Recovering Affine Motion and Defocus Blur Simultaneously

Zarina Myles and Niels da Vitoria Lobo
Computer Science Department
University of Central Florida
Orlando, FL 32816
e-mail: {myles, niels}@cs.ucf.edu

Abstract

There are at least two situations in practical computer vision where displacement of a point in an image is accompanied by a defocus blur. The first is when a camera of limited autofocal capability moves in depth, and the second is when a limited autofocal camera zooms. Motion and zooming are two popular strategies for acquiring more detail or for acquiring depth. The defocus blur has been considered noise or at best been ignored. However, the defocus blur is in itself a cue to depth, and hence we proceed to show how it can be calculated simultaneously with affine motion. We first introduce the theory, then develop a solution method and finally demonstrate the validity of the theory and the solution by conducting experiments with real scenery.

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. To elaborate the first situation, consider a camera of fixed focal length. When the camera's motion has a component along the line of sight, some scene points move into focus, and some move out. In either case, between two images in a stop-and-shoot sequence, the fixed focal length leads to blurring and de-blurring of feature points. 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 [8][6][10], to compute the real center of the image [6] and to generally capture greater detail of a scene without moving. We term this situation *defocussed zoom*.

Measuring the level of blur/de-blur 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/de-blur, 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 [12][14][3][15][11] 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. We propose that one should simultaneously compute the level of blur and the affine motion that features undergo.

Older techniques for optical flow computation have been analyzed and compared [1]. Recent techniques include those which handle large affine transformations [9] and those which extend the affine model to include second order transformations of moving planar patches [5]. Despite all these works, illumination changes, photometric motion effects, and image features moving in and out of focus when scene elements move in depth, cause problems that are not yet robustly manageable. It is the last of these problematic effects, i.e., the changes in defocus due to motion, that will be addressed here.

In this paper, we compute affine parameters as well as blur simultaneously from a pair of input images. We first introduce the theoretical model of blur and affine motion. Then we propose an iterative solution method. We demonstrate the validity of our theory and the proposed solution by reporting experimental results with real scenery. Note that in this paper 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[7] 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. 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 approxi-

mation 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 other words the optical flow $[u,v]^T$ in that case can be expressed as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1+b_{11} & b_{12} \\ b_{21} & 1+b_{22} \end{bmatrix} \vec{r} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}$$

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

$$I_1(\vec{r}) = I_a(\mathbf{A}\vec{r})$$

[9] 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 $G(\mathbf{A}\vec{r}, \sigma^2\mathbf{A}\mathbf{A}^\mathbf{T})$ (ie. the Gaussian deformed by an affine transformation $\mathbf{A}$) is

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

In order to solve for the affine parameters in $\mathbf{A}$ an overdetermined system can be obtained by convolving at several points $\vec{l_i}$ in both images [9] to get

$$\int I_1(\vec{r}) G(\vec{r} - \vec{l_i}, \sigma^2\mathbf{I}) d\vec{r} \;=\; \int I_a(\mathbf{A}\vec{r}) G(\mathbf{A}\vec{r} - \vec{l_i}, \sigma^2\mathbf{A}\mathbf{A}^\mathbf{T})\; d(\mathbf{A}\vec{r}) +$$
$$[(\mathbf{A}-\mathbf{I})\vec{l_i}]^T \int I_a(\mathbf{A}\vec{r}) G'(\mathbf{A}\vec{r} - \vec{l_i}, \sigma^2\mathbf{A}\mathbf{A}^\mathbf{T})\; d(\mathbf{A}\vec{r}) \quad (2)$$

where l_i is any point on the image plane and G' is the derivative of G with respect to the image coordinates. Eqn. (2) is Taylor's first order expansion of $G(\mathbf{A}(\vec{r} - \vec{l_i}), \sigma^2\mathbf{A}\mathbf{A}^\mathbf{T})$ about the point $(\mathbf{A}\vec{r} - \vec{l_i})$.

Eqn. (2) cannot be used to solve for the unknown $\mathbf{A}$ since G on the right hand side cannot be evaluated. Hence we rewrite the above in a linear form of the unknown $\mathbf{B}$. To achieve that we express the generalized Gaussian in terms of the ordinary Gaussian and its derivatives using Taylor's expansion of eqn. (2) about the matrix $\mathbf{B} = \mathbf{0}$ (which corresponds to no affine transformation) where

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

The symbolic processing language Macsyma [2] was used to expand eqn. (1) up to first order terms in $\mathbf{B}$ about the point $\mathbf{B}$=$\mathbf{0}$ to yield

$$G(.,\sigma^2\mathbf{A}\mathbf{A}^\mathbf{T}) \approx G(.,\sigma^2) + \sigma^2 b_{11} G_{xx}(.,\sigma^2)$$
$$+\sigma^2 b_{12} G_{xy}(.,\sigma^2) + \sigma^2 b_{21} G_{yx}(.,\sigma^2) + \sigma^2 b_{22} G_{yy}(.,\sigma^2)$$

When the affine transformation is small, this is a reasonable approximation. Note that this is different from [9]'s equation(16) due to the σ^2 multipliers. Hence eqn. (2) can be revised as

$$\begin{aligned} I_1(\vec{r}) \otimes G(\vec{r} - \vec{l_i}, \sigma^2) \approx\; & I_a(\vec{r_1}) \otimes G(\vec{r_1} - \vec{l_i}, \sigma^2) \\ & +(\mathbf{B}\vec{l_i})^T I_a(\vec{r_1}) \otimes G'(\vec{r_1} - \vec{l_i}, \sigma^2) \\ & +\sigma^2 b_{11}\; I_a(\vec{r_1}) \otimes G_{xx}(\vec{r_1} - \vec{l_i}, \sigma^2) \\ & +\sigma^2 b_{12}\; I_a(\vec{r_1}) \otimes G_{xy}(\vec{r_1} - \vec{l_i}, \sigma^2) \\ & +\sigma^2 b_{21}\; I_a(\vec{r_1}) \otimes G_{yx}(\vec{r_1} - \vec{l_i}, \sigma^2) \\ & +\sigma^2 b_{22}\; I_a(\vec{r_1}) \otimes G_{yy}(\vec{r_1} - \vec{l_i}, \sigma^2) \end{aligned} \quad (3)$$

where $\vec{r_1} = \mathbf{A}\vec{r}$. Second order terms in the components of $\mathbf{B}$ are ignored since they are assumed to be small. The derivation obtained earlier in [9], and our eqn. (3) are now extended to include defocus blur. Denoting the left hand side of eqn. (3) by $\mathcal{I}_1(\vec{r})$ and the right hand side by $\mathcal{I}_a(\mathbf{A}\vec{r})$, we have

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

Now let us assume that image I_a is further transformed by a blur to yield another image I_2(see Fig. 1). It has been widely noted [4] [13] [11] that defocus blur can be modeled as the result of convolution with the Pill-box function (see Fig. 2) given by

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

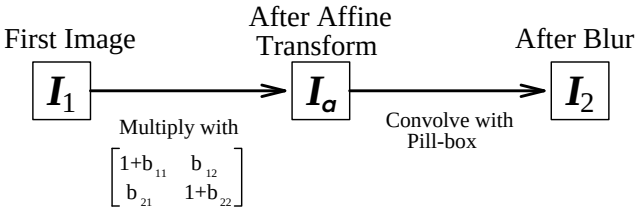


Figure 1: Conceptual Model Incorporating Affine Transformation and Defocus Blur.

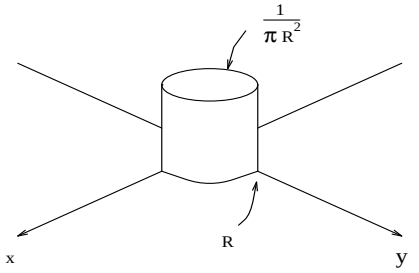


Figure 2: Pill-box Function

where increasing values of the radius R imply increasing levels of blur. We define the generalized Pill-box function $P(\vec{r}, R, \mathbf{A})$ as

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

where $|\mathbf{A}|$ denotes the determinant of $\mathbf{A}$. The above is a Pill-box deformed by an affine transformation $\mathbf{A}$. Using a simple mathematical substitution we see that

$$P(\vec{r} - \vec{l_i}, R, \mathbf{A}^{-1}) = |\mathbf{A}|\ P(\mathbf{A}(\vec{r} - \vec{l_i}), R, \mathbf{I}) \qquad (5)$$

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

$$\mathcal{I}_1(\vec{r})\ P(\vec{r} - \vec{l_i}, R, \mathbf{A}^{-1}) \approx \mathcal{I}_a(\mathbf{A}\vec{r})\ |\mathbf{A}|\ P(\mathbf{A}(\vec{r} - \vec{l_i}), R, \mathbf{I})$$

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

$$\begin{aligned} &\int \mathcal{I}_1(\vec{r})\ P(\vec{r} - \vec{l_i}, R, \mathbf{A}^{-1})\ d(\vec{r}) \\ &\quad \approx \int \mathcal{I}_a(\mathbf{A}\vec{r})\ |\mathbf{A}|\ P(\mathbf{A}(\vec{r} - \vec{l_i}), R, \mathbf{I})\ d(\vec{r}) \\ &\quad = \int \mathcal{I}_a(\mathbf{A}\vec{r})\ P(\mathbf{A}(\vec{r} - \vec{l_i}), R, \mathbf{I})\ d(\mathbf{A}\vec{r}) \qquad (6) \end{aligned}$$

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 eqn. (6) can be expressed as

$$LHS = I_1(\vec{r}) \otimes G(\vec{r} - \vec{l_i}, \sigma^2) \otimes P(\vec{r} - \vec{l_i}, R, \mathbf{A}^{-1})$$

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

$$\begin{aligned} LHS &= I_1(\vec{r}) \otimes G(\vec{r} - \vec{l_i}, \sigma^2) \otimes G(\vec{r}, R^2/2) \\ &= I_1(\vec{r}) \otimes G(\vec{r} - \vec{l_i}, \sigma^2 + R^2/2) \qquad (7) \end{aligned}$$

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

The right side of eqn. (6) can be expressed as

$$\begin{aligned} RHS &= P(\mathbf{A}(\vec{r} - \vec{l_i}), R, \mathbf{I}) \otimes \\ &\quad \Big[I_a(\vec{r_1}) \otimes G(\vec{r_1} - \vec{l_i}, \sigma^2) \\ &\quad + (\mathbf{B}\vec{l_i})^T I_a(\vec{r_1}) \otimes G'(\vec{r_1} - \vec{l_i}, \sigma^2) \\ &\quad + \sigma^2 b_{11}\ I_a(\vec{r_1}) \otimes G_{xx}(\vec{r_1} - \vec{l_i}, \sigma^2) \\ &\quad + \sigma^2 b_{12}\ I_a(\vec{r_1}) \otimes G_{xy}(\vec{r_1} - \vec{l_i}, \sigma^2) \\ &\quad + \sigma^2 b_{21}\ I_a(\vec{r_1}) \otimes G_{yx}(\vec{r_1} - \vec{l_i}, \sigma^2) \\ &\quad + \sigma^2 b_{22}\ I_a(\vec{r_1}) \otimes G_{yy}(\vec{r_1} - \vec{l_i}, \sigma^2) \Big] \\ &= I_2(\vec{r_1}) \otimes G(\vec{r_1} - \vec{l_i}, \sigma^2) \\ &\quad + (\mathbf{B}\vec{l_i})^T I_2(\vec{r_1}) \otimes G'(\vec{r_1} - \vec{l_i}, \sigma^2) \\ &\quad + \sigma^2 b_{11}\ I_2(\vec{r_1}) \otimes G_{xx}(\vec{r_1} - \vec{l_i}, \sigma^2) \\ &\quad + \sigma^2 b_{12}\ I_2(\vec{r_1}) \otimes G_{xy}(\vec{r_1} - \vec{l_i}, \sigma^2) \\ &\quad + \sigma^2 b_{21}\ I_2(\vec{r_1}) \otimes G_{yx}(\vec{r_1} - \vec{l_i}, \sigma^2) \\ &\quad + \sigma^2 b_{22}\ I_2(\vec{r_1}) \otimes G_{yy}(\vec{r_1} - \vec{l_i}, \sigma^2) \qquad (8) \end{aligned}$$

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 Pill-box function. Finally, equating eqns. (7) and (8) we get a relationship which is linear in the unknown $\mathbf{B}$, albeit still non-linear in R.

2.2 General Affine Transform Including Image Translation

We now consider the case when there is an arbitrary image translation $\delta\vec{T}$ along with the above affine transformation and blurring. An estimate of the translation can first be obtained by using a pyramid scheme. 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 $G(\vec{r} - \delta\vec{T}, .)$. 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

$$\begin{aligned} &I_1(\vec{r} + \vec{T_0} + \delta\vec{T}) \otimes G(\vec{r} - \vec{l_i}, \sigma^2 + R^2/2) \\ &\qquad = I_1(\vec{r} + \vec{T_0}) \otimes G(\vec{r} - \vec{l_i} - \delta\vec{T}, \sigma^2 + R^2/2) \end{aligned}$$

Substituting the above in eqn.s (7) and (8) we get

$$\begin{aligned} I_1(\vec{r}+\vec{T_0}) \otimes G(\vec{r}-\vec{l_i}-\delta\vec{T}, \sigma^2+R^2/2) \quad &\approx \\ & I_2(\vec{r_1}) \otimes G(\vec{r_1}-\vec{l_i}, \sigma^2) \\ & +(\mathbf{B}\vec{l_i})^T I_2(\vec{r_1}) \otimes G'(\vec{r_1}-\vec{l_i}, \sigma^2) \\ & +\sigma^2 b_{11}\ I_2(\vec{r_1}) \otimes G_{xx}(\vec{r_1}-\vec{l_i}, \sigma^2) \\ & +\sigma^2 b_{12}\ I_2(\vec{r_1}) \otimes G_{xy}(\vec{r_1}-\vec{l_i}, \sigma^2) \\ & +\sigma^2 b_{21}\ I_2(\vec{r_1}) \otimes G_{yx}(\vec{r_1}-\vec{l_i}, \sigma^2) \\ & +\sigma^2 b_{22}\ I_2(\vec{r_1}) \otimes G_{yy}(\vec{r_1}-\vec{l_i}, \sigma^2) \quad (9) \end{aligned}$$

Thus we now have an eqn.(9) relating the affine unknowns, $\mathbf{B}$ and $\delta\vec{T}$ and the level of blur R. To solve this equation, we linearize in $\delta\vec{T}$ and subsequently in R. To facilitate the solution of this equation linearization is accomplished next. Using the relation that

$$G_1(\vec{r}-\vec{l_i}-\delta\vec{T}, .) \approx G_1(\vec{r}-\vec{l_i}, .) - \delta\vec{T}^T G'(\vec{r}-\vec{l_i}, .)$$

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

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

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

$$Let \quad \sigma^2 + \frac{R^2}{2} = \eta_2 = \eta_1 + \beta$$

where η_1 is the current best estimate of η and β is the residual error in the estimation. $G_1(\vec{r}, \eta_2)$ can be evaluated by expanding by Taylor's series about the point η_1, to give

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

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

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

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

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 $\mathbf{B}$, the residual translation $\delta\vec{T}$ and the radius of blur R.

Note that 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 sub-pixel accuracy.

A pyramid scheme can be used to find a rough estimate of translation and at every level of the pyramid increasingly better values of the translation can be obtained.

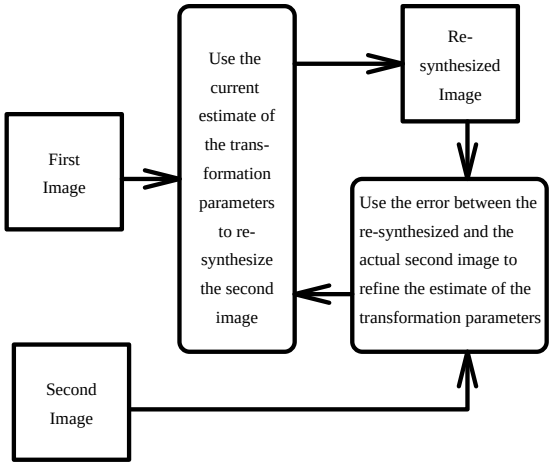


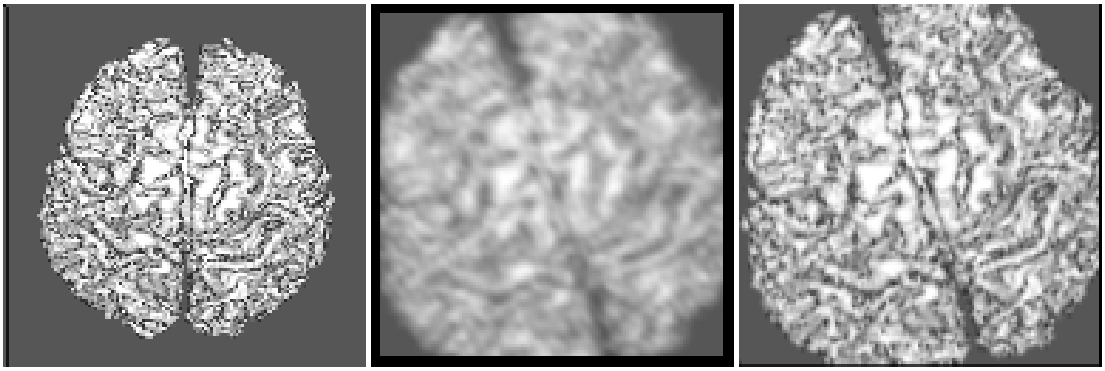
Figure 3: Overview of the Iterative Parameter Estimation Process

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 intermediate synthetic image is then obtained by transforming the first image using the computed affine parameters and then blurring the latter using the Pill-box 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.

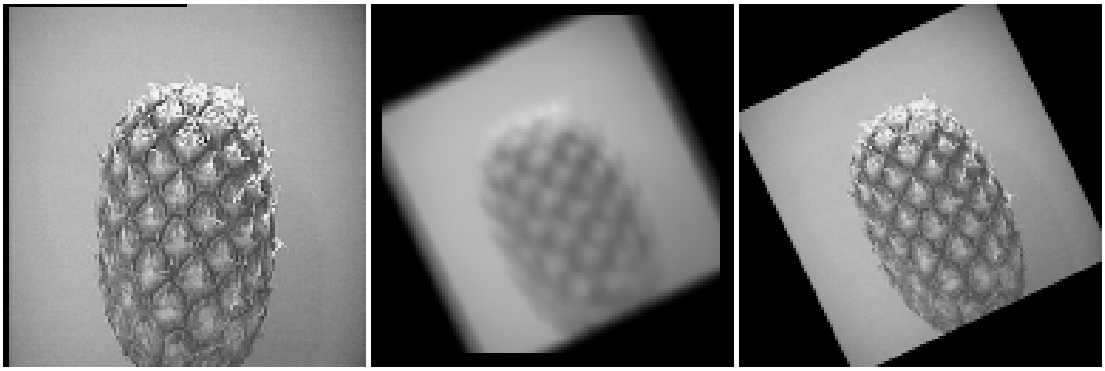
Since the linearization approximation is valid when the deformations are small and the method has to iteratively recover smaller and smaller values of the unknowns, the above method converges.

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, $\mathbf{B}$ is updated as $\mathbf{B}_{new} \leftarrow \mathbf{B}_{residual} \cdot \mathbf{B}_{previous}$ (matrix multiplication) and R is updated as

$$R \leftarrow \sqrt{2}\sqrt{\eta_1 + \beta - \sigma^2} \quad (10)$$



(a) Original Image (b) Artificially Deformed (c) Regenerated Using Recovered Affine Parameters



(d) Original Image (e) Artificially Deformed (f) Regenerated Using Recovered Affine Parameters

Figure 4: Artificially Transformed Real Images: (a) - (c) Brain, (d) - (f) Pineapple.

and a new value of η_1 is generated as $\eta_1 = \sigma^2 + \frac{R^2}{2}$.

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.

4.1 Artificially Deformed Images

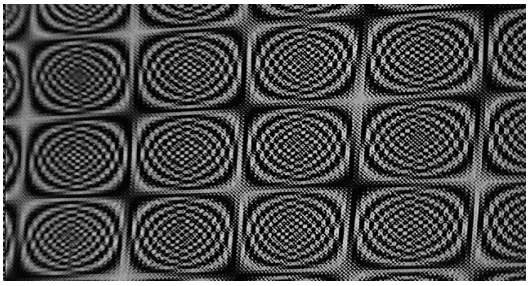
We ran several experiments on a wide range of test images. We artificially deformed them by performing an affine transform (expansion factors ranging from 0.7 to 1.4, rotations up to 30 degrees and image translations within 4 pixels) followed by a blurring operation. The program correctly recovered all parameters. The first image was then transformed according to the parameters recovered with the effect of blurring removed. This method is thus an effective way of de-blurring images from a sequence of images. It is enough if the first image in the sequence is sharp. All the others may have varying levels of blur. Results for two sets of images are shown in Fig. 4. An image of a brain (Fig. 4(a)) 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 and translation of $[-1., .7]^T$) and Pill-box blurred with radius R=3.5 pixels. The artificially transformed image is shown in (Fig. 4(b)). 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. 4(c) has been generated from the first image using the recovered affine parameters at the center of the image. Similarly, Figs. 4(d), (e) and (f) describe an experiment done with a photograph of a pineapple. The affine transformation involved is $\begin{bmatrix} 0.7250 & -0.3381 \\ 0.3381 & 0.7250 \end{bmatrix}$ and blur radius is 4.5 pixels. The affine parameters were accurately recovered as $\begin{bmatrix} 0.7251 & -0.3380 \\ 0.3381 & 0.7252 \end{bmatrix}$ and the recovered radius of blur was 4.47 pixels.

4.2 Pairs of Real Images

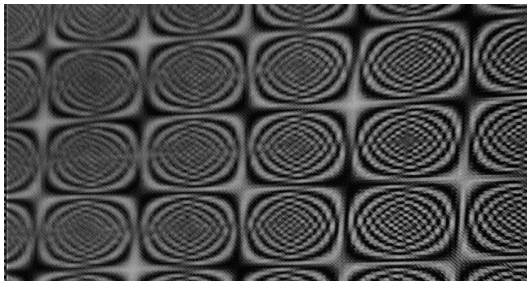
Experiments were then conducted on pairs of images of real scenery. Below we elaborate the results obtained in two of these experiments with real data.

Experiment 1: Geometrical Pattern

In order to facilitate computing the correct affine parameters using point correspondences, geometric



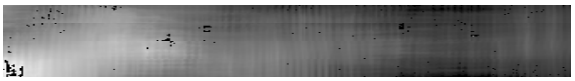
(a) First Image



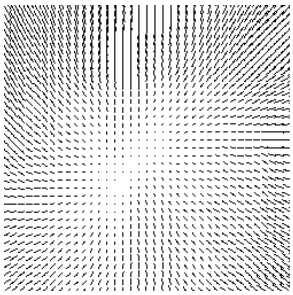
(b) Second Image

(c) Strip from first image

(d) Strip from same position of second image



(e) Radius map of above strip
(black patch in bottom left corner represents points of no convergence)



(f) Flow Field(41 x 41 pixel window) centered in the first image

Figure 5: Experiment 1: Slanted Geometric Pattern

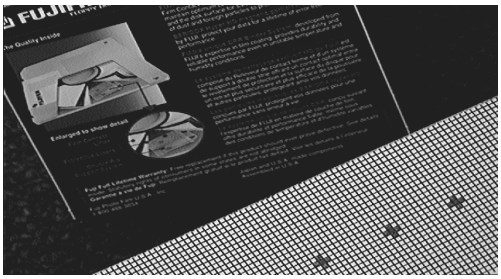
patterns were photographed with a digital camera. Results of the experiment are shown in Fig. 5. Fig. 5(a) shows the first image of a slanted geometrical pattern and Fig. 5(b) shows the second image, which is obtained by keeping the camera stationary and using the zoom mechanism.

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 pill-box blur of 4.4 pixels. Since degree of blur varies as one traverses the image from left to right, the affine parameters and the blur were obtained for a thin horizontal strip. Fig. 5(c) shows a strip from the first image and Fig. 5(d) shows the strip from the same position of the second. A map of the radii of blur in the above strip is shown in Fig. 5(e). The radius varied from 0.5 pixels obtained near the right edge to 7.0 pixels obtained near the left edge of the strip. The flow field in a 41x41 pixel window centered on the image is shown in Fig. 5(f).

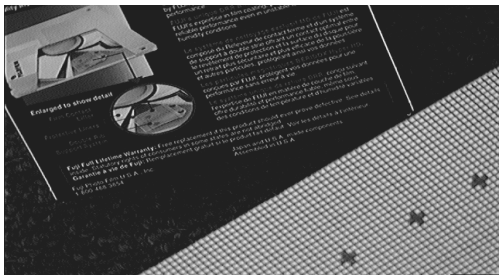
Experiment 2: Grid and Printed Boxes

Images were taken of a scene containing two objects, a box with a grid design with one side placed fronto-parallel with respect to the camera and a printed box above it placed at a very steep angle (see Fig. 6(a) and (b)). Each object was at a different depth with respect to the camera. The first image (Fig. 6(a)) was taken from above. The box with the grid design was then rotated and the camera was moved downwards towards the objects and the second image (Fig. 6(b)) obtained. The box with the grid design is uniformly blurred (due to it being fronto-parallel). The printed box on the other hand, is in sharper focus in certain areas. An estimate of the translation for a point on the grid box was obtained and the residual transformation at a 31x31 patch centered at that point was computed. Figs. 6(c) and (d) show the original patches in the two images. The affine transformation and blur were computed in a 31x31 area. Figs. 6(f) and (g) show the corresponding flow map (for the residual component) and the radius map computed within the patch.

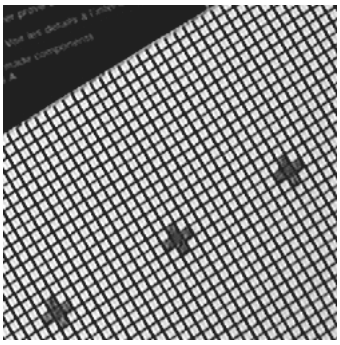
The affine transformation at the center of the patch was computed to be $\begin{bmatrix} 1.0633 & -0.1337 \\ 0.1375 & 1.0598 \end{bmatrix}$ and the radius of the blur was 2.7 pixels. The computed radius values in the map varied very slightly from 2.6 pixels to 2.8 pixels,as can be expected for a fronto-parallel plane. Fig. 6(e) shows the regenerated image for visual comparison. While Figs. 6(c)-(g) describe results for the grid box, results for the printed box part of the scene (Fig. 6(a) and (b)) are shown in Figs. 7(a)-(e). For the printed box, corresponding patches were extracted. For this box object, the second image is



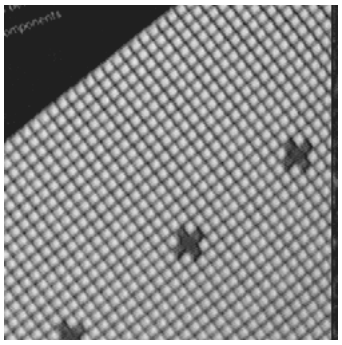
(a) First Image



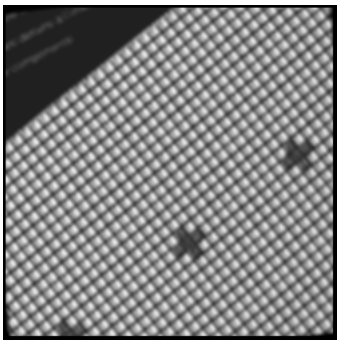
(b) Second Image



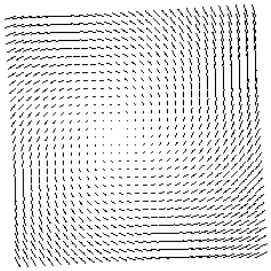
(c) Patch of first image



(d) Corresponding patch of second



(e) First regenerated using all recovered parameters



(f) Flow Field(31 x 31 pixel window)

(g) Radius Map(31 x 31 pixel window)

Figure 6: Experiment 3(a): Box with grid design

in sharper focus than the first. This phenomenon is detected automatically because the parameter in eqn. (10) becomes negative. In this case the algorithm reverses the order of the images and proceeds to compute the inverse transform and blur. Thus this method can be used to segment an image into areas which are blurred, those which are sharpened and those which maintain the same level of clarity. Results of the inverse transformation for this printed box are shown in Fig. 7. On close examination of the two original images it was seen that the level of blur in the upper part of the printed box is the same. However, in the lower part of the printed box, the second image is in sharper focus. This relative change in blur level can be noted in Fig. 7(e). White regions in the radius map represent points of non-convergence. In Figs. 5, 6, 7 since we do not have ground truth, we can only note qualitatively that the blur radius maps and the affine transformations appear correct.

Note that in this experiment the camera's aperture was deliberately set so that motion induced a large level of blur. This was done to demonstrate that the algorithm is able to handle large deformations. Also, in a depth-from-defocussed-motion system it may be useful to set up a stop-and-shoot sequence where alternate images are blurred so that the blurring information is used in the computation of depth.

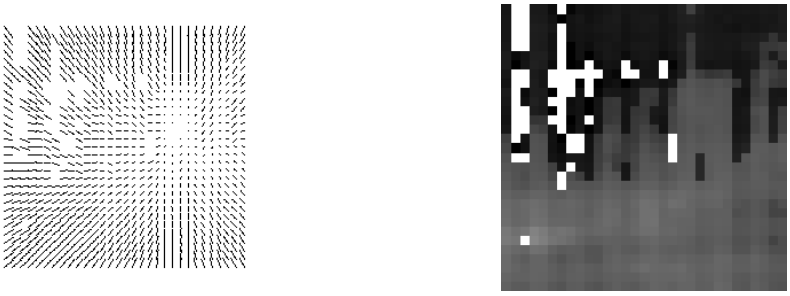
5 Discussion and Conclusion

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 [11] handles the 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 and remained focally unperturbed during the affine motion.

Our method's primary drawback is that it requires

(a) Patch of second image (b) Corresponding patch of first image (c) Second regenerated using all recovered parameters



(d) Flow Field(31 x 31 pixel window) (e) Radius Map(31 x 31 pixel window) (White region represents points of no convergence)

Figure 7: Experiment 3(b): Disk box

the existence of fairly large planar patches in the images and future work will address this.

Acknowledgments

We thank Niels Häring for extensive proof-reading. Both authors receive support from NSF grant CCR-9410459.

References

[1] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *IJCV*, 12:43–77, 1994.

[2] R. Bogen. *Macsyma Reference Manual*. Cambridge, Mass., Symbolics Incorporated, 1983.

[3] J. Ens and P. Lawrence. A matrix method for determining depth from focus. *CVPR*, pp:600–606, 1991.

[4] B. Horn. *Robot Vision*. McGraw-Hill, New York, 1986.

[5] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization. *CVPR*, pp:454–460, 1994.

[6] J.-M. Lavest, G. Rives, and M. Dhome. Three dimensional reconstruction by zooming. *IEEE Trans. on Robotics and Automation*, 9:196–207, 1993.

[7] H. Liu, T. Hong, M. Herman, and R. Chellappa. A generalized motion model for estimating optical flow using 3-d hermite polynomials. *ICPR-A*, pp:361–366, 1994.

[8] J. Ma and S. Olsen. Depth from zooming. *J. Optical Society of America*, 7:no.4:1883–1890, 1990.

[9] R. Manmatha. A framework for recovering affine transforms using points, lines, or image brightness. *CVPR*, pp:141–146, 1994.

[10] B. G. Mobasseri and S. Doraiswamy. Virtual motion: 3-d scene recovery using focal length-induced optic flow. *IEEE International Conference on Image Processing*, 3:78–82, 1994.

[11] S. Nayar, M. Watanabe, and M. Noguchi. Real-time focus range sensor. *ICCV*, pp:995–1001, 1995.

[12] A. Pentland. A new sense for depth of field. *IEEE Trans.PAMI*, 9:522–531, 1987.

[13] M. Subbarao. Focussed image recovery from two defocussed images recorded with different camera settings. Tech. 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. *CVPR*, pp:61–67, 1993.

[15] Y. Xiong and S. Shafer. Depth from focusing and defocusing. *IUW*, pp:967–976, 1993.