delta\hat{\sigma}_l$ is high. Therefore the uncertainty in distance is higher. At other positions, the error is usually small or moderate.

Two more set of experiments, each similar to the one described above but with different camera settings were performed. The camera was focused to distances 14 inches and 17 inches respectively for the first and second set of experiments. Nine images were recorded and processed in each set of experiments. The results of these experiments were similar to those reported here.

We found that noise cleaning step described earlier was very important in our experiments. A small non-zero value of image derivative caused by noise at pixels far away from the position of the edge affects the estimation of $\hat{\sigma}_l$ considerably. The farther a pixel is from the edge, the higher its effect on the final result (see error analysis in the appendix) because the distance is squared in the computation of $\hat{\sigma}_l$.

The effective range of our method depends on the constants m_l, c_l in equation (7.9) and the image quality in terms of spatial and gray level resolution. This method is more effective for objects at shorter distances than at longer distances because blur varies linearly with inverse distance. The maximum distance of the cardboard in our experiments was about 8 feet. The range can be increased by using a custom designed camera which produces high quality images.

17. Conclusion

We have described a general method for determining the distance of a simple class of objects by measuring the degree of image blur. The method is more accurate for nearby objects than for distant objects. One limitation of the method is that it is restricted to isolated objects; presence of other objects nearby (within a distance of about twice the spread parameter of the object) affects depth estimation.

Methods for determining the distance of more complicated objects can be found in (Subbarao, 1988,89). But these methods use two images as opposed to only one image in the method presented here.

18. References

Born, M. and E. Wolf. 1975. *Principles of Optics.* Pergamon Press.

Gaskill, J. D., 1978. *Linear Systems, Fourier Transforms, and Optics.* John Wiley & Sons.

Goodman, J. W., 1968. *Introduction to Fourier Optics.* McGraw-Hill Book Company.

Grossman, P. 1987 (Jan.). Depth from focus. *Pattern Recognition Letters 5*. pp. 63-69.

Hecht, E., and A. Zajac. 1979. *Optics.* Addison-Wesley Publishing Company.

Horn, B.K.P. 1968. Focusing. Artificial Intelligence Memo No. 160, MIT.

Horn, B.K.P. 1986. *Robot Vision.* McGraw-Hill Book Company.

Hwang, T., J. J. Clark, and A. L. Yuille, 1989. A depth recovery algorithm using defocus information. *Proceedings of Computer Vision and Pattern Recognition,* San Diego, California.

Jarvis, R. A. March 1983. A perspective on range finding techniques for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* PAMI-5, No. 2, pp. 122-139.

Krotkov, E. 1986. Focusing. MS-CIS-86-22. Grasp lab 63. Dept. of Computer and Information Science. University of Pennsylvania.

Ligthart, G. and F. C. A. Groen. 1982. A comparison of different autofocus algorithms. *Proceedings of the International Conference on Pattern Recognition.*

Ogle, K.N. 1968. *Optics.* Chapter VII, Section 2. Charles C Thomas . Publisher. Springfield, Illinois.

Pentland, A.P. 1987. A new sense for depth of field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 4, pp. 523-531.

Pentland, A.P., T. Darrel, M. Turk, and W. Huang, 1989. A simple, real-time range camera. *Proceedings of Computer Vision and Pattern Recognition,* San Diego, California.

Rosenfeld, A., and A.C. Kak. 1982. *Digital Picture Processing,* Vol. I. Academic Press, Inc.

Schlag, J.F., A.C. Sanderson, C.P. Neuman, and F.C. Wimberly. 1983. Implementation of automatic focusing algorithms for a computer vision system with camera control. CMU-RI-TR-83-14. Robotics Institute. Carnegie-Mellon University.

Schreiber, W.F. 1986. *Fundamentals of Electronic Imaging Systems.* Springer-Verlag. Section 2.5.2.

Subbarao, M. 1987a. Direct recovery of depth-map I: differential methods. *Proceedings of the IEEE Computer Society workshop on Computer Vision,* pp. 58-65, Miami Beach, Florida, Nov. 30 to Dec. 2.

Subbarao, M. 1987b (May). Progress in research on direct recovery of depth and motion. Technical report 87-04. Image Analysis and Graphics Laboratory, State University of New York at Stony Brook.

Subbarao, M. 1987c. *Method and apparatus for determining the distances between surface patches of a three-dimensional spatial scene and a camera system.* U. S. patent application no. 126407 (pending). Nov. 1987.

Subbarao, M. December 1988. Parallel depth recovery by changing camera parameters. *Second International Conference on Computer Vision,* Florida, USA, pp. 149-155.

Subbarao, M. 1989. Efficient depth recovery through inverse optics. Editor: H. Freeman, *Machine Vision for Inspection and Measurement*, Academic press, Boston, pp. 101-126.

Subbarao, M. and G. Natarajan. June 1988. Depth recovery from blurred edges. *Proceedings of Conference on Computer Vision and Pattern Recognition,* pp. 498-503, Ann Arbor, Michigan.

Tenenbaum, J. M., 1970. *Accommodation in Computer Vision.* Ph.D. Dissertation, Stanford University.

APPENDIX

A. Edge spread function for a circular aperture

Referring to equations (8.1) and (6.8) we have

$$e_1(x) = h_1(x,y) * u(x) \tag{a1}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_1(\xi,\eta)\, u(x-\xi)\, d\xi\, d\eta$$

Figure 8 is a diagram representing the above equation. In this figure, (ξ,η) define a Cartesian coordinate system. C is a circle of radius r and centered at the origin. $h_1(\xi,\eta)$ is zero outside the circle C and is $1/(\pi r^2)$ inside the circle. AB is a line given by $\xi = x$. $u(x-\xi)$ is 1 to the left of this line, and is 0 to the right of this line. Therefore, if $|x| < r$ it can be shown that

$$e_1(x) = \frac{1}{\pi r^2} \times [\text{Area of circle } C \text{ to the left of line } AB] \tag{a2}$$

$$= \frac{1}{\pi r^2} \times \begin{bmatrix} \text{Area of circle C} - \\ \text{Area of sector OADB} + \\ \text{Area of triangle OAB} \end{bmatrix}$$

$$= \frac{1}{\pi r^2} \times \left[\pi r^2 - \frac{1}{2}\,(2\,\theta)\, r^2 + \frac{1}{2}\, x\,(2\,y) \right]$$

$$= \frac{1}{\pi r^2} \times \left[\pi r^2 - r^2\, \mathrm{Cos}^{-1}\left(\frac{x}{r}\right) + x\,\sqrt{r^2 - x^2} \right].$$

We have above one part of the result of equation (8.2). It is trivial to verify the other two parts of equation (8.2) from the diagram in Figure 8.

B.1 Error Analysis

The inverse distance u^{-1} is linearly related to the spread parameter σ_l (equation 7.9). Therefore, the error in u^{-1} denoted by δu^{-1} is also linearly related to the error in σ_l denoted by $\delta\sigma_l$. We will now derive expressions for $\delta\sigma_l$ in terms the effects of noise and digitization.

Let the distance between two pixels be the unit of spatial distance. Also let x be a continuous parameter along a row of pixels such that the pixels from left end to the right end are located at positions $0, 1, 2, ..., N-1$. We will denote the noise-free analog image signal by $g(x)$ and its partial derivative with respect to x by $g_x(x)$.

At any pixel j for $j=0, 1, 2, ..., N-1$, let the measured value of image brightness be $\hat{g}(j)$ such that

$$\hat{g}(j) = g(j) + n(j) \tag{b1}$$

where $n(j)$ is a random variable representing zero-mean additive noise. $n(j)$ represents the combined effect of both signal noise and quantization, i.e.

$$n(j) = n_s(j) + n_q(j) \tag{b2}$$

where n_s is the analog signal noise, and n_q is the quantization noise. Usually n_s and n_q are of the same order, and both are zero-mean. Typically n_q has a uniform distribution in the range [–0.5,+0.5] in units of gray level. Therefore its' standard deviation is $1/(2\sqrt{3})$. The standard deviation of n_s is usually about a fraction of a gray level (about 0.1 or so). Let σ_n, σ_{ns}, and σ_{nq} denote the noise standard deviations of n, n_s, and n_q, respectively. Then we have $\sigma_n^2 = \sigma_{ns}^2 + \sigma_{nq}^2$.

Let $\hat{g}_x(j)$ be the estimated partial derivative with respect to x at pixel j. Using equations (10.3), (10.4), we have

$$\hat{g}_x(j) = \hat{g}(j+1) - \hat{g}(j) \; . \tag{b3}$$

$$= g(j+1) - g(j) + n(j+1) - n(j) \; ,$$

$$= b\, \tilde{l}(j) + n_x(j)$$

where

$$\tilde{l}(j) = e(j+1) - e(j) \tag{b4}$$

and

$$n_x(j) = n(j+1) - n(j)\ . \tag{b5}$$

$\tilde{l}$ is a discrete estimation of l in the absence of noise. The error in this estimation given by $\tilde{l} - l$ is purely due to image sampling at finite discrete intervals rather than at infinitesimally small intervals. This error decreases as the size of the sampling interval decreases.

n_x is effectively a random variable which is a summation (or difference) of two random variables $n(j)$, $n(j+1)$. Since the two random variables are uniformly distributed and zero-mean, their summation and difference both represent a random variable with identical statistical properties. If the noise n at adjacent pixels are independent, then n_x is a random variable having a triangular distribution with its peak value being 1.0 located at the origin and decreasing linearly on either side and becoming zero at -1.0 and +1.0. The standard deviation of this noise is $\sqrt{2}\ \sigma_n$.

B.2 Error in the computation of *b*

We have

$$\Delta b = \hat{b} - b \tag{b6}$$

$$= \sum_j \hat{g}_x(j) - b$$

$$= \sum_j \left[b\ \tilde{l}(j) + n_x(j) \right] - b$$

$$= b \left[\sum_j \tilde{l}(j) - 1 \right] + n_b$$

where

$$n_b = \sum_j n_x(j)\ . \tag{b7}$$

n_b is a summation of many zero-mean independent random variables. Therefore n_b is also zero mean and follows approximately the normal distribution. If the summation involves m pixels, then the standard deviation of n_b is $\sqrt{m}$ times the standard deviation of n_x, i.e. $\sqrt{2m}\,\sigma_n$. Due to the special noise cleaning method employed in our algorithm, m is usually about 4 times the spread parameter σ_l and therefore the standard deviation of n_b becomes approximately $2\sqrt{2\sigma_l}\ \sigma_n$. Therefore, the error $\delta\sigma_l$ itself depends on σ_l!

The term $\sum_j \tilde{l}(j) - 1$ gives the error purely due to discrete sampling. As the sampling rate increases, this error decreases. Given the sampling rate and the analytic equation for $e(x)$, this error can be estimated purely through computational means.

B.3 Error in $\bar{j}$

$$\hat{\bar{j}} - \bar{j} = \frac{1}{\hat{b}} \sum_j j\ \hat{g}_x(j) - \frac{1}{b}\int_{-\infty}^{\infty} x\ g_x(x)\ dx \tag{b8}$$

$$= \frac{1}{\hat{b}} \sum_j j\ \left(b\ \tilde{l}(j) + n_x(j)\right) - \frac{1}{b}\int_{-\infty}^{\infty} x\ b\ l(x)\ dx$$

$$= \left[\frac{b}{\hat{b}} \sum_j j\ \tilde{l}(j) - \int_{-\infty}^{\infty} x\ l(x)\ dx \right] + \frac{1}{\hat{b}} \sum_j j\ n_x(j)\ .$$

If there is no error in $\hat{b}$, i.e. $\hat{b}=b$, then the term in square brackets above gives the error due to discrete sampling. The last term above is a weighted sum of many independent identically distributed random variables. It is a zero-mean random variable which approximately follows normal distribution. If the upper and lower limits of summation are $p+1$ and $p+m$, then the standard deviation of this random variable is

$$\frac{\sqrt{(p+1)^2+(p+2)^2+(p+3)^2+...+(p+m)^2}}{\hat{b}}\ \sqrt{2}\ \sigma_n\ . \tag{b9}$$

As mentioned earlier, in our algorithm m is approximately four times σ_l. Therefore, setting $m=4\sigma_l$ and $p=0$ we get

$$2\sqrt{\frac{\sigma_l(4\sigma_l+1)(8\sigma_l+1)}{3}}\ \ \frac{\sigma_n}{\hat{b}}\ . \tag{b10}$$

We see above that the effect of this noise reduces inversely with the estimated edge height $\hat{b}$. Therefore, the larger the edge height, the lower is the error in its estimated position.

B.4 Error in σ_l^2

$$\hat{l}(j) = \frac{1}{\hat{b}}\,\hat{g}_x(j) \tag{b11}$$

$$= \frac{b}{\hat{b}}\,\tilde{l}(j) + \frac{n_x(j)}{\hat{b}}\ .$$

Now, neglecting error in σ_β^2, we have

$$\Delta\sigma_l^2 = \hat{\sigma}_l^2 - \sigma_l^2 \tag{b12}$$

$$= \sum_j (j - \hat{\bar{j}})^2\,\hat{l}(j) - \int_{-\infty}^{\infty} (x - \bar{j})^2\ l(x)\ dx$$

$$= \left[\frac{b}{\hat{b}}\sum_j (j - \hat{\bar{j}})^2\,\tilde{l}(j) - \int_{-\infty}^{\infty} (x - \bar{j})^2\ l(x)\ dx\right] + \frac{1}{\hat{b}}\sum_j (j - \hat{\bar{j}})^2 n_x(j)\ .$$

In the above equation, if the error in b and $\bar{j}$ are negligible (i.e. $\hat{b}$=b and $\hat{\bar{j}}$=$\bar{j}$) then the term in square brackets gives the effect of discrete sampling, and the other term gives the effects of signal noise and quantization noise. In this case, the effect of discrete sampling is independent of the edge height b but the effect of noise reduces inversely with the edge height. The noise term is actually a weighted sum of many independent identically distributed zero-mean random variables. Therefore this term represents another zero-mean random variable. If $p+1$ and $p+m$ are the limits of summation, then the standard deviation of the noise term is

$$\frac{1}{\hat{b}}\sqrt{\sum_{i=1}^{m}(p+i-\hat{\bar{j}})^4}\ \ \sqrt{2}\ \sigma_n\ . \tag{b13}$$

As in the previous cases, the value of m is about $4\sigma_l$. An examination of the noise term shows that the farther a pixel is from the location of the edge, the more is the effect of noise at that pixel on the computed value of σ_l. This is due to the factor

$(j - \hat{j})^2$ **in the noise term.**

The error $\delta\sigma_l$ is given by

$$\delta\sigma_l = \frac{\Delta\sigma_l^2}{2\sigma_l} \ . \qquad \text{(b14)}$$

Finally the error δu^{-1} in inverse distance is obtained from equation (7.9) as

$$\delta u^{-1} = \frac{\delta\sigma_l - c_l}{m_l} \ . \qquad \text{(b15)}$$

The error analysis here brings out the effects of quantization, discrete sampling, and signal noise, on the error in the final result. This analysis suggests ways of improving the computational algorithm. It is clear that the effect of image signal noise can be reduced by smoothing the image parallel to the edge (i.e. perpendicular to the gradient direction). For example, a smoothing filter such as a simple averaging filter or a Gaussian filter can be employed. Suppose, for example, we average m rows of pixels parallel to the edge, then the standard deviation of original noise n will be reduced by a factor of $1/\sqrt{m}$.

Averaging parallel to the edge does not reduce quantization noise because the quantization errors in pixels parallel to the edge are highly correlated.

The effect of all three factors-- quantization, discrete sampling, and signal noise, can be reduced if one has prior knowledge about the form of the edge spread function $e(x)$. In this case, one can fit a curve corresponding to this form to the data through a least square error minimization technique. For example, if e is known to be a smooth analytic function, then one can fit piecewise polynomials to the data. These polynomials can be used to perform exact differentiation and integration rather than the differencing and summing operations employed in our computational algorithm. For example, equation (10.5) can be used instead of (16.2).

An alternative method is to directly estimate σ_l from $e(x)$ rather than going through several computational steps. For example, if the form of e is known to be as in equation (8.2), then one can compute the mean square error between the data and $e(x)$ for various values of r, and find the value of r which minimizes the the mean square error. σ_l then is given by $r/2$. This method involves searching for r that minimizes the mean square error. An initial estimate of r can be obtained by the algorithm employed in our experiments.

This value can then be refined through an iterative search.

$\mu(\cdot)$: mean value of $\cdot$ $\sigma(\cdot)$: standard deviation of $\cdot$

$\mu(\sigma_\beta) = 0.977$ pixels $\sigma(\sigma_\beta) = 0.0628$ pixels

u inches	$\mu(\hat{\bar{j}})$ pixels	$\sigma(\hat{\bar{j}})$ pixels	$\mu(\hat{b})$ gray levels	$\sigma(\hat{b})$ gray levels	$\mu(\hat{\sigma}_l)$ pixels	$\sigma(\hat{\sigma}_l)$ pixels
8	40.85	0.638	46.88	3.407	7.060	0.4543
11	37.37	0.133	48.94	1.197	4.165	0.0951
14	31.34	0.104	48.19	1.509	2.552	0.0758
17	34.06	0.083	49.88	0.696	1.510	0.0760
20	35.00	0.058	47.75	1.031	0.598	0.0794
24	33.10	0.059	47.50	1.275	0.011	0.3283
33	35.64	0.076	41.38	0.781	0.999	0.0548
48	31.16	0.158	33.94	1.088	1.698	0.1306
100	30.00	0.373	25.56	1.273	2.598	0.0857

Table 1

Address for correspondence:

Muralidhara Subbarao
Department of Electrical Engineering
State University of New York at Stony Brook
Stony Brook, NY 11794-2350, USA

Phone No. (516) 632-8405/8400

Index Terms

depth recovery, depth-from-focusing, defocused images,
point spread function, line spread function, edge spread function,
paraxial geometric optics, blur circle, convolution,
moments of a function, camera parameters.