Continuous Focusing of Moving Objects using Image Defocus

Gopal Surya Murali Subbarao

Department of Electrical Engineering
State University of New York, Stony Brook, NY 11794-2350
email: gopal@sbee.sunysb.edu murali@sbee.sunysb.edu

ABSTRACT

A Depth-from-Defocus method named STM[11–13] was presented recently for stationary objects. Here we extend STM for continuous focusing of moving objects. The method is named Continuous STM or CSTM. Focusing is done by moving the lens with respect to the image detector. Two variations of CSTM - CSTM1 and CSTM2 - are presented. CSTM1 is a straight forward extension of STM described in.[12] It involves calibration of the camera for a number (about 6 in our implementation) of discrete lens positions. In CSTM2 the camera is calibrated only for one lens position. The calibration data corresponding to other lens positions are obtained by transforming the data of the one lens position for which the camera is calibrated. In the experimental results presented here, the focusing error in lens position was about 2.25% for CSTM1 and about 3% for CSTM2.

1 Introduction

We recently proposed a Depth-from-Defocus algorithm[11–13] using a new Spatial Domain Convolution/ Deconvolution Transform (S-Transform).[10] This method, named S-Transform Method or STM, uses only two images taken with different camera parameters to estimate the distance of an object. All computations are done in the spatial domain and are local in nature. Hence this method can yield the scene depth map and can be implemented in parallel. STM has been implemented on a prototype camera system named Stonybrook Passive Autofocusing and Ranging Camera System or SPARCS. A large number of experiments (about 600) have yielded an RMS error of about 2.3% in autofocusing application. Two variations of STM are described in,[12] one where the lens position and focal length are changed and another where the diameter of camera aperture is changed.

In this paper, we address the problem of continuously focusing the camera on a moving object by changing the lens position. Such a situation may arise in an autofocusing video camera and in robotic vision, where the objects in the scene may be slowly moving. Wei and Subbarao[14] have recently proposed a Fourier domain approach for focusing on moving objects. They reported a focusing accuracy of about 4.3%. Here, we describe a spatial domain method based on STM. The method is named Continuous STM or CSTM. The focusing accuracy of CSTM is about 2.3 - 3%. Using CSTM it is also possible to obtain denser depth maps of the scene, than what can be obtained by using Fourier domain methods such as.[14]

Two variations of CSTM - CSTM1 and CSTM2 - are presented. CSTM1 is a straight forward extension of the STM described in.[12] It involves calibration of the camera for a number (about 6 in our implementation) of discrete lens positions. In CSTM2 the camera is calibrated just once corresponding to one lens position. The calibration data corresponding to other positions are obtained by transforming the data of the one lens position for which the camera is calibrated. Experiments show that the difference in performance between CSTM1 and CSTM2 is marginal.

2 Camera model

A schematic diagram of a camera system with variable camera parameters is shown in Figure 1. It consists of an optical system with two lenses L1 and L2. The effective focal length f is varied by moving one lens with respect to the other. O.A. is the optical axis, P1 and P2 are the principal planes, Q1 and Q2 are the principal points, ID is the image detector, D is the aperture diameter, s is the distance between the second principal plane and the image detector, u is the distance of the object from the first principal plane, and v is the distance of the focused image from the second principal plane.

The distance s, focal length f, and the aperture diameter D, will be referred together as *camera parameters* and denoted by $\mathbf{e}$, i.e.,

$$\mathbf{e} = (s, f, D). \tag{1}$$

In order to illustrate the theoretical basis of CSTM we take the optical system to be circularly symmetric around the optical axis, and use a paraxial geometric optics model for image formation. This is a good approximation in practice to actual image formation process modeled by physical optics.[1] However, CSTM itself is applicable to physical optics model also.

In Figure 1, u denotes the object distance and v denotes the distance of the focused image. These quantities are related to the focal length f by the well-known lens formula,

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v}. \tag{2}$$

In Figure 1, if the object point p is not in focus, then it gives rise to a blurred image p'' on the image detector ID. According to geometric optics, the blurred image of p has the same shape as the lens aperture but scaled by a factor. This holds irrespective of the position of p on the object plane. Since we have taken the aperture to be circular, the blurred image of p is also a circle with uniform brightness inside the circle and zero outside. This is called a *blur circle*.

Let the light energy incident on the lens from the point p during one exposure period of the camera be one unit. Then, the blurred image of p is the response of the camera to a unit point source and hence it is the Point Spread Function (PSF) of the camera system. This PSF will be denoted by $h(x, y)$.

Let R be the radius of the blur circle and q be the scaling factor defined as $q = 2R/D$. In Figure 1, from similar triangles and from the lens formula (2) we obtain

$$R = q\frac{D}{2} = s\frac{D}{2}\left[\frac{1}{f} - \frac{1}{u} - \frac{1}{s}\right] \tag{3}$$

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. After magnification normalization, the normalized radius $R' = s_0 R/s$ of the blur circle can be expressed as a function of the camera parameter setting and object distance u as

$$R' = \frac{Ds_0}{2}\left[\frac{1}{f} - \frac{1}{u} - \frac{1}{s}\right]. \tag{4}$$

If we assume the camera to be a lossless system (i.e., no light energy is absorbed by the camera system) then

$$\int\int h(x,y)\ dx\ dy \ = \ 1 \tag{5}$$

because the light energy incident on the lens was taken to be one unit. Using this and the fact that the blur circle has uniform brightness inside a circle of radius R' and zero outside, we obtain the PSF to be a cylindrical function:

$$h_1(x,y) = \begin{cases} \frac{1}{\pi R'^2} & \text{if } x^2+y^2 \leq R'^2 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where h_1 is the PSF according to paraxial geometric optics.

In practice, the image of a point object is not a crisp circular patch of constant brightness as suggested by geometric optics. Instead, due to diffraction, poly-chromatic illumination, lens aberrations, etc., it will be a roughly circular blob with the brightness falling off gradually at the border rather than sharply. Therefore, as an alternative to the above cylindrical PSF model, often[6] a two-dimensional Gaussian is suggested which is defined by

$$h_2(x,y) \ = \ \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{7}$$

where σ is a spread parameter corresponding to the *standard deviation* of the distribution of the PSF. In practice, it is found that σ is proportional to R', i.e. $\sigma \ = \ k\ R'$ for $k > 0$ where k is a constant of proportionality characteristic of the given camera. Except when σ is very small (in which case diffraction effects dominate), in most practical cases $k = \frac{1}{\sqrt{2}}$ is a good approximation.[8] Since the blur circle radius R' is a function of $\mathbf{e}$ and u, σ can be written as $\sigma(\mathbf{e},u)$. (However, the image of an actual point light source for our camera was quite close to a cylindrical function and was far from a Gaussian.)

If the radius R' is a constant over some region on the image plane, the camera acts as a linear shift invariant system. 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., $g(x,y) = h(x,y) * f(x,y)$ 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 where σ_h is the standard deviation of the distribution of any function h. It can be defined as the square root of the second central moment of the function h. For a rotationally symmetric function it is given by

$$\sigma_h^2 = \int\int (x^2+y^2) h(x,y)\ dxdy \tag{8}$$

Using the polar co-ordinate system it can be shown[8] that the spread parameter σ_{h_1} corresponding to h_1 is $R'/\sqrt{2}$. Therefore from equation (4) we have

$$\sigma_{h_1} = mu^{-1} + c \ \text{ where } \ m = -\frac{Ds_0}{2\sqrt{2}} \ \text{ and } \ c = \frac{Ds_0}{2\sqrt{2}}\left[\frac{1}{f} - \frac{1}{s}\right] \tag{9}$$

We see that for a given camera setting (i.e., for a given value of the camera parameters s, f, D) the spread parameter σ_{h_1} depends linearly on inverse distance u^{-1}. Similarly it can be shown that the spread parameter σ_{h_2} of h_2 is σ. Therefore from equation (4) we again obtain $\sigma_{h_2} = mu^{-1} + c$.

3 Distance of Moving Objects (CSTM)

In this section we present the outline of the theoretical basis for determining distance. Details can be found in.[12] Let $f(x,y)$ be the focused image of a planar object at distance u. The *focused image* $f(x,y)$ at a point

(x, y) of a scene is defined as the total light energy incident on the camera aperture (entrance pupil) during one exposure period from the object point along the direction corresponding to (x, y).

Let $g_1(x, y)$ and $g_2(x, y)$ be two images of the object recorded for two different camera parameter settings $\mathbf{e}_1$ and $\mathbf{e}_2$ where

$$\mathbf{e_1} \ = \ (s_1, f_1, D_1) \quad \text{and} \quad \mathbf{e_2} \ = \ (s_2, f_2, D_2). \tag{10}$$

The images g_1 and g_2 are normalized with respect to magnification, brightness, and other factors such as sensor response and vignetting as necessary.

For a planar object perpendicular to the optical axis, the blur circle radius R' is a constant over the image of the object (this may not be obvious at first sight, but it can be proved easily). In this case the camera acts as a linear shift invariant system. Therefore g_i will be equal to the convolution of the focused image $f(x, y)$ with the corresponding point spread function $h_i(x, y)$. In brief this can be expressed by $g_1 = h_1 * f$ and $g_2 = h_2 * f$. Let the spread parameter σ_h for h_1 be σ_1 and for h_2 be σ_2.

Now from equation (9) we can write

$$\sigma_1 = m_1 u^{-1} + c_1 \tag{11}$$

where

$$m_1 = -\frac{D_1 s_0}{2\sqrt{2}} \text{ and } c_1 = \frac{D_1 s_0}{2\sqrt{2}} \left[\frac{1}{f_1} - \frac{1}{s_1}\right]. \tag{12}$$

Similarly we obtain

$$\sigma_2 = m_2 u^{-1} + c_2 \tag{13}$$

Therefore, σ_1 can then be expressed in terms of σ_2 as

$$\sigma_1 = \alpha\sigma_2 + \beta, \quad \text{where } \alpha = \frac{m_1}{m_2} \text{ and } \beta = c_1 - c_2\frac{m_1}{m_2}. \tag{14}$$

We assume that in a small image neighborhood the focused image $f(x, y)$ can be adequately approximated by a cubic polynomial in (x, y). This assumption has been relaxed in the implementation, using smoothed differentiation filters as discussed in.[12] In our application, the image neighborhood is of size 9×9 pixels. Using the S-Transform the following deconvolution expressions have been derived in.[12]

$$f = g_1 - \frac{1}{4}\,\sigma_1^2\ \nabla^2 g_1 \ \text{ and } \ f = g_2 - \frac{1}{4}\,\sigma_2^2\ \nabla^2 g_2 \tag{15}$$

In the above two relations, the dependence of all functions on (x, y) is understood but has been dropped from notation only for convenience. It can easily be shown that for a cubic polynomial, $\nabla^2 g_1 = \nabla^2 g_2$. Defining $g = (g_1 + g_2)/2$ and equating the right hand sides of the equations in (15) and squaring first and then integrating over a small region around the point (x, y) we get

$$\int\!\!\int (g_1 - g_2)^2 dx dy = \frac{1}{16}(\sigma_1^2 - \sigma_2^2)^2 \int\!\!\int (\nabla^2 g)^2\ dx dy \tag{16}$$

which can be expressed as

$$(\sigma_1^2 - \sigma_2^2)^2 = G^2 \tag{17}$$

$$\text{where } G^2 = 16\ \frac{\int\int (g_1 - g_2)^2 dx dy}{\int\int (\nabla^2 g)^2\ dx dy} \quad \Rightarrow (\sigma_1^2 - \sigma_2^2) = G' \tag{18}$$

where $G' = \pm G$. The sign of G' is ambiguous, but this ambiguity is not inherent. It was introduced by the squaring of equation (16). The ambiguity can be resolved from the given images g_1 and g_2 in one of several ways. As one example, if g_1 is more blurred than g_2 then $\sigma_1^2 > \sigma_2^2$ and therefore the sign is positive, otherwise the sign is negative. It is easy to determine which of g_1 and g_2 is more blurred. From the theory on Depth-from-Focus methods it is well-known that the gray-level variance of an image is a good measure of the degree of focus of the image. Therefore, if v_1 , v_2 are the gray-level variances of g_1, g_2 respectively, then the sign is positive if $v_1 < v_2$ and negative otherwise. Therefore

$$G' = \begin{cases} +G & \text{if } v_1 < v_2 \\ -G & \text{otherwise} \end{cases}$$

Now substituting for σ_1 in terms of σ_2 using equation (14) into equation (18) yields

$$\sigma_2^2(\alpha^2 - 1) + 2\alpha\beta\sigma_2 + \beta^2 = G' \tag{19}$$

The above equation can be solved as a quadratic in σ_2.

In our experiments, $D_1 = D_2$ and therefore $\alpha = 1.0$. In this case the above quadratic equation in σ_2 reduces to a linear equation. Therefore we get the unique solution:

$$\sigma_2 = \frac{G' - \beta^2}{2\beta} \tag{20}$$

Ideally it should be possible to compute the value of σ_2 at one pixel (x, y) in the image and obtain an estimate of the distance. But because of noise and digitization, it is necessary to combine information from many pixels in an image region. σ_2 is computed at each pixel in a neighborhood of size 48×48 and a histogram of the values is obtained. The histogram is smoothed by a Parzen window and the mode of the resulting distribution is taken to be the best estimate of σ_2. Once σ_2 is determined the object distance u can be obtained using a look-up table or calculated from equation (13). The distance u can then be substituted into the lens formula to obtain v. Moving the lens such that $s = v$ in Figure 1 results in autofocusing the object. In our experiments, the lens position v for focusing the object was determined using a look-up table with σ_2 as the index to the table. The look-up table itself was created by a calibration process. This approach was found to be more accurate than using equation (13) and the lens formula.

Until now what we have described is the STM for determining distance and autofocusing of a stationary object. In this case the two images g_1 and g_2 are recorded at lens positions S_1 and $S_1 + \Delta S$.

In the case of a moving object, the two images g_1 and g_2 will have to be recorded at the same instant. They cannot be recorded in sequence one after the other because the object would have moved during the time it takes to move the lens from one position to the other. Figure 5 shows a scheme for recording g_1 and g_2 simultaneously using a beam splitter and two image detectors ID1 and ID2. The effective distance of ID1 and ID2 from the lens are set to be S_1 and $S_2 = S_1 + \Delta S$.

Let ID1 be the image detector in Figure 5 on which we want to continuously record the focused image of a moving object. When the lens is moved along the optical axis to focus the moving object, both S_1 and S_2 change but $S_2 - S_1 = \Delta S$ remains the same. CSTM1 is based on this camera architecture. CSTM1 is a direct extension of STM applied to moving objects using a camera system which is similar to the one shown in Figure 5. In general, look-up tables have to be used for better accuracy in focusing. In the case of CSTM1, look-up tables have to be obtained for a number of discrete values of S_1. The data at other values of S_1 are obtained through interpolation. Therefore, CSTM1 involves camera calibration for a number of lens positions.

CSTM2 is based on the following observation. It can be shown that if we take two images g_1 and g_2 keeping the aperture diameter constant ($D_1 = D_2$), then $\alpha = 1.0$ and $\beta = c_1 - c_2$. So equation (14) becomes

$$\sigma_1 = \sigma_2 + c_1 - c_2. \tag{21}$$

Therefore, if we know $\sigma = \sigma_2$ for a lens position S_2, then we can compute $\sigma = \sigma_1$ for any other lens position S_1, by adding a known constant $c_1 - c_2$. Consequently, if a look-up table is available to find the lens focused position using σ_2, then the same look-up table can be transformed to obtain the lens focused position using σ_1.

The validity of equation (21) can be verified experimentally. Figure 4 shows calibration data for several lens positions. The X-axis denotes object distances specified in lens step number and Y-axis is the blur parameter σ (see next section for more details). It is found that the data for different lens positions are roughly shifted versions of the same curve. Equation (21) however is not exact, but only a good approximation for actual camera systems because our PSF model is not exact.

In the next section we describe the SPARCS system and then discuss in detail the implementation of this algorithm. We also give experimental results on real world objects.

4 Implementation

4.1 SPARCS

CSTM described above for moving objects was implemented on a camera system named Stonybrook Passive Autofocusing and Ranging Camera System (SPARCS). A block diagram of the system is shown in Figure 2. SPARCS consists of a SONY XC-77 CCD camera and an Olympus 35-70 mm motorized lens. Images from the camera are captured by a frame grabber board (Quickcapture DT2953 of Data Translation). The frame grabber board resides in an IBM PS/2 (model 70) personal computer. The captured images are processed in the PS/2 computer.

The lens system consists of multiple lenses and focusing is done by moving the front lens forward and backward. The lens can be moved either manually or under computer control. To facilitate computer control of the lens movement there is a stepper motor with 97 steps, numbered 0 to 96. Step number 0 corresponds to focusing an object at distance infinity and step number 96 corresponds to focusing a nearby object, at a distance of about 55cm from the lens. The motor is controlled by a microprocessor, which can communicate with the IBM PS/2 through a digital I/O board (Contec mPIO24/24). Pictures taken by the camera can be displayed in real time on a color monitor (SONY PVM-1342 Q). The images acquired and stored in the IBM PS/2 can be transferred to a SUN workstation. In effect, the system is set up such that, a C program running on the PS/2 can move the lens to any desired step number and take pictures and process them.

Figure 3 shows a plot of the lens step number (the first column) along the x-axis and the reciprocal of best focused distance $1/D_0$ along the y-axis. This data was obtained from the lens manufacturer. The plot indicates that the lens step number and the reciprocal of best focused distance have an almost linear relationship. This is in fact predicted by the lens formula. Based on this relationship, we often find it convenient to specify distances of objects in terms of lens step number rather than in units of length such as meter. For example, when the "distance" of an object is specified as step number n, it means that the object is at such a distance D_0 that it would be in best focus when the lens is moved to step number n. The precise relationship between n and D_0 is given by Figure 3.

4.2 CSTM for moving objects

The overall operation of SPARCS for finding distance and autofocusing of a moving object is summarized as a flow-chart in Figure 6. The stepwise operation is also explained briefly with comments below. In the experiments, initially, the zoom setting of the lens was set to be 35 mm focal length and the F-number was set to be 4. The camera gain was set to +6db.

The lens is first moved to S_1 and a first image $g_1(x, y)$ is obtained. Optionally we can specify the number of image frames (typically 4) to be recorded which are then averaged to reduce noise. Such frame averaging is particularly needed under low illuminations, and in the presence of flickering illumination such as fluorescent lamps. This was clearly evident from a number of tests on SPARCS.

The lens is then moved to S_2 and a second image $g_2(x, y)$ is recorded. Again several frames may be recorded and averaged. The object to be ranged/focused can be selected by specifying a region in the image. The default region is the center of the image. The size of the region is also an option and the default size is 72×72. The two images are then normalized with respect to brightness. This is done by dividing the grey level of each pixel by the mean grey level of the entire image. Our implementation does not normalize the images with respect to other types of distortions such as vignetting and sensor response characteristics, as their effects are not significant for our camera. We have also ignored the magnification normalization, as the change in magnification due to change in lens position was found to be negligible (about 2%).

The images are then smoothed using the least-squares polynomial fit filters proposed by Meer and Weiss.[5] The filter size is 9×9. The Laplacian of the two smoothed images are then obtained using the differentiation filters of Meer and Weiss.[5]

The sign of G' is found by computing the gray-level variances of the original (unsmoothed) images g_1 and g_2. G^2 is calculated at every pixel by integrating over a 9×9 window centered at the pixel. G' is then calculated at every pixel. The value of the camera constants α and β are calculated from a knowledge of the camera parameters (see Table 1). An estimate of σ_2 is then obtained at every pixel using equation (20). Due to border effects of smoothing filter and integration, the estimates of σ_2 is limited to the interior 48×48 region of the original 72×72 images. A histogram of the estimated σ_2 is computed. The bin size of the histogram was 0.1 (the expected range of σ_2 was from about -10.0 to +10.0 pixels). The histogram was smoothed using a Parzen Window of size 5 bins. The mode of the histogram was taken to be the best estimate of σ_2. This value is used to estimate the distance of the object. In autofocusing application, from σ_2, the lens step number which will bring the object to focus is determined. The lens is then moved to this step number to accomplish autofocusing.

In obtaining the object distance or lens step number for focusing from the computed value of σ_2, a look-up table is used. The look-up table itself is obtained through calibration as described in.[12]

Suppose the result of the first trial is S_f (step number n). After the first iteration, the lens is moved to S_f (step number n) to focus on the object. Since the object would have moved in the mean time, we again take two images at lens positions $S_1 = S_f$ and $S_2 = S_f + \Delta S$ (steps n and $n + 30$ in our implementation). In a camera system as shown in Figure 5, it is not necessary to move the lens. Both the images can be obtained simultaneously. Hence the time between two iterations can be very small. The entire procedure is repeated with the two new images. Every time, a new focus position S_f is calculated using the appropriate calibration table (CSTM1) or by using the same calibration table, but shifted versions of it (CSTM2). The amount by which the calibration table has to be shifted depends on the previous focusing position. Everytime a new result is obtained, two new images are taken and the entire procedure is repeated in a loop. This ensures that the object is always in focus (on ID1) whether the object is stationary or moving.

4.3 Experiments

We used five different planar objects in our experiments (Figure 7). A center region of 72 x 72 pixels, that is usually used for computation is highlighted, in the Tiger image. After filtering, the useful region for computation of σ will be only 48 x 48 pixels. We have tried our experiments with even smaller regions upto 32 x 32 pixels and obtained satisfactory results, as long as there is some contrast in that region.

For CSTM1 we calibrated the system at 6 different lens positions, namely steps 40, 50, 60, 70, 80 and 90. The calibration results are plotted in Figure 4, where the X-axis is the object distance in step number and Y-axis is the value of sigma. In the Figure, the plot "cal10.dat", was obtained using images recorded at lens steps of 10 and 40, the plot "cal20.dat" was obtained corresponding to lens steps of 20 and 50 and so on. The plots can be seen to be more or less shifted versions of each other. We assumed that the calibration characteristics do not change much in a 10 step interval and hence the choice of these 6 lens positions. Instead of calibrating at all the 96 lens positions we just calibrated at these 6 almost uniformly spaced lens positions and for other lens positions we obtained the calibration data by merely shifting the nearest available calibration data.

The essence of CSTM is that we can take two images from any arbitrary lens positions S_f and $S_f + \Delta S$ to compute distance. To demonstrate this fact, we placed the objects at a known distance (say step 10) and the program was run by specifying a different starting lens position (10,20,30,40,50 or 60), everytime. It means that the first time the two images were taken at steps 10 and 10+30 = 40 and the program was run without changing the object position. The second time the object position was still the same as before but the program was run with starting lens steps of $S_1 = 20$ and $S_2 = 20 + 30 = 50$. This procedure was repeated for other lens positions of S_1 = 30,40,50 and 60.

Suppose we placed an object at step 10 and specified step 60 as the starting lens position. It is equivalent to the case when the object actually moved from step 60 to 10 (about 2 meters), a relatively high velocity, between two runs of the program. With one single object and for one single object position we did 6 experiments, corresponding to the 6 different starting lens positions mentioned above. The experiments were then repeated for 18 different object distances. Thus, with one single object we performed 6 * 18 = 108 experiments. For five different objects, the total number of experiments becomes 108 * 5 = 540.

During calibration, it was found that the estimated value of σ_2 (in Figure 4) was unreliable when both the images g_1 and g_2 (on which the estimation was based) were highly blurred. For this reason, calibration was limited to the case when the lesser blurred image, say g_1, was recorded at a position that was at most 25 lens steps away (corresponding to a radius of blur circle of about 7 pixels) from the focused lens position and the higher blurred image, say g_2, was recorded at a lens position that was at most 25+30 = 55 lens steps away (corresponding to a radius of blur circle of about 14 pixels) from the focused lens position. It is for this reason that the plots in Figure 4 do not cover the entire range (0 to 96) of lens positions. For example, "cal10.dat" in Figure 4 covers the range from step 10 to step 65. The range 66 to 96 is not covered because in that case the two images would be highly blurred. The range 0 to 10 steps is not covered by any plot because lens positions 0 and 5 correspond to placing objects at distances of 5.30 meters and 9.03 meters from the camera. Due to space restrictions in our laboratory, we were not able to place objects farther than 5 meters, and therefore the calibration data for these two points were not obtained.

If both images g_1 and g_2 are highly blurred, then reliable focusing can be achieved by iterating CSTM twice. The first iteration gives a rough estimation of the focused lens position. The lens is moved to this position and CSTM is applied again. In this case, the images will not be highly blurred as in the first iteration. Therefore good focusing will be achieved. In the experiments, the object was not moved between two iterations. However, modest movement (less than 20 lens steps) will not significantly alter the performance of CSTM.

Some of the results of CSTM1 are plotted in Figure 8. The X-axis indicates the experiment number. Since there are five objects and six lens positions, the number of experiments for each distance is 30. The Y-axis indicates the estimated distance in step number. The plot "step 17" shows the results when the objects are placed at step 17, and images are obtained with different lens positions. The other two plots in Figure 8 show the results when the objects are placed at step 56 and step 85 respectively. Ideally these plots should have been straight lines parallel to the X-axis.

The first set of experiments included 440 trials of the case when the two images were not highly blurred (corresponding to less than 25 steps of blur (7 pixel radius) for one image and less than 55 steps blur (14 pixel radius) for the second image). For these trials, the RMS error in focusing for one iteration of CSTM1 was 2.22 steps out of 97 steps, or about 2.3%. In terms of the radius of blur circle the error is about 0.417 pixel. The second set of experiments included the 440 trials of the first set and an additional 100 trials where both images were highly blurred (according to the criteria explained earlier). In order to perform trials for the highly blurred cases, when necessary, the calibration data in Figure 4 was extended through simple linear extrapolation of the plots. Two iterations of CSTM1 were performed for each of the 540 trials in the second set of experiments. The RMS error in focusing for these trials was 2.3 steps out of 97 steps or 2.4%. In terms of the radius of blur circle this error corresponds to about 0.432 pixel.

For experiments on CSTM2, only one set of calibration data corresponding to the plot “cal40.dat” in Figure 4 was used. This data set was shifted by appropriate amounts to obtain other required calibration data such as the plots labelled “cal10.dat”,“cal20.dat” etc., in the figure. Experiments similar to those described earlier for CSTM1 were repeated for CSTM2. The first set of experiments included 440 trials with one iteration, for the case when the two images were not highly blurred. The RMS error in focusing for these experiments was 2.9 steps out of 97 steps or about 3.0%. The second set of experiments included 540 trials which included the 100 trials where the two images were highly blurred. CSTM2 was run for two iterations as before and the RMS error in focusing was 3.05 steps out of 97 steps or about 3.1%.

A focusing error of 3% (corresponding to a radius of blur circle of about 0.56 pixel) is not perceptible by humans. Therefore, the results of CSTM are quite satisfactory. However, further improvement can be obtained by using a Depth-from-Focus (DFF) method and searching only in a small interval near the estimated lens position.

5 Conclusions

The DFD method based on STM has been extended to continuously focus on moving objects. It has been successfully demonstrated on an actual camera system built by us. Two variations of continuous focusing - CSTM1 and CSTM2- are presented. CSTM1 involves straight forward extension of the STM described in[12] and involves extensive camera calibration. The focusing accuracy was 2.3% by calibrating the camera system at 6 different pairs of lens positions. In CSTM2, the camera is calibrated just once corresponding to one lens position. The calibration data corresponding to other positions are obtained by transforming the data obtained for the one single position. A theoretical justification for this has been provided. The focusing accuracy of CSTM2 was found to be about 3% in lens position. The marginal improvement in accuracy of CSTM1 was achieved at the cost of a more cumbersome calibration procedure.

A typical application for CSTM is in autofocusing of video cameras, where it is necessary to quickly focus on objects which keep changing their positions. CSTM can also be used to continuously obtain a rough depth map of a dynamic scene. The resolution of the depth map can then be improved by using stereo vision techniques, if desired.

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

6 REFERENCES

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

[2] P. Grossman, “Depth from Focus”, *Pattern Recognition Letters* 5, pp. 63–69, Jan. 1987.

[3] J. Ens and P. Lawrence, “A Matrix Based Method for Determining Depth from Focus”, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1991.

[4] E. Krotkov, "Focusing", *International Journal of Computer Vision*, 1, 223-237, 1987.

[5] P. Meer and I. Weiss, "Smoothed Differentiation Filters for Images", *Journal of Visual Communication and Image Representation*, 3, 1, 1992.

[6] A. P. Pentland, "A New Sense for Depth of Field", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 4, pp. 523–531, 1987.

[7] S. K. Nayar, "Shape from Focus System" *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Champaign, Illinois, pp 302-308 June 1992.

[8] M. Subbarao, and G. Natarajan, "Depth Recovery from Blurred Edges", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Ann Arbor, Michigan, pp. 498-503, June 1988.

[9] M. Subbarao, and T. Wei, "Depth from Defocus and Rapid Autofocusing : A practical Approach", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Champaign, Illinois, June 1992

[10] M. Subbarao, "Spatial-Domain Convolution/Deconvolution Transform ", Tech. Report No. 91.07.03, Computer Vision Laboratory, Dept. of Electrical Engineering, State University of New York, Stony Brook, NY 11794-2350.

[11] M. Subbarao and G. Surya, "Application of Spatial-Domain Convolution/Deconvolution Transform for Determining Distance from Image Defocus", Vol. 1822, Proceedings of SPIE conference, OE/TECHNOLOGY '92, Boston, Nov. 1992, pp. 159 - 167.

[12] M. Subbarao and G. Surya, "Depth from Defocus: A Spatial Domain Approach", Tech. Report No. 92.12.03, Computer Vision Laboratory, Dept. of Electrical Engineering, State University of New York, Stony Brook, NY 11794-2350. (Revised version to appear in International Journal of Computer Vision).

[13] G. Surya and M. Subbarao, "Depth from Defocus by Changing Camera Aperture: A Spatial Domain Approach", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York, pp. 61-67, June 1993.

[14] T. Wei and M. Subbarao, "Continuous Focusing of Moving Objects Using DFD1F", Tech. Report No. 93.07.08, Computer Vision Laboratory, Dept. of Electrical Engineering, State University of New York, Stony Brook, NY 11794-2350.

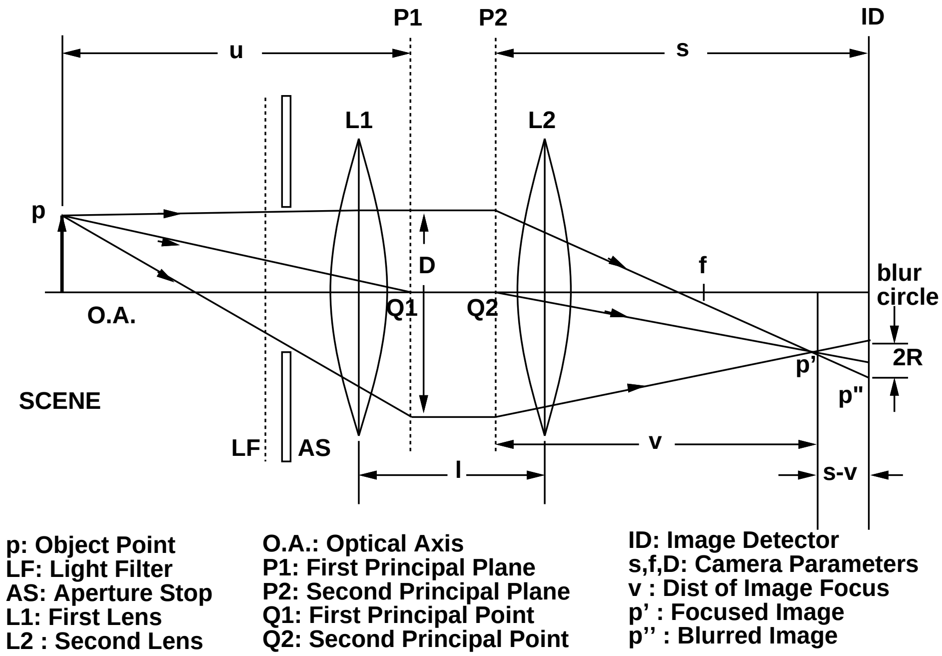
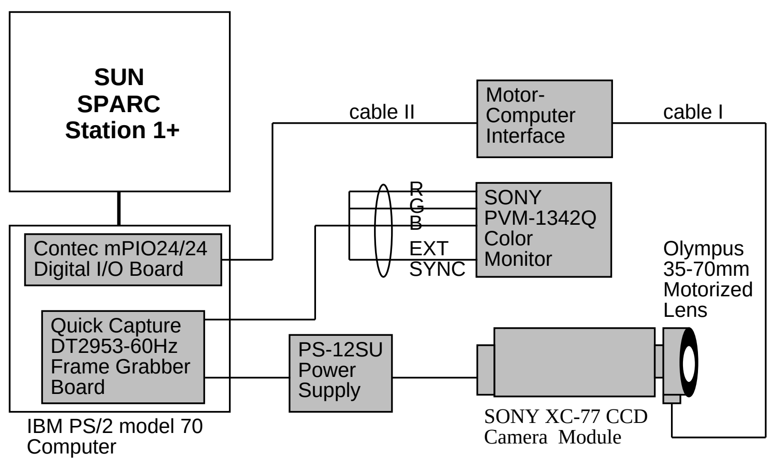


Figure. 1 Camera Model and Camera Parameters



Stonybrook Passive Autofocusing and Ranging Camera System-SPARCS - is a prototype camera system developed at the Computer Vision Laboratory for experimental research in robotic vision, State University of New York at Stony Brook.

Figure. 2 Block Diagram of SPARCS

Figure 3. Distance Vs Step Number

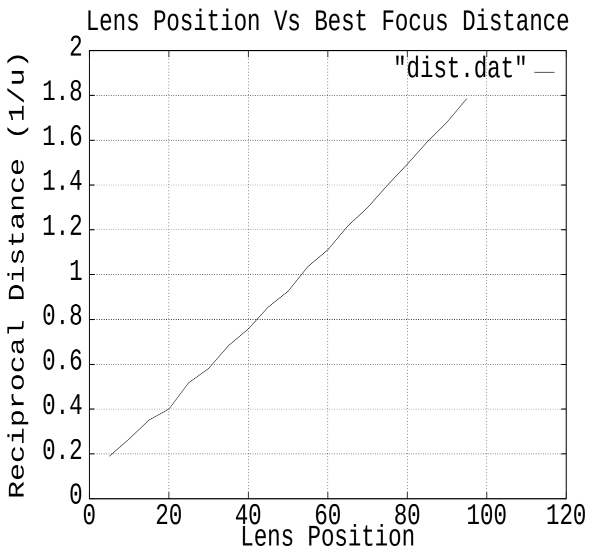
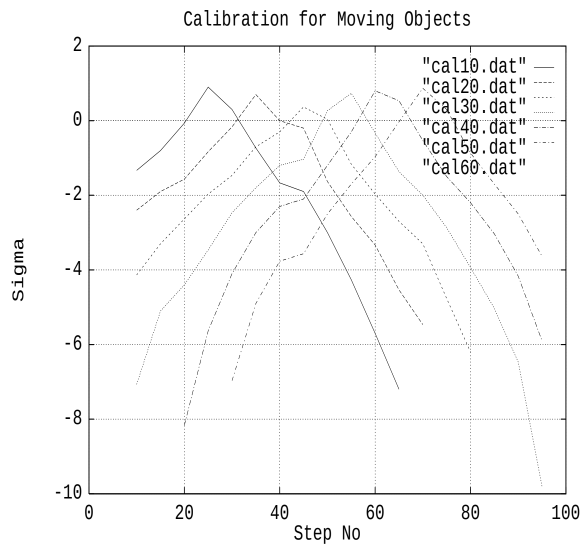


Figure 4. Sigma Vs Distance



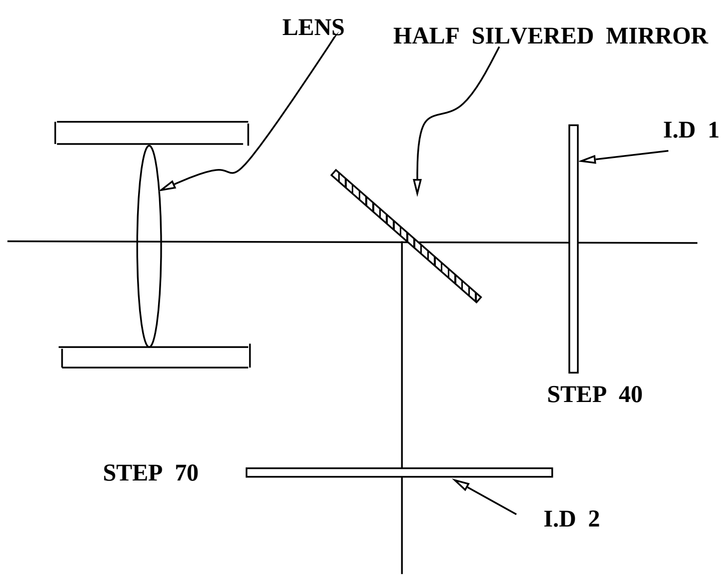


Figure 5. Hardware Implementation

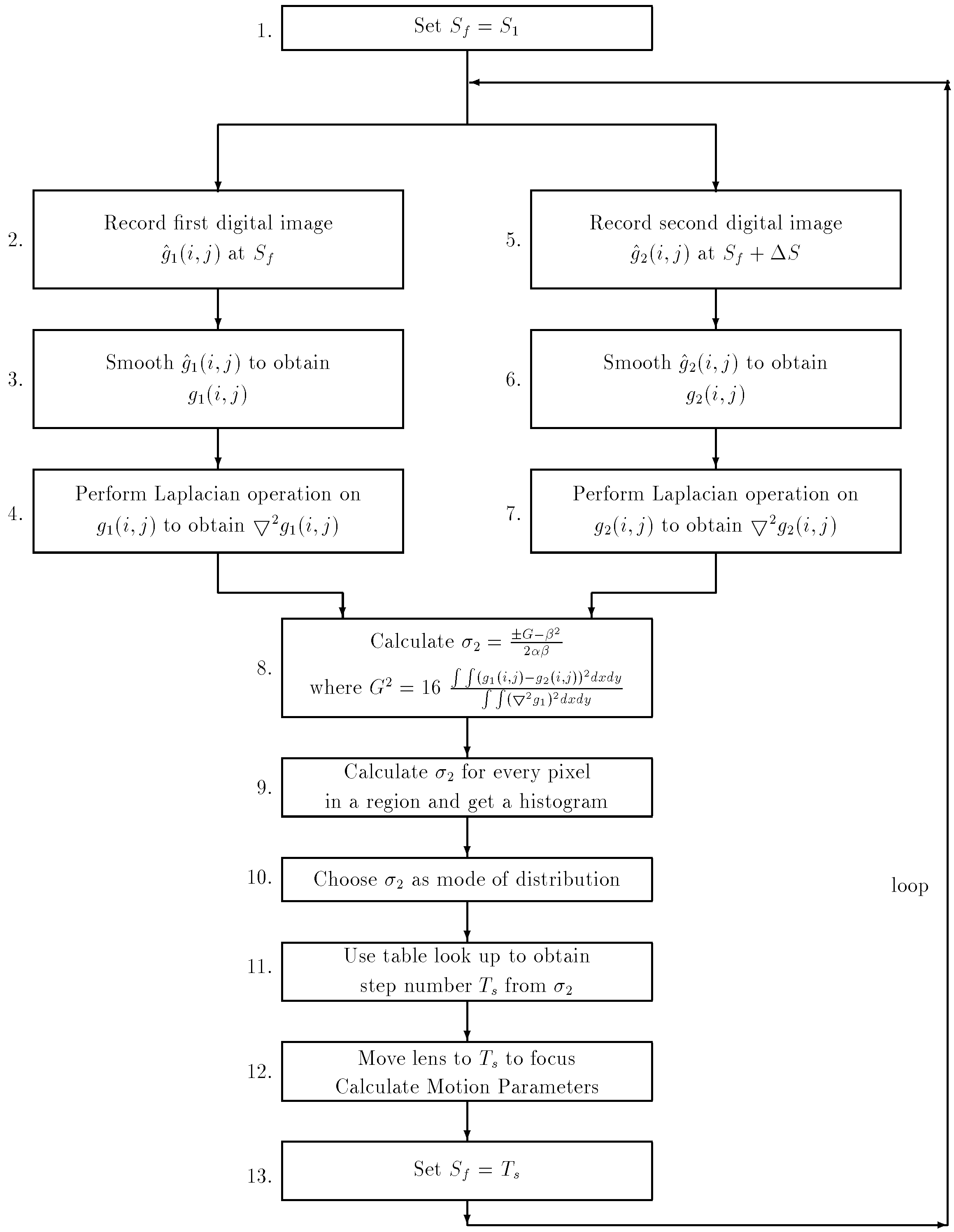


Figure 6. Flow Chart of CSTM

Figure 7. Objects Used: Tiger, Face, G.S., Micky, Chars

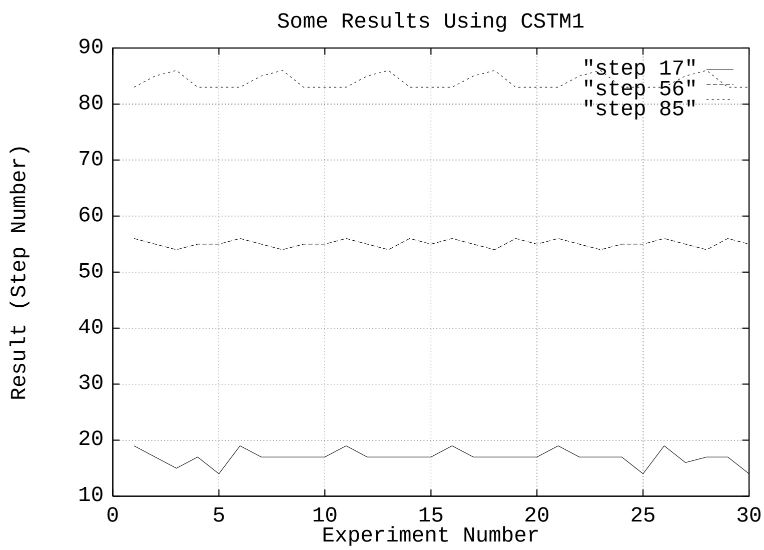


Figure 8. Some results on Moving Objects