Recovering Affine Motion and Defocus Blur Simultaneously

Zarina Myles, *Student Member, IEEE*,
and Niels da Vitoria Lobo, *Member, IEEE*

Abstract—Motion in depth and/or zooming cause defocus blur. We show how the defocus blur in an image can be recovered simultaneously with affine motion. We introduce the theory, develop a solution method and demonstrate the validity of the theory and the solution by conducting experiments with real scenery.

Index Terms—Defocus blur, affine motion, optical flow, defocussed motion.

1 Introduction

THERE are at least two situations in practical computer vision where displacement of a point in an image is accompanied by a defocus blur. Firstly, when a camera of fixed focal length moves along the line of sight some scene points move into focus and others get blurred due to defocus. We term this situation *defocussed motion*. The second situation arises when the camera zooms into a scene without an auto-focus mechanism (or with a slow one). Zooming has recently begun to be explored as a means to obtain depth [7], [5], [9], to compute the real center of the image [5] and to generally capture greater detail of a scene without moving. We term this situation *defocussed zoom*.

Measuring the level of blur/deblur due to defocus/focus caused by either camera zoom or camera motion along the optical axis will achieve two effects. Firstly, it will allow the correct optical flow to be computed despite the blur/deblur, and secondly it can be used to obtain initial estimates of depth for structure-from-motion and structure-from-zoom algorithms. These initial estimates of depth would be computed using shape from defocus/focus methods.

Work on shape from focusing and shape from defocus [11], [14], [2], [16], [10] involving at least two images taken with different camera parameters, has been conducted. However, these algorithms measure the level of blur (and compute depth) but do not compute blur in the presence of motion of the features in the image.

Older techniques for optical flow computation have been analyzed and compared [1]. Recent techniques include those which handle large affine transformations [15], [8] and those which extend the affine model to include second order transformations of moving planar patches [4]. Despite all these works, illumination changes, photometric motion effects, and image features moving in and out of focus when scene elements move in depth, are phenomena that have not been addressed satisfactorily. Here, we address the changes in defocus due to motion.

This paper computes both the blur level and affine parameters simultaneously from a pair of input images. We introduce the theoretical model of blur and affine motion; then we propose an iterative solution method. We report experimental results with real scenery. We are not concerned with motion blur since the underlying model is that of a stop-and-shoot sequence.

2 Theoretical Formulation

In this section, we develop the model and derive an equation relating the unknown parameters, namely the affine transformation and the level of blur.

It has been shown [6] that optical flow can be approximated by an affine transformation in the case of negligible out-of-plane rotations and in the case of planar patches. The theoretical formulation has been developed to handle scenes consisting of large, planar patches within which the affine transform model for optical flow is appropriate. Assume we have two images I_1 and I_2 where the second image is obtained after a large camera motion consisting of zooming in or out and rotation in the image plane. Even with improved technology there is a mechanical limit to the speed with which a camera can refocus itself. Hence, the large changes in depth cause the camera to get out of focus and it is appropriate to include defocus blur in the model of transformation that describes the geometric changes undergone between the two images. In the formulation below, we develop a linear relationship amongst the unknowns, namely the parameters of the affine transformation as well as the change in the level of blur between the two images. The linear approximation is valid if the unknowns take small values, and hence an iterative technique is developed to correctly recover *large* unknown parameters.

For the sake of clarity in the development of the formulation, we will first assume only a four-parameter affine transformation $\mathbf{A}$, where

$$\mathbf{A} = \mathbf{I} + \mathbf{B} = \begin{bmatrix} 1+b_{11} & b_{12} \\ b_{21} & 1+b_{22} \end{bmatrix}.$$

Here, the matrix $\mathbf{B}$ is composed of small elements which indicate the difference of $\mathbf{A}$ from the identity matrix. The formulation is then extended to the case of the general six-parameter affine transform which includes image translation as well. In this case the optical flow $[u, v]^T$ in that case can be expressed as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \vec{r} + \begin{bmatrix} T_x \\ T_y \end{bmatrix},$$

where $\vec{r} = [x, y]^T$ represents the underlying coordinate axes and $\vec{T} = [T_x, T_y]^T$ represents the pure translation component of optical flow.

2.1 Four-Parameter Affine Transformation

Let image I_a be an affinely transformed version of image I_1 where the transformation can be represented by the four-parameter matrix $\mathbf{A}$. Since deforming an image is equivalent to deforming the underlying coordinate axes, we can write $I_1(\vec{r}) = I_a(\mathbf{A}\vec{r})$. Manmatha [8] has shown that convolution of the first image I_1 with a Gaussian (G) is equivalent to the convolution of the second image I_a with a Gaussian which has been deformed by the same transformation $\mathbf{A}$. That is,

$$I_1(\vec{r}) \otimes G(\vec{r}, \sigma^2) = I_a(\mathbf{A}\vec{r}) \otimes G(\mathbf{A}\vec{r}, \sigma^2 \mathbf{A}\mathbf{A}^{\mathbf{T}})$$

where σ is any arbitrarily chosen value for standard deviation, $\otimes$ denotes convolution, the ordinary Gaussian $G(\vec{r}, \sigma^2)$ is

$$G(\vec{r}, \sigma^2) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\vec{r}^2}{2\sigma^2}\right)$$

and the generalized Gaussian $G(\mathbf{A}\vec{r}, \sigma^2 \mathbf{A}\mathbf{A}^{\mathbf{T}})$ (i.e., the Gaussian deformed by an affine transformation $\mathbf{A}$) is

- *The authors are with the School of Computer Science, University of Central Florida, Orlando, FL 32816. E-mail: {myles, niels}@cs.ucf.edu.*

Manuscript received 25 Mar. 1996; revised 2 Apr. 1998. Recommended for acceptance by S. Nayar.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 106712.

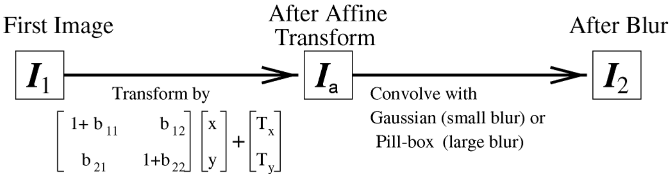

First Image
After Affine Transform
After Blur
I_1
I_a
I_2
Transform by
$\begin{bmatrix} 1+b_{11} & b_{12} \\ b_{21} & 1+b_{22} \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}$
Convolve with
Gaussian (small blur) or
Pill-box (large blur)

Fig. 1. Conceptual model incorporating affine transformation and defocus blur.

$$G\left(\mathbf{A}\vec{r}, \sigma^2\mathbf{A}\mathbf{A}^{\mathbf{T}}\right) = \frac{1}{2\pi \det(\mathbf{A})\sigma^2}\exp\left(-\frac{\vec{r}^T\left(\mathbf{A}\mathbf{A}^{\mathbf{T}}\right)^{-1}\vec{r}}{2\sigma^2}\right). \qquad (1)$$

To solve for the affine parameters in **A** an overdetermined system can be obtained by convolving at several points $\vec{l}_i$ in both images [8] to get

$$\int I_1(\vec{r})G\left(\vec{r} - \vec{l}_i, \sigma^2\mathbf{I}\right)d\vec{r} \approx \int I_a(\mathbf{A}\vec{r})G\left(\mathbf{A}\vec{r} - \vec{l}_i, \sigma^2\mathbf{A}\mathbf{A}^{\mathbf{T}}\right)d(\mathbf{A}\vec{r})$$

$$+\left[(\mathbf{A}-\mathbf{I})\vec{l}_i\right]^T \int I_a(\mathbf{A}\vec{r})G'\left(\mathbf{A}\vec{r} - \vec{l}_i, \sigma^2\mathbf{A}\mathbf{A}^{\mathbf{T}}\right)d(\mathbf{A}\vec{r}) \quad (2)$$

where l_i is any image point, G' is the derivative of G w.r.t. the image coordinates. Equation (2) is Taylor's first order expansion of $G\left(\mathbf{A}\left(\vec{r} - \vec{l}_i\right), \sigma^2\mathbf{A}\mathbf{A}^{\mathbf{T}}\right)$ about the point $\left(\mathbf{A}\vec{r} - \vec{l}_i\right)$.

Equation (2) cannot be solved for the unknown **A** since G on the R.H.S. can't be evaluated. Hence we rewrite it in a linear form of the unknown **B**. To achieve this we express the generalized Gaussian in terms of the ordinary Gaussian and its derivatives using Taylor's expansion of (2) about the matrix **B = 0** (meaning no affine transformation) where $\mathbf{B} = \mathbf{A} - \mathbf{I} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$. Equation (1) can be expanded up to first order terms in **B** about point **B** = **0** yielding

$$G(., \sigma^2\mathbf{A}\mathbf{A}^{\mathbf{T}}) \approx G(., \sigma^2) + \sigma^2\, b_{11}\, G_{xx}(., \sigma^2) + \sigma^2\, b_{12}\, G_{xy}(., \sigma^2)$$

$$+\ \sigma^2\, b_{21}\, G_{yx}(., \sigma^2) + \sigma^2\, b_{22}\, G_{yy}(., \sigma^2)$$

When the affine transformation is small, this is a reasonable approximation. Ignoring second and higher order terms in the components of **B** and setting $\vec{r}_1 = \mathbf{A}\vec{r}$, (2) is rewritten as

$$I_1(\vec{r}) \otimes G\left(\vec{r} - \vec{l}_i, \sigma^2\right) \approx I_a\left(\vec{r}_1\right) \otimes G\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right) + \left(\mathbf{B}\,\vec{l}_i\right)^T I_a\left(\vec{r}_1\right) \otimes G'\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right)$$

$$+\sigma^2 b_{11} I_a\left(\vec{r}_1\right) \otimes G_{xx}\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right) + \sigma^2 b_{12} I_a\left(\vec{r}_1\right) \otimes G_{xy}\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right)$$

$$+\sigma^2 b_{21} I_a\left(\vec{r}_1\right) \otimes G_{yx}\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right) + \sigma^2 b_{22} I_a\left(\vec{r}_1\right) \otimes G_{yy}\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right). (3)$$

The derivation obtained earlier in [8], and our (3) are now extended to include defocus blur. Denote the left hand side of (3) by $\mathcal{I}_1(\vec{r})$ and the right hand side by $\mathcal{I}_a(\mathbf{A}\vec{r})$. Then

$$\mathcal{I}_1(\vec{r}) \approx \mathcal{I}_a(\mathbf{A}\vec{r}). \qquad (4)$$

Now let us assume that image I_a is further transformed by a blur to yield another image I_2 (see Fig. 1). The actual point spread function is given by the convolution of the diffraction limited point spread function (Airy disc) with the geometrically aberrated (defocused) blur function (circle) ([12, pp. 147-150]). At lower levels of blur, diffraction effects dominate and the blur can be approximated by the Gaussian, which is an approximation of the Airy disc. At higher levels of blur, the defocus effects dominate and the blur can be approximated by a convolution with a pill-box function as used in [3], [13], [10] (see Fig. 2). Hence, our model assumes the Gaussian model for small levels of blur and the pill-box model for larger levels.

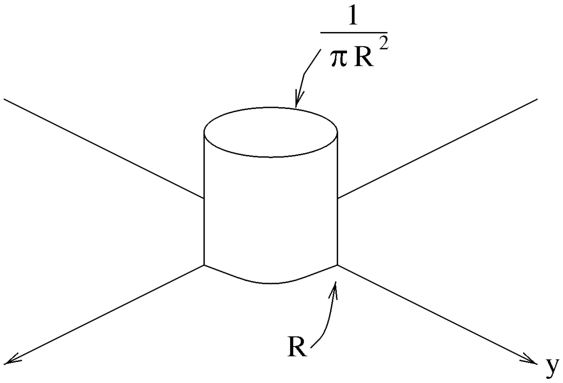


Fig. 2. Pillbox function.

$$Pillbox(\vec{r}, R) = \begin{cases} \frac{1}{\pi R^2} & for\,|\vec{r}| \le R \\ 0 & otherwise \end{cases},$$

where increasing values of the radius R imply increasing levels of blur. Define the generalized Pillbox function, a pillbox deformed by affine **A**, as

$$P(\vec{r}, R, \mathbf{A}) = \begin{cases} \frac{1}{\pi R^2|\mathbf{A}|} & for\left|\mathbf{A}^{-1}\vec{r}\right| \le R \\ 0 & otherwise \end{cases},$$

where $|\mathbf{A}|$ denotes the determinant of **A**. Using mathematical substitution we see that

$$P\left(\vec{r} - \vec{l}_i, R, \mathbf{A}^{-1}\right) = |\mathbf{A}|P\left(\mathbf{A}\left(\vec{r} - \vec{l}_i\right), R, \mathbf{I}\right). \qquad (5)$$

In order to convolve the right hand side of (4) with the appropriate Pill-box function, we first multiply (4) by (5) to yield

$$\mathcal{I}_1(\vec{r})P\left(\vec{r} - \vec{l}_i, R, \mathbf{A}^{-1}\right) \approx \mathcal{I}_a(\mathbf{A}\vec{r})|\mathbf{A}|P\left(\mathbf{A}\left(\vec{r} - \vec{l}_i\right), R, \mathbf{I}\right).$$

Integrating both sides with respect to $\vec{r}$, we get

$$\int \mathcal{I}_1(\vec{r})P\left(\vec{r} - \vec{l}_i, R, \mathbf{A}^{-1}\right)d(\vec{r}) = \int \mathcal{I}_a(\mathbf{A}\vec{r})|\mathbf{A}|P\left(\mathbf{A}\left(\vec{r} - \vec{l}_i\right), R, \mathbf{I}\right)d(\vec{r})$$

$$= \int \mathcal{I}_a(\mathbf{A}\vec{r})P\left(\mathbf{A}\left(\vec{r} - \vec{l}_i\right), R, \mathbf{I}\right)d(\mathbf{A}\vec{r}). \quad (6)$$

The last step follows from the fact that

$$\int d\vec{r} = \int\int dxdy = \frac{1}{|\mathbf{A}|}\int\int d(x_1)d(y_1),$$

where

$$(x_1, y_1)^T = \vec{r}_1 = \mathbf{A}\vec{r}.$$

The left-hand side of (6) represents a convolution at point $\vec{r}$. If this is repeated at every point in the image, we have

$$LHS = \mathcal{I}_1(\vec{r}) \otimes G\left(\vec{r} - \vec{l}_i, \sigma^2\right) \otimes P\left(\vec{r} - \vec{l}_i, R, \mathbf{A}^{-1}\right).$$

We will assume that the unknowns **B** and R are very small. The function $P\left(\vec{r}, R, \mathbf{A}^{-1}\right)$ can then be approximated by the Gaussian $G\left(\vec{r}, \frac{R^2}{2}\right)$, to give

$$LHS = I_1(\vec{r}) \otimes G\left(\vec{r} - \vec{l}_i, \sigma^2\right) \otimes G\left(\vec{r}, R^2/2\right)$$
$$= I_1(\vec{r}) \otimes G\left(\vec{r} - \vec{l}_i, \sigma^2 + R^2/2\right). \quad (7)$$

The last step uses the result that the convolution of two Gaussians yields another Gaussian.

The right side of (6) can be expressed as

$$RHS = P\left(\mathbf{A}\left(\vec{r} - \vec{l}_i\right), R, \mathbf{I}\right) \otimes$$
$$\Big[I_a\left(\vec{r}_1\right) \otimes G\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right) + \left(\mathbf{B}\vec{l}_i\right)^T I_a\left(\vec{r}_1\right) \otimes G'\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right)$$
$$+ \sigma^2 b_{11} I_a\left(\vec{r}_1\right) \otimes G_{xx}\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right) + \sigma^2 b_{12} I_a\left(\vec{r}_1\right) \otimes G_{xy}\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right)$$
$$+ \sigma^2 b_{21} I_a\left(\vec{r}_1\right) \otimes G_{yx}\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right) + \sigma^2 b_{22} I_a\left(\vec{r}_1\right) \otimes G_{yy}\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right)\Big]$$
$$= I_2\left(\vec{r}_1\right) \otimes G\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right) + \left(\mathbf{B}\vec{l}_i\right)^T I_2\left(\vec{r}_1\right) \otimes G'\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right)$$
$$+ \sigma^2 b_{11} I_2\left(\vec{r}_1\right) \otimes G_{xx}\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right) + \sigma^2 b_{12} I_2\left(\vec{r}_1\right) \otimes G_{xy}\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right)$$
$$+ \sigma^2 b_{21} I_2\left(\vec{r}_1\right) \otimes G_{yx}\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right) + \sigma^2 b_{22} I_2\left(\vec{r}_1\right) \otimes G_{yy}\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right). \quad (8)$$

The last step follows because convolution is commutative, and because I_2 is obtained from I_a as a result of a convolution with a Pillbox function. Finally, equating (7) and (8) we get a relationship which is linear in the unknown **B**, albeit still nonlinear in *R*.

2.2 General Affine Transform Including Image Translation

Now add arbitrary image translation $\delta\vec{T}$ to the above affine transformation and blurring. We see that convolution about a point $\vec{r} + \delta\vec{T}$ by a Gaussian $G(\vec{r},.)$ is equivalent to convolving about the point $\vec{r}$ by the Gaussian $G\left(\vec{r} - \delta\vec{T},.\right)$. Hence, once a rough estimate of the image translation $\vec{T}_0$ is known, the residual translation $\delta\vec{T}$ is computed by using the result

$$I_1\left(\vec{r} + \vec{T}_0 + \delta\vec{T}\right) \otimes G\left(\vec{r} - \vec{l}_i, \sigma^2 + R^2/2\right) =$$
$$I_1\left(\vec{r} + \vec{T}_0\right) \otimes G\left(\vec{r} - \vec{l}_i - \delta\vec{T}, \sigma^2 + R^2/2\right).$$

Substituting the above in (7) and (8) we get

$$I_1\left(\vec{r} + \vec{T}_0\right) \otimes G\left(\vec{r} - \vec{l}_i - \delta\vec{T}, \sigma^2 + R^2/2\right) \approx I_2\left(\vec{r}_1\right) \otimes G\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right)$$
$$+ \left(\mathbf{B}\vec{l}_i\right)^T I_2\left(\vec{r}_1\right) \otimes G'\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right)$$
$$+ \sigma^2 b_{11} I_2\left(\vec{r}_1\right) \otimes G_{xx}\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right) + \sigma^2 b_{12} I_2\left(\vec{r}_1\right) \otimes G_{xy}\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right)$$
$$+ \sigma^2 b_{21} I_2\left(\vec{r}_1\right) \otimes G_{yx}\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right) + \sigma^2 b_{22} I_2\left(\vec{r}_1\right) \otimes G_{yy}\left(\vec{r}_1 - \vec{l}_i, \sigma^2\right). \quad (9)$$

Thus (9) relates the affine unknowns, **B** and $\delta\vec{T}$ and the level of blur *R*. To solve this equation, we linearize in $\delta\vec{T}$ and subsequently in *R*.

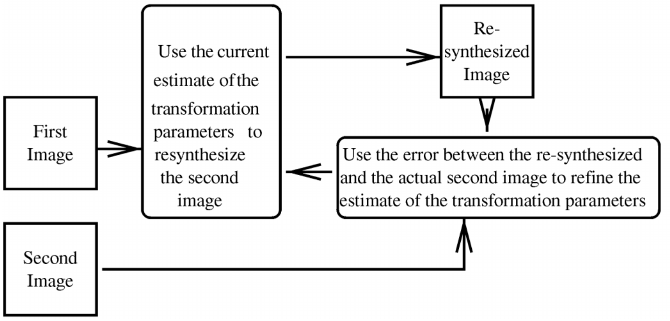

Use the current estimate of the transformation parameters to resynthesize the second image
Re-synthesized Image
First Image
Use the error between the re-synthesized and the actual second image to refine the estimate of the transformation parameters
Second Image

Fig. 3. Overview of the iterative parameter estimation process.

$$G_1\left(\vec{r} - \vec{l}_i - \delta\vec{T}, .\right) \approx G_1\left(\vec{r} - \vec{l}_i, .\right) - \delta\vec{T}^T G'\left(\vec{r} - \vec{l}_i, .\right)$$

(which is a Taylor's series approximation about the point $\left(\vec{r} - \vec{l}_i\right)$) we obtain (9) in a linear form of the unknowns **B** and $\delta\vec{T}$.

Finally, we linearize in the unknown radius of blur, *R*. We first set $\sigma^2 = \eta$ and define

$$G_1(\vec{r}, \eta) = \frac{1}{2\pi\eta} \exp\left(\frac{-\vec{r}^T\vec{r}}{2\eta}\right).$$

Let $\sigma^2 + \frac{R^2}{2} = \eta_2 = \eta_1 + \beta$ where η_1 is the current best estimate of η and β is the residual error in the estimation. Expanding $G_1(\vec{r}, \eta_2)$ by Taylor's series about the point η_1, we get

$$G_1(\vec{r}, \eta_2) = G_1(\vec{r}, \eta_1) + \beta \left.\frac{\partial G_1}{\partial \eta}\right|_{\eta=\eta_1}.$$

Expanding the left hand side of (9) about η_1,

$$I_1(\vec{r}) \otimes G\left(\vec{r}, R^2 + \sigma^2\right) \approx I_1(\vec{r}) \otimes G_1(\vec{r}, \eta_1) + \beta I_1(\vec{r}) \otimes \left.\frac{\partial G_1}{\partial \eta}\right|_{\eta=\eta_1} =$$
$$I_1(\vec{r}) \otimes G_1(\vec{r}, \eta_1) + 2\beta I_1(\vec{r}) \otimes \nabla^2 G_1\Big|_{\eta=\eta_1}.$$

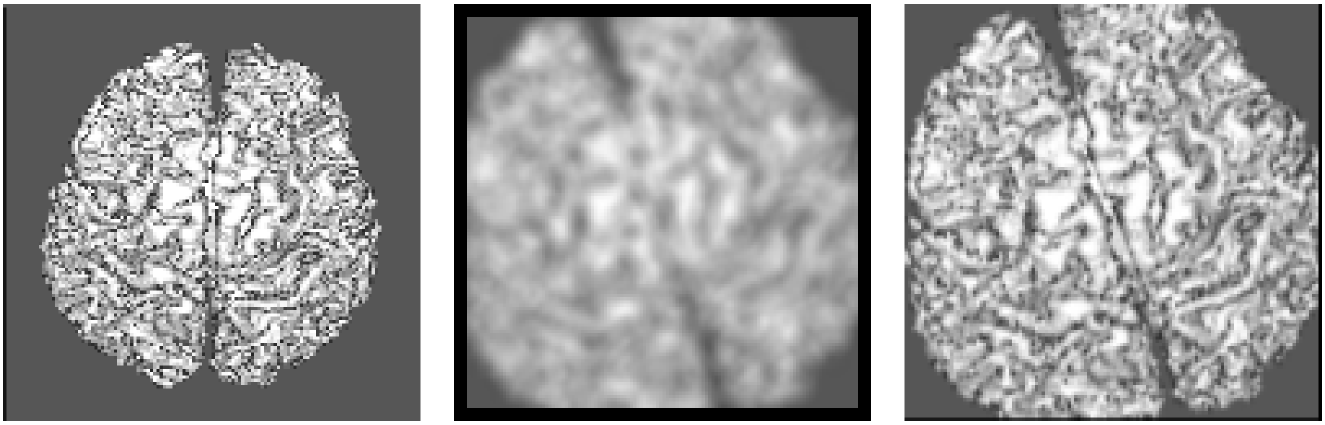
The last step follows from $\frac{\partial G_1}{\partial \eta} = 2\nabla^2 G_1$.

The overall computation is embedded in a two-level pyramid scheme. All parameters are estimated using the higher level first, and then are propagated as initial estimates for the detailed level. A translational component of $\left(T_x^s, T_y^s\right)^T$ in an image shrunk by a factor of *s* corresponds to $\left(sT_x^s, sT_y^s\right)^T$ in the original image. The pyramid is constructed by performing bilinear interpolation to shrink overlapping patches of the original images.

3 Solution Method

Using one or more values of σ (in our experiments, values of σ were chosen as 1.75, 2.5, 3.0, 3.5, 4.5) as well as different values of $\vec{l}_i$ an over-determined system is obtained to solve for the unknowns, the affine parameters **B**, the residual translation $\delta\vec{T}$ and the radius of blur *R*. In practice even if the initial estimate of the translation is three to four pixels in error, the above method is able to rightly identify the exact image translation to subpixel accuracy.

Since the linearization is an approximation which holds true for only small values of the unknowns, an iterative scheme was developed to handle large deformations (see Fig. 3). At the first iteration approximate values of the unknowns are obtained. An



(a) (b) (c)

Fig. 4. Artificially transformed real image. (a) Original image. (b) Artificially deformed. (c) Regenerated using recovered affine parameters.

intermediate synthetic image is then obtained by transforming (via bilinear interpolation) the first image using the computed affine parameters and then blurring the latter using the Pillbox model of blur.

At the next iteration the difference between this intermediate image and the second image is computed and the overall affine parameters and degree of blur are calculated. The first image is then transformed using these *new* parameters to form the *next* intermediate image. This process is repeated until the residual of the linear system is below a predetermined threshold.

As the linearization approximation is valid when the deformations are small and the method has to iteratively recover decreasing values of the unknowns, the above method converges.

The parameters $\mathbf{B}$ and η_1 are initialized to $\mathbf{0}$ and some σ^2 respectively, corresponding to no affine transformation or blurring. At every iteration, we update $\mathbf{A}_{new} \leftarrow \mathbf{A}_{residual} \cdot \mathbf{A}_{previous}$ (matrix multiplication) and R as

$$R \leftarrow \sqrt{2}\sqrt{\eta_1 + \beta - \sigma^2} \tag{10}$$

and compute a new value of η_1 as $\eta_1 = \sigma^2 + \frac{R^2}{2}$. A negative value for $\eta_1 + \beta - \sigma^2$ indicates that the second image has been sharpened instead of blurred. This allows automatic segmentation of an image into areas which are blurred, sharpened or focally unperturbed (i.e., $R = 0$).

4 Implementation and Results

We first describe results obtained when a real image was artificially deformed using large affine parameters and substantial levels of blur. The method was then implemented on two real image pairs, i.e., the second image in a pair was obtained by camera motion or zoom rather than artificially generated.

We ran several experiments on a wide range of test images. We artificially deformed them by performing an affine transform using bilinear interpolation (expansion factors ranging from 0.7 to 1.4, rotations up to 30 degrees and image translations within four pixels) followed by a blurring operation. The program correctly recovered all parameters. The first image was then transformed according to the parameters recovered with the effect of blurring removed.

At every point a texture measure was computed. In a small patch (typically 10 × 10) around the pixel the difference between each pixel and its neighbor (at its right and below it) was summed. When the average difference was less than 10 (gray-levels) the pixel was not considered to have enough texture and the recovery of affine parameters was not attempted there. The overdetermined system was solved using a least mean squares algorithm. We used gradient descent. The residual of the least squares system is interpreted as a measure of confidence in the observed parameters. It was experimentally determined that the number and organization of the points l_i (see (2)) at which convolutions are performed depend upon both the level of affine deformation as well as the level of blur. For instance, when recovering relatively large values, such as a radius of blur of 3.5 pixels and a scaling factor of 1.2 as well as in-the-plane translation of $[2, 2]^T$, one needed 121 evenly spaced points within a 41×41 grid. However for smaller deformations such as a radius of blur of one pixel, a scale factor of 1.04, an in-the-plane rotation of 2^o, and an in-the-plane translation of $[1.0, 0.5]^T$ pixels, 25 evenly-placed points l_i were chosen within a 17×17 grid. If the approximate level of deformation is known a priori, the number and configuration of points l_i can be judiciously selected. The optical flow is the translational component of the computed affine transformation.

4.1 Artificially Deformed Image

An image of a brain (Fig. 4a) was transformed by

$$\begin{bmatrix} 1.2216 & -0.4446 \\ 0.4446 & 1.2216 \end{bmatrix} + \begin{bmatrix} -1.0 \\ 0.7 \end{bmatrix}$$

(scaling of 1.3, rotation of 20^o, translation of $[-1, .7]^T$) and Pill-box blurred with radius $R = 3.5$ pixels.

The artificially transformed image is shown in (Fig. 4b). The affine transformation recovered were

$$\begin{bmatrix} 1.2215 & -0.4446 \\ 0.4446 & 1.2218 \end{bmatrix} + \begin{bmatrix} -1.0 \\ 0.7 \end{bmatrix}$$

and the radius of the blur was computed to be 3.5 pixels. Fig. 4c has been generated from the first image using the recovered affine parameters at the center of the image.

4.2 Pairs of Real Images

Experiments were then conducted on pairs of images of real scenery. Below we elaborate the results obtained in two of these experiments with real data.

4.2.1 *Experiment 1: Geometrical Pattern*

To facilitate computing the correct affine parameters using point correspondences, geometric patterns were photographed with a digital camera. Results are shown in Fig. 5.

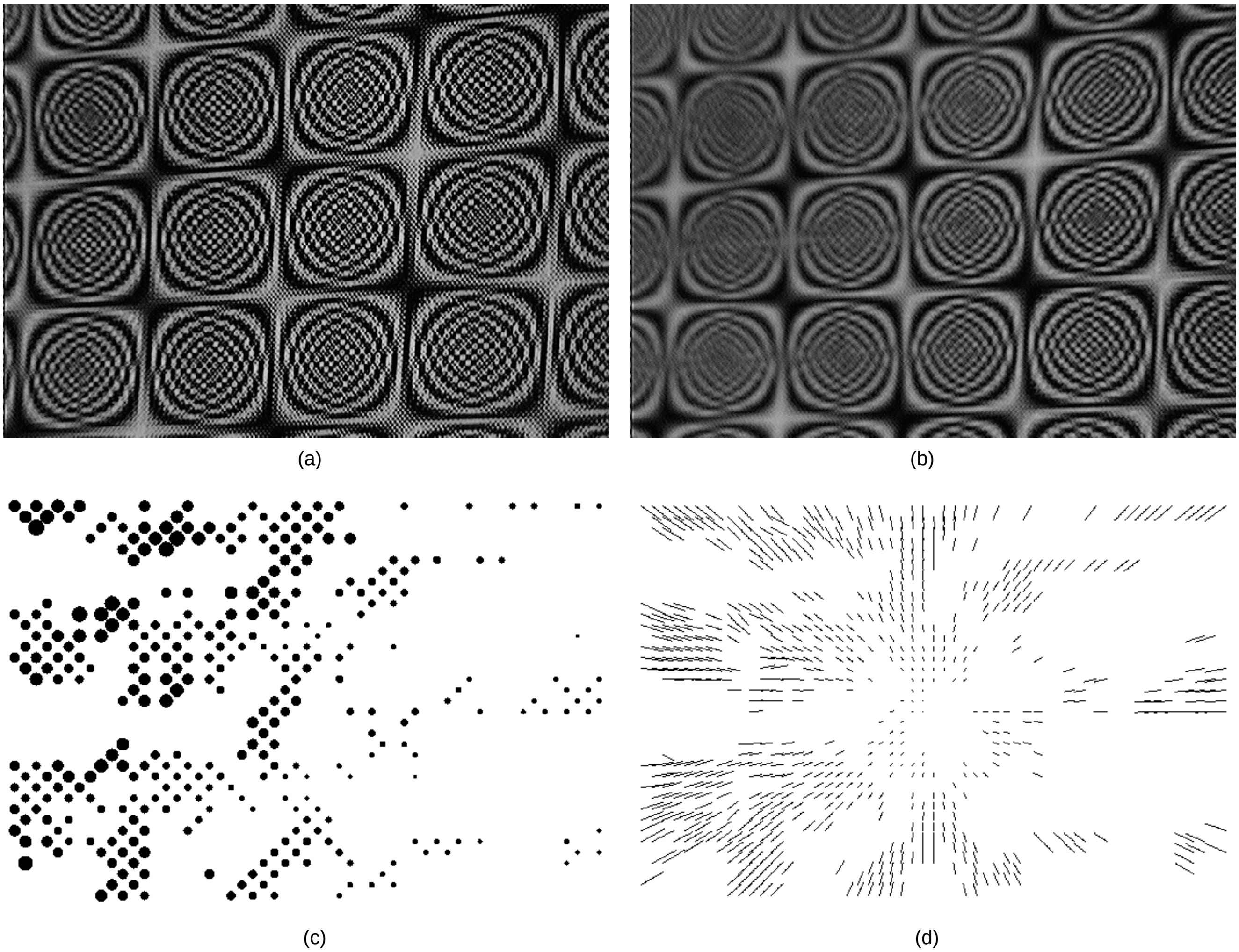


Fig. 5. Experiment 1: Slanted geometric pattern. (a) First image. (b) Second image. (c) Radius map in central patch. Circle size is proportional to radius of blur. (d) Flow field.

Fig. 5a is the first image of a slanted geometrical pattern and Fig. 5b is the second image, which is obtained by keeping the camera stationary and using the zoom mechanism.

The computation was run on this pair of images. The affine parameters obtained at the center were

$$\begin{bmatrix} 0.9400 & 0.0146 \\ 0.0149 & 0.9327 \end{bmatrix}$$

along with an image translation of

$$\begin{bmatrix} -0.12 \\ -0.05 \end{bmatrix}$$

and the radius of pillbox blur of 4.4 pixels.

The degree of blur and flow field in a central window of size 440 × 260 is displayed for every twentieth pixel. The blur level increases as one travels from the bottom right to the top left and this can be seen in Fig. 5c. Fig. 5d is the flow field in the same window.

4.2.2 *Experiment 2: Two Objects at Different Depths*

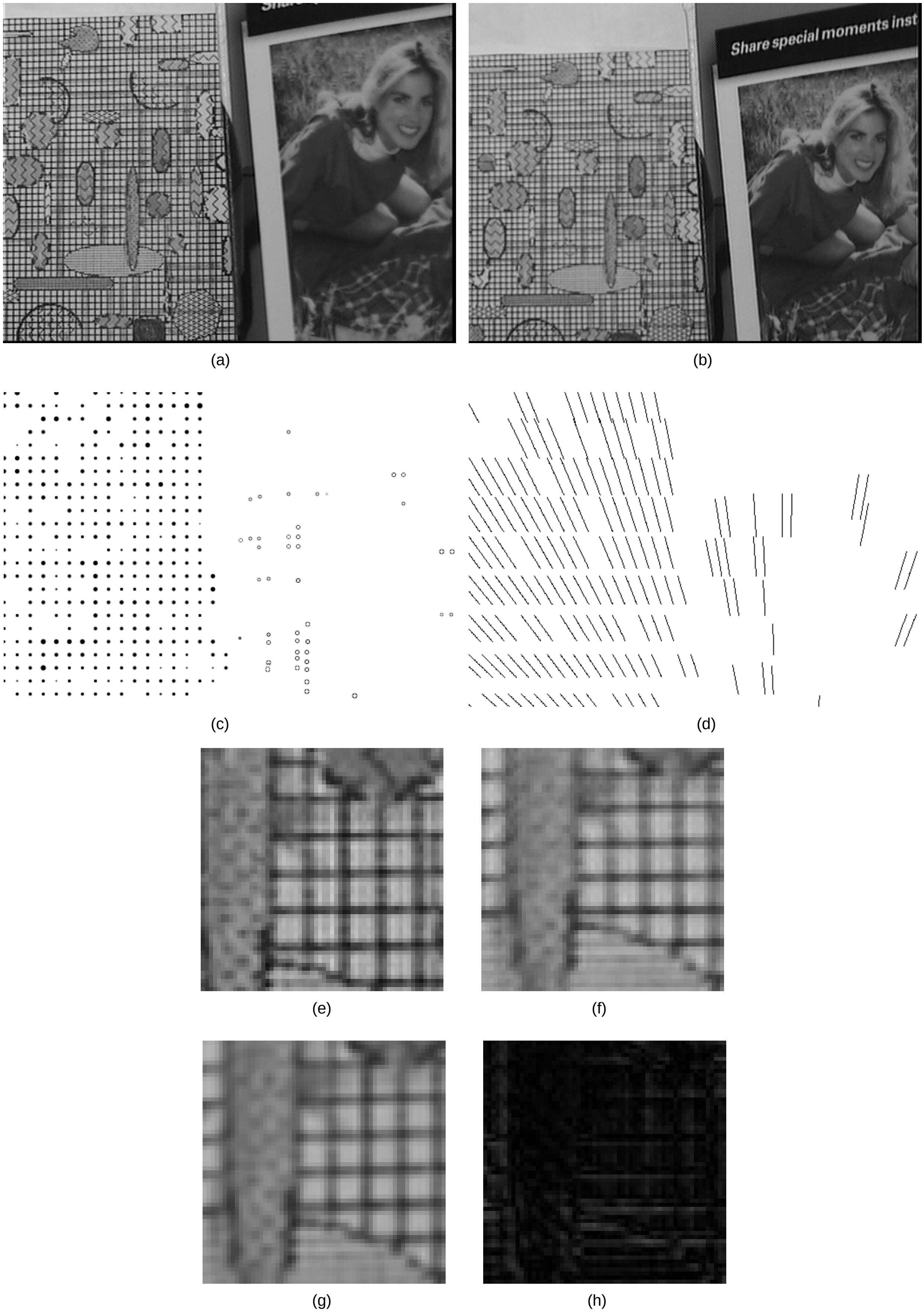
In the second experiment, we imaged two objects. One was a box with an irregular design, the other was a slanted box with a picture of a woman. The camera's translation component caused the left object to get blurred and the right to get sharpened. The program correctly recognized the two situations. We show the subsampled radius map and motion for a 520 × 360 patch in the image. The radii corresponding to points which got sharper/blurred are shown as open/filled-in circles. Since the flow is large, the flow image is highly subsampled. Areas corresponding to the woman's dress have insufficient texture to compute the motion and blur.

To assess the accuracy of the blur parameter, we considered a 60 × 60 patch (Fig. 6e) centered on the position (205, 320). This patch was affine transformed and blurred using the recovered parameters. The modified patch (Fig. 6f) was compared with the equivalent patch in the second image (Fig. 6g). In the difference image (Fig. 6h), the average absolute error of the difference was 12 gray values and the maximum error was 67. In this experiment, the camera's aperture was deliberately set so that motion induced a large level of blur in order to demonstrate that the algorithm is able to handle large deformations.

5 Discussion and Conclusion

We conducted experiments to study the stability of the blur estimates with respect to noise.

A series of experiments was performed where the second image was deformed with increasing levels of affine deformation. In each experiment both images were subjected to increasing levels of Gaussian noise. In all the experiments the radius of blur in the second image was set to three pixels. The percentage error of the computed radius is plotted in Fig. 7.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Fig. 6. Experiment 2: Two objects at different depths. (a) First image. (b) Second image. (c) Radius map in 520 × 360 patch. Open/filled circles show areas which have sharpened/blurred. (d) Flow field in same patch. (e) Patch from first image. (f) Regenerated with recovered values. (g) Corresponding patch in second image. (h) Difference image.

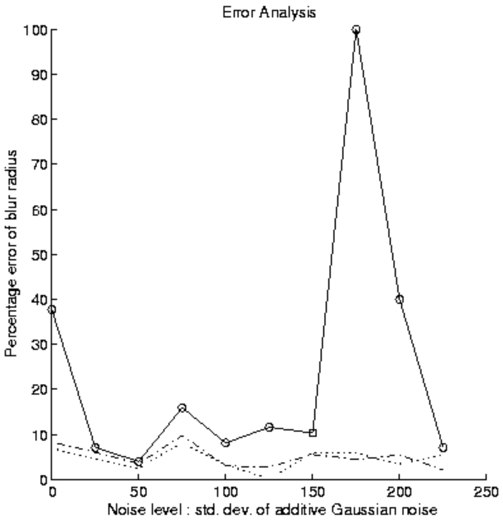


Fig. 7. Percentage of error in blur radius versus Gaussian noise level.

An artificially generated image of a sine pattern was employed. In Experiment 1 (represented in Fig. 7 by a dotted line) the second image was rotated by 15^o, scaled by a factor of 1.2 and translated by [1, 0] pixels. In Experiment 2 (represented by a dot-dash line) the second image was rotated by 20^o, scaled by a factor of 1.4 and translated by [2, 2] pixels. In the third experiment (represented by a line and circles), the image was rotated 25^o, scaled by a factor of 1.5 and translated [3, 3] pixels. In all the experiments zero-mean Gaussian noise was added to each pixel of both images. The standard deviation of the Gaussian was varied from zero to 225 in steps of 25. An error of 100 percent in the figure implies that covergence did not take place. It was seen that the performance deterioriated gradually as the level of additive noise was increased and as the degree of deformation of the second image was increased. Thus, we conclude that the computation is robust to noise.

We have introduced a novel method to measure affine motion and the defocus blur simultaneously. We have experimentally demonstrated the validity of our model using real image pairs. A future goal is to use the recovered blur to get initial estimates of depth in an iterative computation of shape from motion.

Recent work [10] handles displacement of image points due to blur using a telecentric lens. Our method obviates the need for an additional lens by including the deformation in a more comprehensive computational model. In addition, we have shown how the method can be used to segment an image into regions which have blurred, sharpened or remained focally unperturbed during the affine motion. Our method's primary drawback is that it requires the existence of fairly large planar patches in the images and future work will address this.

ACKNOWLEDGMENTS

The authors thank NSF (CCR-9410459) for support and Niels Häring for proofreading.

REFERENCES

[1] J.L. Barron, D.J. Fleet, and S.S. Beauchemin, "Performance of Optical Flow Techniques," *Int'l J. Computer Vision*, vol. 12, pp. 43-77, 1994.

[2] J. Ens and P. Lawrence, "A Matrix Method for Determining Depth From Focus," *Computer Vision and Pattern Recognition*, pp. 600-606, 1991.

[3] B.K.P. Horn, *Robot Vision*. New York, NY: McGraw-Hill, 1986.

[4] M. Irani, B. Rousso, and S. Peleg, "Recovery of Ego-Motion Using Image Stabilization," *Computer Vision and Pattern Recognition*, pp. 454-460, 1994.

[5] J.-M. Lavest, G. Rives, and M. Dhome, "Three Dimensional Reconstruction by Zooming," *IEEE Trans. Robotics and Automation*, vol. 9, pp. 196-207, 1993.

[6] H. Liu, T.H. Hong, M. Herman, and R. Chellappa, "A Generalized Motion Model for Estimating Optical Flow Using 3-D Hermite Polynomials," *Int'l Conf. Pattern Recognition-A*, pp. 361-366, 1994.

[7] J. Ma and S.I. Olsen, "Depth From Zooming," *J. Optical Soc. Am.*, vol. 7, no. 4, pp. 1,883-1,890, 1990.

[8] R. Manmatha, "A Framework for Recovering Affine Transforms Using Points, Lines, or Image Brightness," *Computer Vision and Pattern Recognition*, pp. 141-146, 1994.

[9] B.G. Mobasseri and S. Doraiswamy, ”Virtual Motion: 3-D Scene Recovery Using Focal Length-Induced Optic Flow," *IEEE Int'l Conf. Image Processing*, vol. 3, pp. 78-82, 1994.

[10] S. Nayar, M. Watanabe, and M. Noguchi, "Real-Time Focus Range Sensor," *Int'l Conf. Computer Vision*, pp. 995-1,001, 1995.

[11] A.P. Pentland, "A New Sense for Depth of Field," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, pp. 522-531, 1987.

[12] W.J. Smith, *Modern Optical Engineering,* 2nd Edition. McGraw-Hill, 1990.

[13] M. Subbarao, "Focussed Image Recovery From Two Defocussed Images Recorded With Different Camera Settings," Technical Report 93.10.21, State Univ. of New York, 1993.

[14] G. Surya and M. Subbarao, "Depth From Defocus by Changing Camera Aperture: A Spatial Domain Approach," *Computer Vision and Pattern Recognition*, pp. 61-67, 1993.

[15] P. Werkhoven and J.J. Koenderink, "Extraction of Motion Parallax Structure in the Visual System I," *Biological Cybernetics*, vol. 63, pp. 185-191, 1990.

[16] Y. Xiong and S. Shafer, "Depth From Focusing and Defocusing," *Image Understanding Workshop*, pp. 967-976,1993.