Computer Modeling and Simulation of Camera Defocus

Murali Subbarao and Ming-Chin Lu
Department of Electrical Engineering
State University of New York
Stony Brook, NY 11794

Abstract

A computer model is presented for the formation and sensing of defocused images in a typical CCD camera system. A computer simulation system named IDS has been developed based on the computer model. IDS takes as input the camera parameters and the scene parameters. It produces as output a digital image of the scene as sensed by the camera. IDS consists of a number of distinct modules each implementing one step in the computer model. The modules are independent and can be easily modified to enhance the simulation system. IDS is being used by our group for research on methods and systems for determining depth from defocused images. IDS is also being used for research on image restoration. It can be easily extended and used as a research tool in other areas of machine vision. IDS is machine independent and hence portable. It provides a friendly user interface which gives the user full access and control to the parameters and intermediate results. IDS is available for distribution to other research groups upon payment of a nominal fee.

1 Introduction

The experimental verification of computer vision theories often requires sophisticated and expensive camera systems. When such camera systems are not available, one can use a computer to simulate the camera system and to evaluate the theories. Such simulation requires the development of a mathematical model of the camera system and a computational implementation of the model. In this paper we describe a mathematical model and its computational implementation which is useful in testing theories on image defocus.

Previous literature [6, 7, 8] in computer vision on camera system modeling is mostly descriptive in nature. Shafer's work [8] is perhaps the first to deal with a detailed description of the image sensing process. Many important aspects of the image sensing process which are routinely ignored by computer vision researchers are discussed in detailed by him. He has presented a twelve-parameter model for a robot imaging system — six parameters in camera position and orientation, three in the optical system, and three in sensitivity. In addition, he provides valuable information on the state of the art in the imaging system technology and calibration. More recently, Subbarao and Nikzad [12] have presented another detailed mathematical model of a camera system. The camera model in this paper is derived from that of Subbarao and Nikzad. Unlike previous work, here we have gone one step further and implemented the model proposed by us. The implementation is nearly optimal in terms of computational resources. Many examples of images obtained through simulation are included in this paper.

Our goal is to present what we believe to be a *useful* mathematical model for computer implementation rather than a *detailed* mathematical model. The model makes explicit the sequence of transformations that the light incident on a camera system undergoes before being sensed by the image detector, and the sequence of operations by which the sensed image is converted to digital data. Based on this model, an Image Defocus Simulator (IDS)

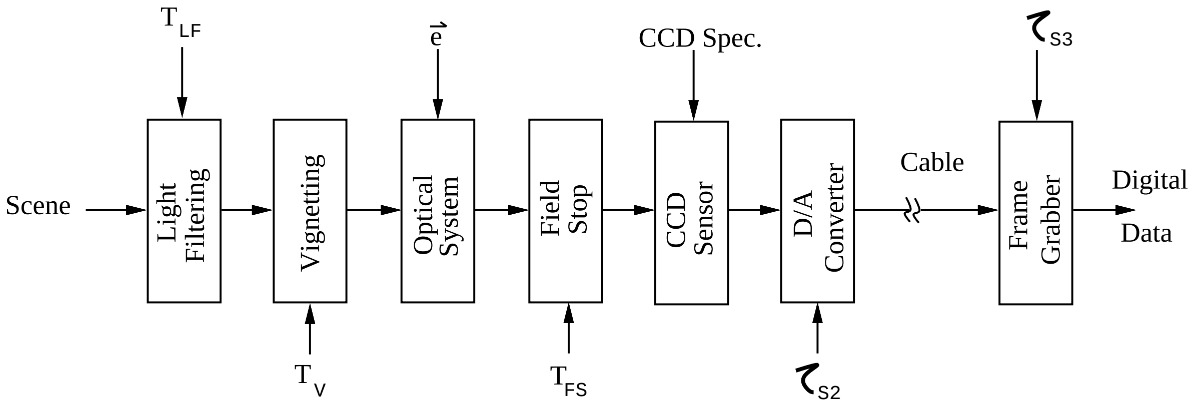


Fig. 1 Block diagram of a video camera system

has been developed.

An user interface has been developed for using IDS. It is a menu-driven interface which gives the user easy access/control to the input parameters and intermediate results. The output image is displayed on a monitor, and it can be stored for further processing.

This paper is organized as follows: Section 2 briefly discusses the mathematical model of the camera system; Section 3 describes the IDS computer simulation system; Section 4 presents some examples of the simulated pictures and the application of camera defocus; and finally, Section 5 concludes the paper with a few remarks and future work related parallel implementation of IDS.

2 Camera Model

A block diagram of a typical CCD video camera system used in machine vision applications is shown in Fig. 1. Inside the optical system module, there are in general many lenses and aperatures. However, we are concerned only with the terminal properties of the aggregate [3] as shown in Fig. 2. The terminals of this module are an *entrance pupil* (effective or real) representing a finite aperture through which the light must pass to reach the imaging elements and an *exit pupil* (again effective or real) representing a finite aperture

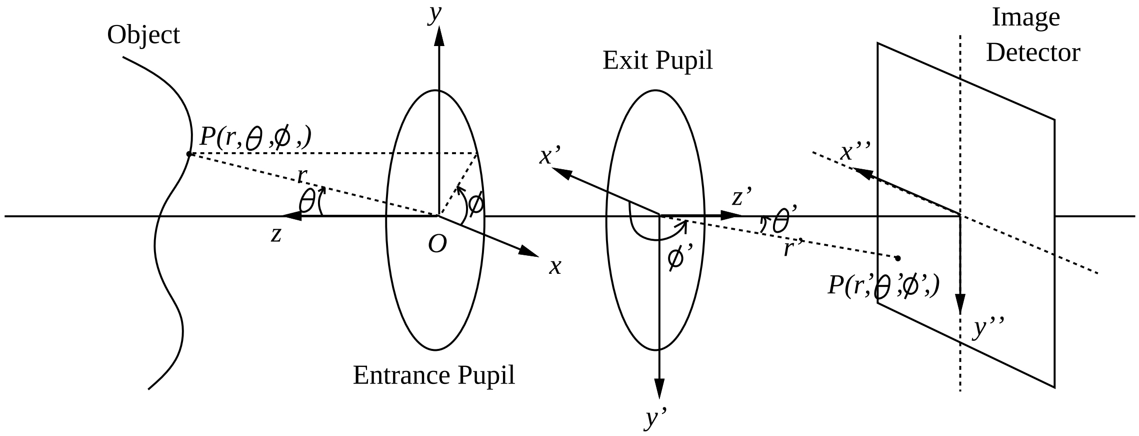


Fig. 2 Entrance pupil coordinate system.

through which light must pass as it leaves the imaging element on its way to the image detector plane. It is also assumed that the passage of light between the entrance and exit pupil planes can be adequately described by geometric optics [4].

2.1 Notations and Assumptions

In the subsequent discussions, we use a spherical coordinate system with its z-axis along the optical axis and the origin fixed in the plane of the entrance pupil as shown in Fig. 2 (called Entrance Pupil Coordinate System, or EPCS for short). All the distance measurements are made with respect to O.

We also consider the scene to contain only objects with opaque surfaces. In such case, for every direction defined by (θ, ϕ) in the EPCS, there is a *unique* point P on a visible object in the scene as shown in Fig. 2. If r is the distance from P to the origin O, then the geometry of visible surfaces in the scene can be completely defined by a function denoted by $r(\theta, \phi)$ which gives *geometric information* about visible surfaces in the scene.

What the camera "observes" due to the point source P is the electromagnetic field distribution produced by the source at the entrance pupil. This field distribution can be specified by a very general form, however, we shall restrict ourselves to a simpler case. We will only consider incoherent and unpolarized light. The quantities we are interested are the radius r of the wavefront incident on the entrance pupil and the energy strength of the wavefront as a function of its wavelength λ. These quantities are assumed to remain the same during one exposure period of the camera.

Let $f(\theta, \phi, \lambda, t)$ be the power of light of wavelength λ incident on the entrance pupil from the direction (θ, ϕ) at time t. The inputs to the camera system are the functions $r(\theta, \phi)$ which describes the geometry of the scene and $f(\theta, \phi, \lambda, t)$ which describes the appearance of the scene. They are characterized this way to decouple the photometric property $f(\theta, \phi, \lambda, t)$ and the geometric property $r(\theta, \phi)$.

2.2 Mathematical Modeling

Having defined the input to the camera as the photometric characteristic of the scene specified by $f(\theta, \phi, \lambda, t)$ and the geometry of the scene specified by $r(\theta, \phi)$, we will now define a sequence of transformations which transforms the input to the output.

2.2.1 Light Filter

Typically, camera systems have a light filter to control the spectral content of light entering the camera system. Filters that block infrared rays are widely employed. The characteristic of a light filter can be specified in terms of a transmittance function $T_{LF}(\lambda)$ where $0 \leq T_{LF}(\lambda) \leq 1.0$. Therefore, the output of this stage is

$$f_1(\theta, \phi, \lambda, t) = f(\theta, \phi, \lambda, t) \cdot T_{LF}(\lambda) \tag{1}$$

where $f(\theta, \phi, \lambda, t)$ is the input light energy defined earlier, and $f_1(\theta, \phi, \lambda, t)$ is the light energy transmitted by the filter. Usually, light filters have uniform characteristics in space and time. However, if the characteristics change with time and space, then the transmittance function takes the form $T_{LF}(\theta, \phi, \lambda, t)$.

2.2.2 Vignetting

When there are multiple apertures in the optical system along the optical axis displaced with respect to each other, the effective light energy transmitted by the system decreases with increasing inclination of light rays with respect to the optical axis [4, 5]. The effect of vignetting can be specified by a function $T_V(\theta, \phi)$ where $0 \leq T_V(\theta, \phi) \leq 1.0$. The relationship between input and output of this stage is:

$$f_2(\theta, \phi, \lambda, t) = f_1(\theta, \phi, \lambda, t) \cdot T_V(\theta, \phi) \tag{2}$$

2.2.3 Optical System

As discussed earlier, an image forming optical system can be characterized by specifying its properties at the entrance pupil and the exit pupil of the optical system. These properties determine the intensity distribution of light which in fact corresponds to the *Point Spread Function* (PSF) of the imaging system. When the point sources in the scene are *incoherent*, the intensity distribution produced on the image detector by different point sources can be simply summed to obtain the overall image. In this case, the imaging system acts as a linear system with the characteristics specified by a point spread function $h(\theta, \phi, \theta', \phi', r(\theta, \phi), \vec{e})$ where (θ, ϕ) is the direction of the point source, (θ', ϕ') is the direction of the point on the image detector in the image plane, and $\vec{e}$ is a vector specifying the parameters of the imaging system.

In order to specify the effect of the imaging system on the input, we transform the spherical coordinates (θ, ϕ) and (θ', ϕ') to Cartesian coordinates (x, y) and (x', y') in the scene space and the object space, respectively. The relations are:

$$\begin{aligned} x &= \sin\theta\ \cos\phi, & x' &= \sin\theta'\ \cos\phi', \\ y &= \sin\theta\ \sin\phi, & y' &= \sin\theta'\ \sin\phi'. \end{aligned} \tag{3}$$

Under this transformation, $f_2(\theta, \phi, \lambda, t)$ and $h(\theta, \phi, \theta', \phi', r(\theta, \phi), \vec{e})$ can be equivalently represented as $f_2'(x, y, \lambda, t)$ and $h'(x, y, x', y', r(x, y), \vec{e})$, respectively. Hence, the output of the optical system will be:

$$f_3(x', y', \lambda, t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h'(x, y, x', y', r(x, y), \vec{e})\, f_2'(x, y, \lambda, t)\, dx\, dy \tag{4}$$

If the scene and the optical system are such that the PSF is spatially invariant in the region of interest (*i.e.* isoplanetic region [2]), then the above integral becomes a convolution operation:

$$f_3(x', y', \lambda, t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h'(x' - x, y' - y, r(x, y), \vec{e})\, f_2'(x, y, \lambda, t)\, dx\, dy \tag{5}$$

2.2.4 Field Stop

Since the area of the photosensors on the image detector plane is limited in extent, only a limited part of the image plane is exposed to light. The extent of the photosensor area determines the field of view of the imaging system which can be specified by the transmittance function of the Field Stop $T_{FS}(x', y')$ where $T_{FS}(x', y')$ has a value of 1 inside the image detector region and a value of 0 outside. Therefore, the output will be:

$$f_4(x', y', \lambda, t) = f_3(x', y', \lambda, t) \cdot T_{FS}(x', y') \tag{6}$$

2.2.5 CCD Sensor

The transducer on the image detector which converts light energy to electrical energy is not uniformly sensitive with respect to the wavelength λ. To take this effect into account, we model the sensitivity by $T_s(\lambda)$ with $0 \le T_s(\lambda) \le 1.0$. This will transform f_4 to:

$$f_{5a}(x', y', \lambda, t) = f_4(x', y', \lambda, t)\, T_s(\lambda) \tag{7}$$

Since the output of the photosensor on the image detector depends on the total light energy incident on the detectors, the light energy has to be integrated with respect to the wavelength λ, *i.e.*

$$f_{5b}(x', y', t) = \int_{-\infty}^{\infty} f_{5a}(x', y', \lambda, t)\, d\lambda \tag{8}$$

The image sensor is exposed to incident light for a finite duration of time. During the period when the sensor is exposed, the strength of the incident light may vary because of the changing area of the aperture stop with time $T_{AS}(t)$. In a CCD camera, this is equivalent to measuring, periodically, the charge collected by the CCD elements and then to clear the charge in the elements. The effect of the exposure function is:

$$f_{5c}(x', y', t) = \int_{-\infty}^{\infty} f_{5b}(x', y', \tau)\, T_{AS}(\tau - t)\, d\tau \tag{9}$$

If $T_{AS}(t)$ is symmetric, then the above operation becomes a convolution operation.

Next, we take into account the physical size of the photosensor and the sensor noise. Let $R(x, y)$ be a function whose value is 1 inside the surface area of a photosensor and 0 otherwise, $n_s(x', y', t)$ be the CCD sensor noise, then we will get the following output:

$$f_{5d}(x', y', t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{5c}(\alpha, \beta, t)\, R(\alpha - x', \beta - y')\, d\alpha\, d\beta \;+\; n_s(x', y', t) \tag{10}$$

If $R(x, y)$ is again symmetric, then the first term to the right of the equal sign in the above equation is also a convolution.

We will assume that the continuous signal $f_{5d}(x', y', t)$ is sampled in time periodically at fixed intervals of τ_{s1} and sampled in space on a discrete rectangular grid of points separated by a distance of x_s along the horizontal direction and by a distance of y_s along the vertical direction as shown in Fig. 3. The resulting sampled output will be:

$$f_{5e}(x', y', t) = f_{5d}(x', y', t)\, \frac{1}{|\tau_{s1}||x_s||y_s|}\, \mathrm{comb}\left(\frac{x'}{x_s}, \frac{y'}{y_s}, \frac{t}{\tau_{s1}}\right) \tag{11}$$

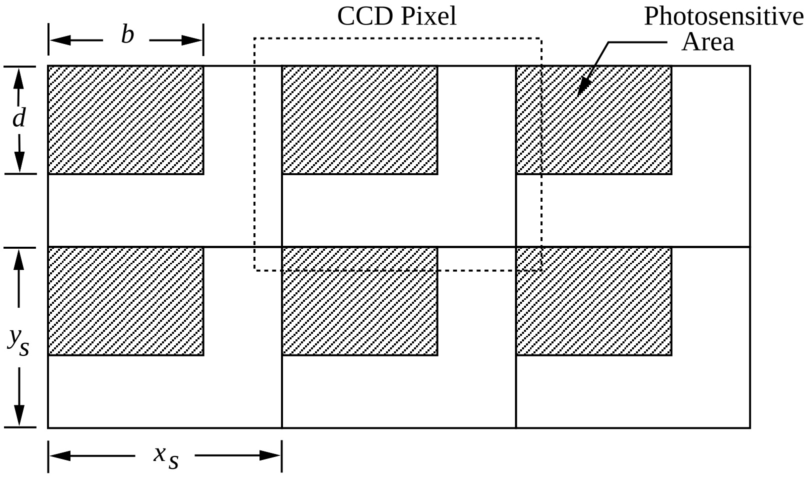


Fig. 3 Typical geometry of pixels on a CCD image detector

which is converted to a discrete function, represented as a three-dimensional matrix $f_{5f}[i,j,k]$, as:

$$f_{5f}[i,j,k] = f_{5e}(i \cdot x_s,\ j \cdot y_s,\ k \cdot \tau_{s1}) \tag{12}$$

for $i = 0, 1, \cdots, M-1$; $j = 0, 1, \cdots, N-1$; $k = 0, 1, \cdots, K-1$ where M is the number of columns, N is the number of rows, and K is the number of image frames.

Ideally, we require the output of the photosensitive elements to be proportional to the light energy incident on them. However, in practice, their output is not proportional but some other function of light energy $S(I)$. Therefore, the output will be

$$f_{5g}[i,j,k] = S(f_{5f}[i,j,k]) \tag{13}$$

Finally, the three-dimensional array is converted to one-dimensional sequence of numbers which can be converted to analog signals and transmitted over a cable. The resulting output is:

$$f_5[i + j \cdot M + k \cdot M \cdot N] = f_{5g}[i,j,k] \tag{14}$$

where we assume, for simplicity, that all rows are scanned in sequence one by one and that all the synchronization pulses for video monitor are ignored.

2.2.6 Digital-to-Analog Conversion

The discrete sequence of numbers represented by $f_5[i]$ is converted to an analog signal using a sample-and-hold circuit. The time interval τ_{s2} between two numbers is an input parameter to this stage. Note that $\tau_{s2} \leq \frac{\tau_{s1}}{M \cdot N}$. The operation of the reconstruction circuit can be thought of as interpolating the sequence of values $f_5[i]$ defined at $i \cdot \tau_{s2}$ to get the analog signal, *i.e.*,

$$f_{6a}(t) = h_{sh}(t) \star \left[\sum_{l=0}^{K \cdot M \cdot N - 1} f_5[l] \, \delta(t - l \cdot \tau_{s2}) \right] \tag{15}$$

where $h_{sh}(t)$ is the effective impulse response of the sample-and-hold interpolation circuit and $\star$ is the convolution operator.

Finally, the video signal $f_{6a}(t)$ is amplified before it is transmitted over a video cable so that any noise introduced in the cable will not dominate the signal. The resulting output signal is:

$$f_6(t) = f_{6a}(t) \star h_a(t) + n_a(t) \tag{16}$$

where $h_a(t)$ and $n_a(t)$ are the impulse response and the additive noise of the amplifier, respectively.

2.2.7 Frame Grabber

The input signal to this stage is

$$f_{7a}(t) = f_6(t) \star h_c(t) \; + \; n_c(t) \tag{17}$$

where $h_c(t)$ and $n_c(t)$ are the impulse response and additive noise of the cable, respectively. This analog signal is sampled at intervals of τ_{s3} by multiplying it with $\sum_{n=-\infty}^{\infty} \delta(t - n\tau_{s3})$. If $\tau_{s3} \neq \tau_{s2}$, this will cause geometric distortion of the picture. This phenomenon has been called *mismatched electronics* in Schafer[8]. The sampled values are interpolated by an n-th order (usually $n = 0$) sample-and-hold filter. The resulting signal is again sampled by a slightly shifted sampling function to get an impulse train which is converted to a sequence of numbers as follows:

$$
\begin{aligned}
f_{7b}(t) &= \left\{ \left[\sum_{n=-\infty}^{\infty} \delta(t - n\tau_{s3}) \right] \cdot f_{7a}(t) \right\} \star h_{sh}(t) \; + \; n_{sh}(t) & (18) \\
f_{7c}(t) &= \frac{1}{\tau_{s3}} \mathrm{comb} \left(\frac{t - k\tau_{s3}}{\tau_{s3}} \right) \cdot f_{7b}(t) & (19) \\
f_{7d}[i] &= \int_{(i+k)\tau_{s3}^{-}}^{(i+k)\tau_{s3}^{+}} f_{7c}(t)\, dt & (20)
\end{aligned}
$$

where $h_f(t)$ and $n_f(t)$ are the impulse response and additive noise of the sample-and-hold filter, respectively; $0 < k < 1$; and the following equation is used in the derivations.

$$
\sum_{n=-\infty}^{\infty} \delta(t - n\tau) = \frac{1}{\tau} \mathrm{comb} \left(\frac{t}{\tau} \right) \qquad (21)
$$

Finally, the sequence of numbers are quantized and sent to frame buffer for further processing.

3 IDS Computer Simulation System

Based on the model presented in the previous section, a simulation system called IDS (for Image Defocus Simulator) has been developed. It can simulate a typical CCD camera system in machine vision applications. Users can use IDS to verify computer vision theories. Our implementation divides IDS into many modules. Therefore it is very easy to extend the model and simulation system to cover different types of imaging systems used in machine vision.

IDS consists of a simulation engine and an user interface. The simulation engine is a machine-independent module to carry out all the computations involved while the user interface is used to provide a menu-driven I/O interface. They are discussed in the following subsections.

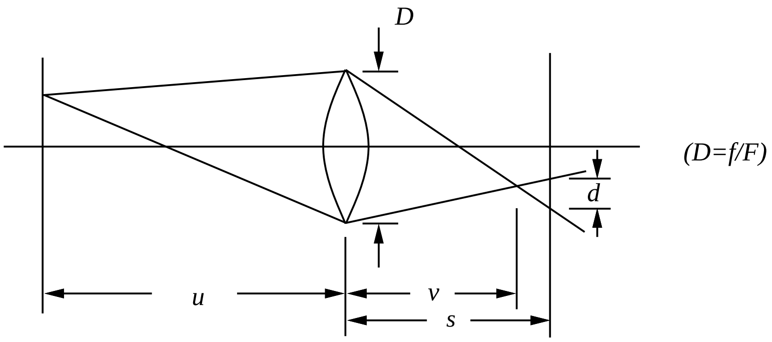


Fig. 4 Blur circle calculation

3.1 Simulation Engine

Consider the scene parameters $f(\theta, \phi, \lambda, t)$ and $r(\theta, \phi)$. If the profile of the scene in a small field-of-view is smooth, then we can approximate $r(\theta, \phi)$ by a constant u which specifies the distance between the scene and the origin O in the EPCS. Under these circumstances, the point spread function will be spatially invariant. If geometric optics is assumed, then the diameter of the blur circle can be computed by using the lens equation and the geometry as shown in Fig. 4. The resulting diameter of the blur circle is:

$$d_m = \frac{f}{v \cdot F}|s - v| \tag{22}$$

$$d_p = \frac{d_m}{\rho} \tag{23}$$

where f is the effective focal length; F is the F-number; d_m is the diameter of blur circle in millimeter; ρ is the CCD pixel element size in millimeter; d_p is the diameter of the blur circle in pixels; v is the distance between the exit pupil and the plane where the object is focused; and s is the distance between the exit pupil and the photosensor plane. If $r(\theta, \phi)$ cannot be approximated by a constant distance u, then we must calculate the blur circle for each point in the scene and do the simulation to find the image in the image plane (implementation of this case will be added to the system in the next version). If the point spread function is pre-calibrated for a particular optical system and stored in a file, then IDS can read in that point spread function without doing the above calculations. The choice is determined by the parameter "psf" which can be "**psf=cylinder**" for geometric optics or "**psf=file** `filename` [**size**] `width height`" for pre-calibrated point spread function. Therefore, one can use geometric optics as well as physical optics. Note that, in this paper we use boldface to represent keywords in parameter specifications. Keywords enclosed by square brackets are optional.

The effect of light filtering and vignetting are specified by the functions T_{LF} and T_V. They can be one of three: (i) a constant, (ii) a **gaussian**(σ_x, σ_y), or (iii) a function tabulated and stored in a file. Based on the above information, IDS can compute equations (1)-(5) where $\vec{e} = (u, f, F)$.

The parameter T_{FS} controls the field of view of the photosensor devices. For a CCD camera, the field stop is rectangular in shape with width A and height B. Therefore, we have

$$T_{FS}(x, y) = \text{rect}\left(\frac{x}{A}, \frac{y}{B}\right) \tag{24}$$

This is implemented by restricting the calculations to a rectangular region specified by the above equation.

In most cases, the exposure function can be approximated by a rectangular function $T_{AS}(t) = \text{rect}(\frac{t}{T})$ where T is the duration of exposure which is typically $\frac{1}{30}$ second. If the object is not moving, then all the timing information can be simply discarded (the implementation of moving objects will be included in the next version). Further, from the typical geometry of pixels on CCD in Fig. 3, we have $R(x, y) = rect(\frac{x}{b}, \frac{y}{d})$ for equation (10). The sensor response function $S(I)$ can be either I^γ (for standard NTSC TV, $\gamma \approx \frac{1}{2.2}$ [9]), $aI + b$ (a, b are constants), or a table read from a file.

Combining all the above information and adding 1) the impulse response functions for sample-and-hold circuit (using h_{sh}), for amplifier (using h_a), and for cable connections (using

h_c); 2) the corresponding noise functions (using n_{sh}, n_a, and n_c); 3) the CCD noise (using n_s); 4) sampling information (using τ_1, τ_2, τ_3); and 5) the CCD geometry (using x_s, y_s), we can carry out equations (6)-(20) directly to complete the simulation.

All the parameters mentioned above can be changed in "Edt Param" operation in the user interface to be discussed later. The default values for these parameters are shown in Fig. 6. Furthermore, to deal with the tremendous amount of data storage expected and the variable size of the resulting output images, a built-in dynamic memory manager is used to achieve the greatest flexibility.

As one can expect, the convolution operation is a critical part in the simulation. Consider the discrete convolution of the $M \times N$ image $f[m,n]$ and the $P \times Q$ point spread function $h[p,q]$:

$$g[\alpha,\beta] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f[m,n]h[\alpha-m,\beta-n] \tag{25}$$

where $m = 0,1,\cdots,M-1$; $n = 0,1,\cdots,N-1$; $p = 0,1,\cdots,P-1$; $q = 0,1,\cdots,Q-1$; $\alpha = 0,1,\cdots,(M+P-1)$; and $\beta = 0,1,\cdots,(N+Q-1)$. This equation can be rewritten as:

$$g[\alpha,\beta] = \sum_{m=0}^{\min\{\alpha,M-1\}}\sum_{n=0}^{\min\{\beta,N-1\}} f[m,n]h[\alpha-m,\beta-n] \tag{26}$$

In this equation, we simply assume that $h[p,q] = 0$ for $p \geq P$, $p < 0$, $q \geq Q$, or $q < 0$. The computational complexity to carry out the convolution directly will be $O((M+P) \times (N+Q) \times \min\{M,P\} \times \min\{N,Q\})$ which is a huge number if the image size is large. On the contrary, if we take the 2-D Discrete Fourier Transform:

$$\begin{aligned} F[k,l] &= \sum_{m=0}^{K-1}\sum_{n=0}^{L-1} f[m,n]W_M^{km}W_N^{ln} \qquad (27)\\ k &= 0,1,\cdots,K-1 \\ l &= 0,1,\cdots,L-1 \end{aligned}$$

as K DFTs of the form:

$$
\begin{aligned}
A[m,l] &= \sum_{n=0}^{L-1} f[m,n] W_N^{ln} \qquad (28)\\
m &= 0,1,\cdots,K-1\\
l &= 0,1,\cdots,L-1
\end{aligned}
$$

followed by L DFTs of the form:

$$
\begin{aligned}
F[k,l] &= \sum_{m=0}^{K-1} A[m,l] W_M^{km} \qquad (29)\\
k &= 0,1,\cdots,K-1\\
l &= 0,1,\cdots,L-1
\end{aligned}
$$

then the computational complexity for using FFT algorithm will be $O((M+P)\times(N+Q)\times \log_2[(M+P)\times(N+Q)])$ for an $M\times N$ image and a $P\times Q$ point spread function. Note that, for small values of P, Q, it will be more efficient to use direct convolution. Therefore, we provide three options in doing convolutions: direct, FFT, and *smart* convolution. In direct (FFT) mode, no matter how big (small) the image/psf size is, direct (FFT) convolution is carried out; while in smart mode, the type of convolution used depends on the number of operations expected. This heuristic will speed up the computation of the convolutions.

Finally, equations (28) and (29) illustrate an idea for parallelization. If the number of processors is unlimited, convolution can be done N times faster than a sequential implementation. However, in real situation, the number of processors are limited and the communications between the processors involve overheads. Therefore the ideal speed up cannot be attained. Parallel implementation of IDS is currently under investigation.

3.2 User Interface

IDS has a menu-driven user interface. Users have full access/control to the internal parameters via a popup window. The output at each stage can also be displayed on the monitor for further analysis and/or evaluation.

Fig. 5 Startup screen of the simulation system

When the simulation system starts execution, users are provided with seven operations and three choices as shown in Fig. 5. All the I/O operations and the “Read Parameters” operation take the string in the “Filename” field as the target filename. The operation “Read Parameters” reads a file which contains all the user-controllable parameters. These parameters are parsed through a built-in LR(1) parser [1] to generate tokens and detect possible syntax errors. The parsing results are then interpreted by an interpreter to give proper parameter values. These parameters can be modified by the “Edt Param” operation which gives a popup window as shown in Fig. 6.

The “Options” operation controls the system-wide options such as input/output image format, convolution method, and how to handle the image border. The input/output image format can be binary/ASCII integers, binary/ASCII floating numbers, or even a n-th order polynomial (input only) specified by the order of the polynomial and the corresponding

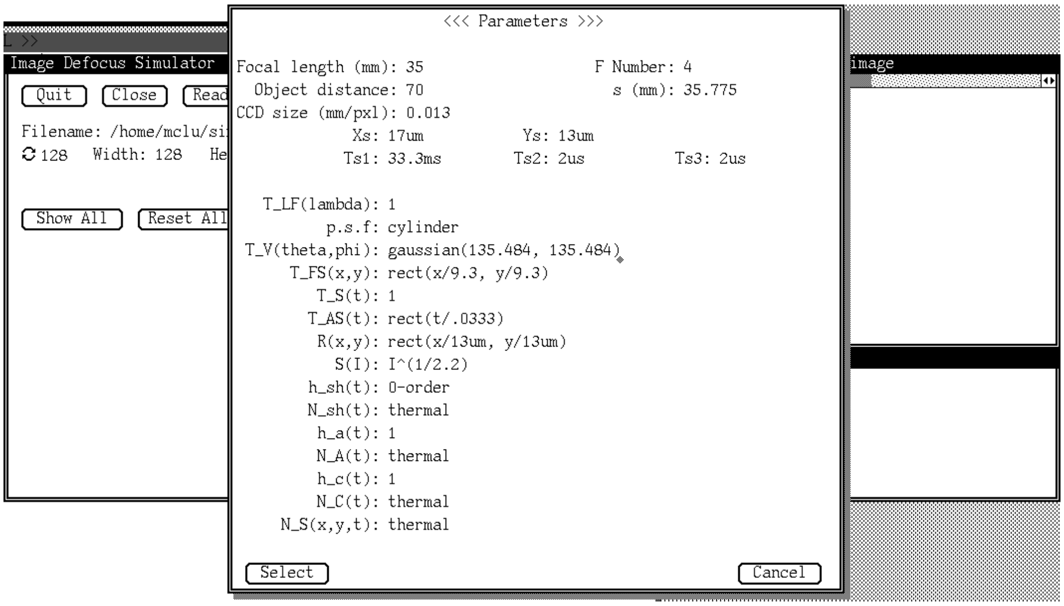


Fig. 6 Default parameter values

coefficients. The convolution method can be direct, FFT, or *smart* mode implementation as described in the previous subsection. The image border can be treated as a zero-padded, mirrored, or periodic image during convolution. For "zero-padded" option, the image outside the field of view is simply treated as a dark area, *i.e.*, an area with zero values. For "periodic" option, the image is considered as $f(aM+m, bN+n) = f(m,n)$ for an $M \times N$ image where $a, b \in \{0, \pm 1, \pm 2, \cdots\}$. And for "mirrored" option, the image is first reflected along its right side border, then the resulting image is reflected along its top border; this gives an image which is four times larger than the original image. This four times larger image is then taken to be wrapped around at its borders. Note that, the "mirrored" option gives a periodic image whose period is twice that of the "periodic" option. After the input image and the parameters are loaded, additional operations are available as shown in Fig. 7.

The "Run", "Step", and "Goto" operations control the execution of the simulation. Users can step or go to any particular module in Fig. 1 to examine the output of a specific module. The value of the pixels of the image can be viewed via "View Value" operation. We also

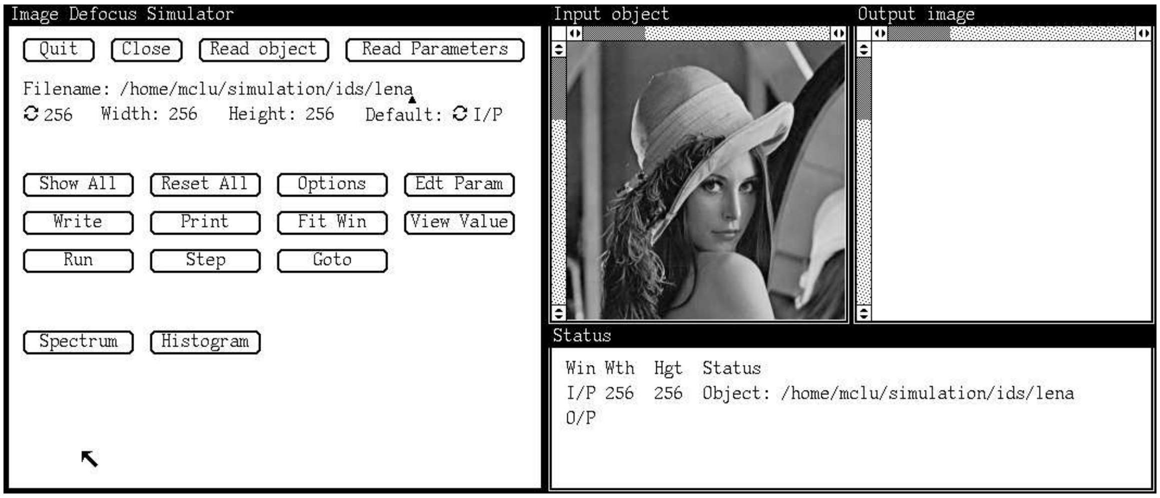


Fig. 7 Full operation menu

provide Fourier spectrum and histogram analysis of a given image by using the "Spectrum" and the "Histogram" operations as shown in Fig. 8. During these two operations, a popup status window is displayed to keep track of the status of each spectrum or histogram popup window such as — when the operation was called and where the source image came from. The magnitude of the spectrum can also be displayed and inspected by simply pressing a mouse button.

Finally, the synthesized image can be saved to a file and/or sent to printer for printing using a halftone algorithm. All the images shown in section 4 are printed in this way.

3.3 Portability

The IDS simulation engine is written in ANSI C which is portable to almost any machine with a C compiler in it. The user interface is implemented for Sunview environment on SUN workstation which is machine-dependent. However, we are planning to port this user interface to X window environment using the primitive X operations to reduce the machine-dependency. Therefore, everyone can use the developed simulation system to con-

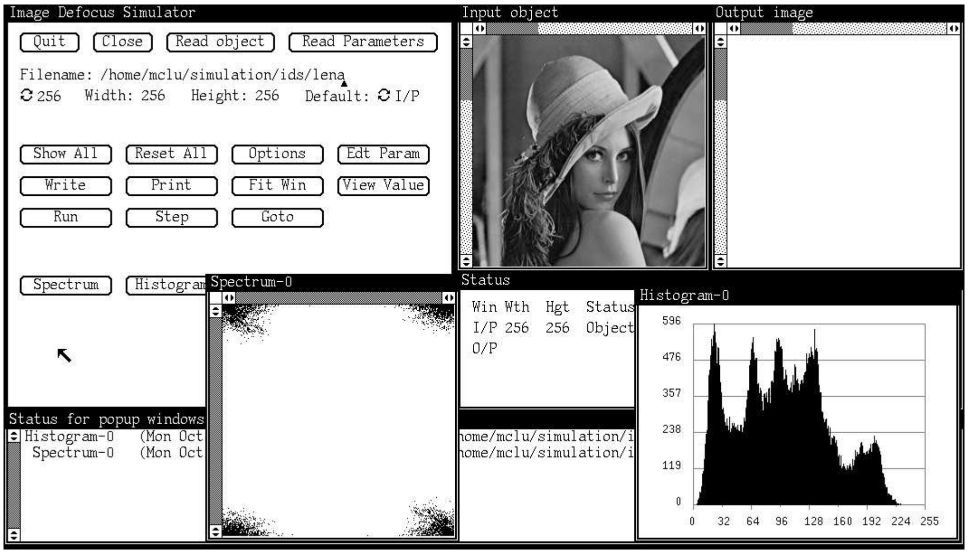


Fig. 8 Spectrum and histogram analysis

duct experiments on machine vision theories without putting too much effort in buying and/or building the expensive camera systems.

4 Examples of Simulated Pictures

In this section, two examples of simulated pictures are presented: one for the image *Lena*, the other for a generated pattern $f(x,y) = 125 * cos2\pi(2x+3y) + 125$. The default parameters are used except for the parameters s and u. The original images and the simulation results are shown in Fig. 9 and Fig. 10-11, respectively. In Fig. 11, we have $A = B = 125$, $\vec{w} = (2,3)$, and $\vec{r} = (x,y)$.

Fig. 10 shows 16 pictures of *Lena* arranged in 4 rows and 4 columns. The distance of the object increases (350mm-8943mm) row-wise from top to bottom whereas the distance s between the lens and the image detector increases (34mm-37mm) column-wise from left to right. The focal length and the F-number are fixed at 35mm and 4 respectively. In Fig. 10, for convolution operation, the "mirror" option mentioned earlier has been used. Fig. 11 is

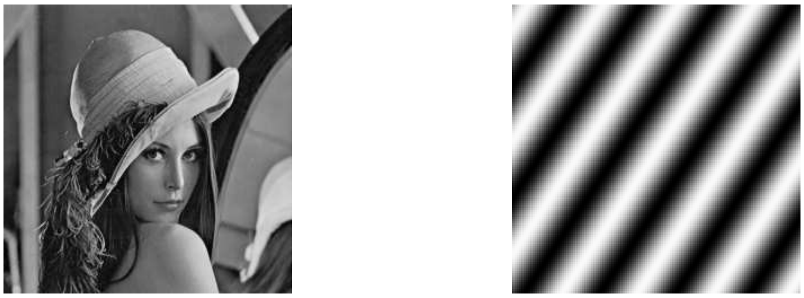


Fig. 9 Original images used in the simulation

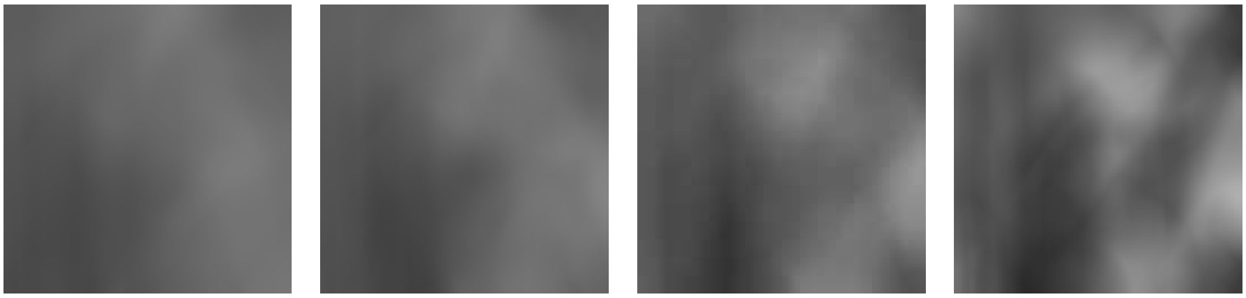
that of a cosine wave. It is similar to Fig. 10, except that convolution is carried out with the "periodic" option. Because of this, some artifacts are visible at the border.

In Fig. 10 and 11, somewhere along a direction parallel to the left-bottom to top-right diagonal, the pictures are focused, whereas, on either side of the this, the image defocus increases. This is consistent with the fact that image defocus should increase when either the object is moved farther or closer from its focused position, or when the image detector is moved farther or closer from its focused position.

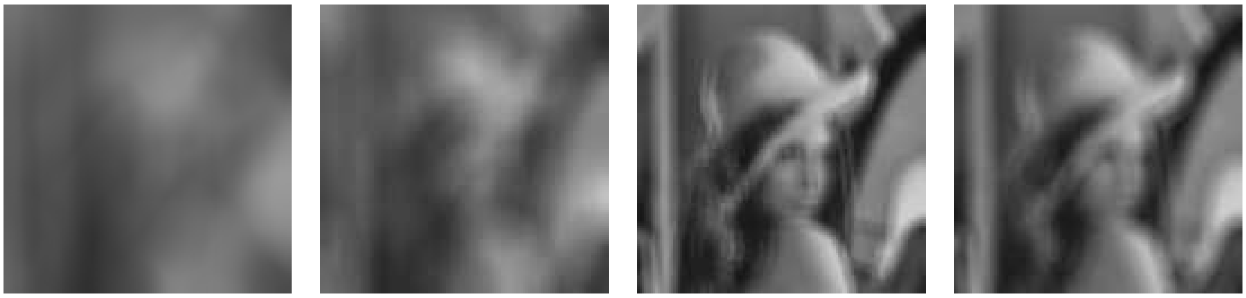
IDS has been used to generate a large number of defocused images in our laboratory. We are at present doing research on two different methods of finding distance of objects [10, 11] from their defocused images. For both these methods, IDS has been a very valuable tool in testing the methods and debugging their implementations. We are also using IDS to test new methods of image restoration for obtaining focused images from their blurred images.

5 Conclusion

A mathematical model has been developed for camera defocus related to computer vision applications. Based on this model, we have developed a computer simulation system which can help users to simulate the formation and sensing of defocused images without using expensive camera systems. Since the simulation system inherently requires a large amount



(a) $s = 34$ $u = 350$ (b) $s = 35$ $u = 350$ (c) $s = 36$ $u = 350$ (d) $s = 37$ $u = 350$

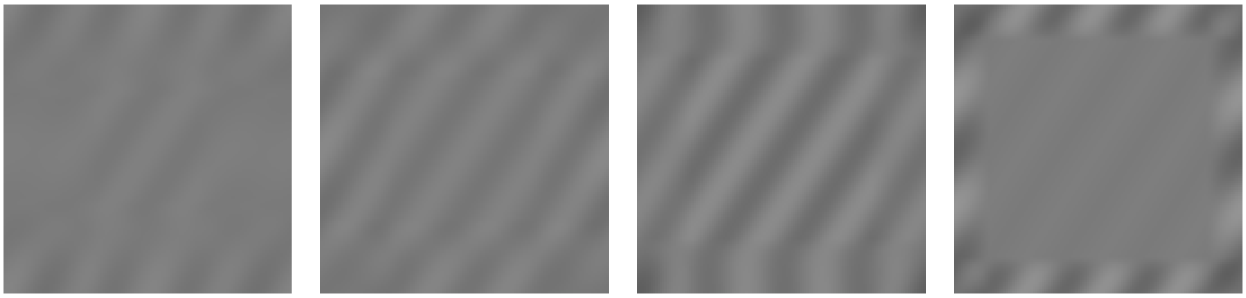


(e) $s = 34$ $u = 910$ (f) $s = 35$ $u = 910$ (g) $s = 36$ $u = 910$ (h) $s = 37$ $u = 910$

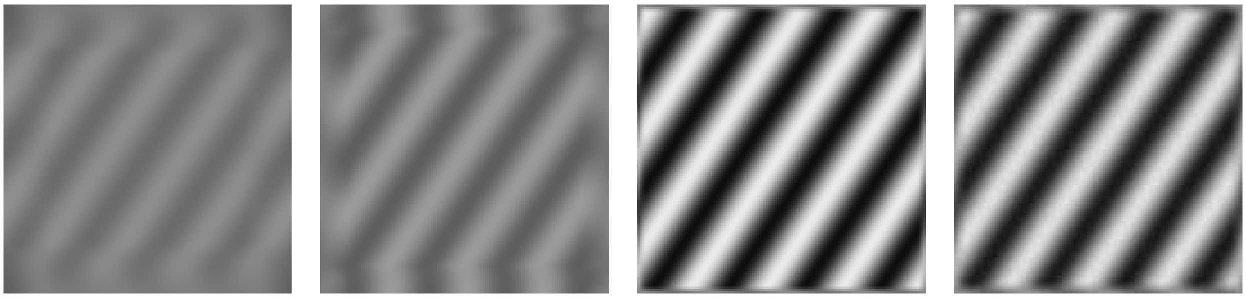
(i) $s = 34$ $u = 1700$ (j) $s = 35$ $u = 1700$ (k) $s = 36$ $u = 1700$ (l) $s = 37$ $u = 1700$

(m) $s = 34$ $u = 8943$ (n) $s = 35$ $u = 8943$ (o) $s = 36$ $u = 8943$ (p) $s = 37$ $u = 8943$

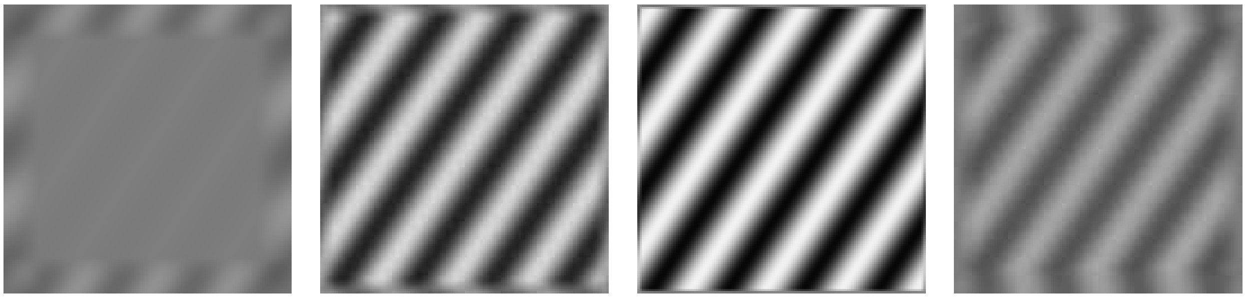
Fig. 10 Simulated pictures for *Lena*



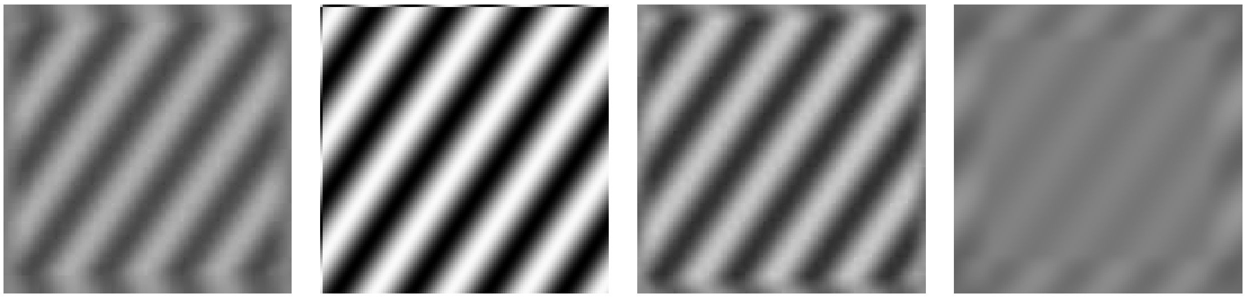
(a) $s = 34$ $u = 350$ (b) $s = 35$ $u = 350$ (c) $s = 36$ $u = 350$ (d) $s = 37$ $u = 350$



(e) $s = 34$ $u = 910$ (f) $s = 35$ $u = 910$ (g) $s = 36$ $u = 910$ (h) $s = 37$ $u = 910$



(i) $s = 34$ $u = 1700$ (j) $s = 35$ $u = 1700$ (k) $s = 36$ $u = 1700$ (l) $s = 37$ $u = 1700$



(m) $s = 34$ $u = 8943$ (n) $s = 35$ $u = 8943$ (o) $s = 36$ $u = 8943$ (p) $s = 37$ $u = 8943$

Fig. 11 Simulated pictures for $f(\vec{r}) = Acos2\pi(\vec{w_0} \cdot \vec{r}) + B$

of computation time in doing convolutions, parallelization of the system is being investigated.

Acknowledgements: The support of this research by the National Science Foundation and the Olympus Optical Corporation is gratefully acknowledged.

References

[1] A. V. Aho, R. Sethi, and J. D. Ullman, *Compilers – Principles, Techniques, and Tools,* Addison-Wesley, Massachusetts, 1986.

[2] M. Born and E. Wolf, *Principles of Optics,* Pergamon Press, Oxford, Sixth edition, 1980.

[3] J. W. Goodman, *Introduction to Fourier Optics,* McGraw-Hill Book Co., 1968.

[4] E. Hecht, *Optics,* Addison-Wesley Publishing Co., 1987.

[5] B. K. P. Horn, *Robot Vision,* McGraw-Hill Book Co., 1986.

[6] M. D. Levine, *Vision in Man and Machine,* McGraw-Hill Book Co., 1985.

[7] A. Rosenfeld and A. C. Kak, *Digital Picture Processing,* Vol. 1, Academic Press, 1982.

[8] S. A. Shafer, "Automation and Calibration for Robot Vision Systems," Technical Report No. CMU-CS-88-147, Computer Science Department, Carnegie Mellon University, 1988.

[9] W. N. Sproson, *Colour Science in Television and Display Systems,* Adam Hilger Ltd., Bristol, 1983.

[10] M. Subbarao, "Efficient Depth Recovery Through Inverse Optics," in H. Freeman (Eds.)., *Machine Vision for Inspection and Measurement,* Academic Press, Boston, pp. 101-126, 1988.

[11] M. Subbarao, "Determining Distance From Defocused Images of Simple Objects," Technical Report No. 89.07.20, Computer Vision Laboratory, Department of Electrical Engineering, State University of New York, Stony Brook, 1989.

[12] M. Subbarao and A. Nikzad, "Model for Image Sensing and Digitization in Computer Vision", Proceedings of SPIE conference, OE/90, Boston, Nov. 1990.