M3.13

AFFINE MODELS FOR IMAGE MATCHING AND MOTION DETECTION

Chiou-Shann Fuh and Petros Maragos

Division of Applied Sciences, Harvard University, Cambridge, MA 02138, USA

ABSTRACT: A model is developed for detecting the displacement field in spatio-temporal image sequences that allows for affine shape deformations of corresponding spatial regions and for affine transformations of the image intensity range. This model includes the block matching method as a special case. A least-squares algorithm is used to find the model parameters. It is experimentally demonstrated that the affine matching model performs better than other standard approaches. The resulting 2-D motion estimates are then used by a 3-D affine model and a least-squares algorithm that recover 3-D rigid body motion and depth from two perspective views.

1 Introduction

Motion detection is a very important problem both in video image coding and in computer vision. In video coding, motion detection is a necessary task for motion-compensated predictive coding and motion-adaptive frame interpolation to reduce the required channel bandwidth. In computer vision systems, motion detection can be used to infer the 3-D motion and surface structure of moving objects with many applications to robot guidance and remote sensing.

There is a vast literature on motion detection; see [1][7] for reviews. The major approaches to computing displacement vectors for corresponding pixels in two time-consecutive image frames can be classified as either using gradient-based methods (which include pixel-recursive algorithms) [2][4][5][8], or correspondence of motion tokens [3][9], or block matching methods [10][7].

Let $I(x, y, t)$ be a *spatio-temporal intensity image signal* due to a moving object, where $p = (x, y)$ is the (spatial) pixel vector. A well-known method to estimate 2-D velocities or pixel displacements on the image plane is the *block matching* method, where

$$E(d) = \sum_{p \in R} |I(p, t_1) - I(p + d, t_2)|^2$$

is minimized over a small spatial region R to find the *optimum displacement vector* d. Minimizing $E(d)$ is closely related to finding d such that the correlation $\sum_{p \in R} I(p, t_1) I(p + d, t_2)$ is maximized; thus, it is sometimes called the *area correlation* method. This approach has been negatively criticized because (i) it is computation-intensive; (ii) it ignores that the region R, which is the projection of the moving object at time $t = t_1$, will correspond to another region R' at $t = t_2$ with deformed shape due to foreshortening of the object surface regions as viewed at two different time instances; (iii) the image signals corresponding to regions R and R' do not only differ with respect to their supports R and R', but also undergo amplitude transformations due to the different lighting and viewing geometries at t_1 and t_2. Nowadays, (i) is not critical any more due to the availability of very fast hardware or parallel computers, but (ii) and (iii) are serious drawbacks. Several researchers have adopted other methods that depend either on (a) constraints among spatio-temporal image gradients, or on (b) tracking features (e.g., edges, blobs). However, (a) performs badly for medium- or long-range motion and is sensitive to noise. (b) is more robust in noise and works for longer-range motion, but feature extraction and tracking is a difficult task and gives sparse motion estimates. By comparison, if problems (ii) and (iii) can be solved, then the block matching method has the advantages of more robustness over (a) and denser motion estimates over (b).

In this paper, we present an improved model for block matching that solves problems (ii) and (iii) by allowing R to undergo affine shape deformations (as opposed to just translations that the block matching method assumes) and by allowing the intensity signal I to undergo affine amplitude transformations. The parameters for this affine model are found via a least-squares algorithm. Several experiments are reported that demonstrate the superiority of our affine model for image matching and motion detection over gradient-based, feature-tracking, or standard block matching methods. Finally, we apply the previous results to recovering the 3-D rigid body motion parameters and depth from two perspective views by using a 3-D affine model whose input 2-D motion correspondences are the displacement vectors that resulted from our affine matching model.

2 Affine Model for Image Matching

We assume that the region R' at $t = t_2$ has resulted from the region R at $t = t_1$ via an *affine* shape deformation $p \mapsto Mp + d$, where

$$Mp + d = \begin{bmatrix} s_x \cos\theta_x & -s_y \sin\theta_y \\ s_x \sin\theta_x & s_y \cos\theta_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

The vector $d = (d_x, d_y)$ accounts for spatial translations, whereas the 2×2 real matrix M accounts for rotations and scalings (compressions or expansions). That is, s_x, s_y are the scaling ratios in the x, y directions, and θ_x, θ_y are the corresponding rotation angles. These kinds of region deformations occur in a moving image sequence. For example, when objects rotate relative to the camera, the region R also rotates. When objects move closer or farther from the camera, the region R gets scaled (expanded or compressed). Displacements by d can be caused by translations

This research was supported by the National Science Foundation under Grant MIPS-86-58150 with matching funds from Bellcore, DEC, TASC, and Xerox, and in part by the Army Research Office under Grant DAAL03-86-K-0171 to the Center for Intelligent Control Systems.

CH2977-7/91/0000-2409 $1.00 © 1991 IEEE

of objects parallel to the image plane as well as by rotations. In addition, we allow the image intensities to undergo an *affine* transformation $I \mapsto rI + c$, where the ratio r adjusts the image amplitude dynamic range and c is a brightness offset. These intensity changes can be caused by different lighting and viewing geometries at time t_1 and t_2.

Thus, given $I(x, y, t)$ at $t = t_1, t_2$, and at various image locations, we select a small analysis region R and find the optimal parameters M, d, r, c that minimize the error functional

$$E(M, d, r, c) = \sum_{p \in R} |I(p, t_1) - rI(Mp + d, t_2) - c|^2$$

The optimum d provides us with the displacement vector. As by-products, we also obtain the optimal M, r, c which provide information about rotation, scaling, and intensity changes. We call this approach the *affine model for image matching*. Note that the standard block matching method is a special case of our affine model, corresponding to an identity matrix M, $r = 1$, $c = 0$. Although d is a displacement vector representative for the whole region R, we can obtain dense displacement estimates by repeating this minimization procedure at each pixel, with R being a small surrounding region. Note that, if R is a square region, its corresponding region R' under the map $p \mapsto Mp + d$ will generally be a rotated and translated parallellogram. More general shape/intensity transformations can be modeled by a *sum* of affine maps, i.e., $I(p, t_1) \mapsto c + \sum_n r_n I(M_n p + d_n, t_2)$, as developed in [6].

Finding the optimal M, d, r, c is a nonlinear optimization problem. While it can be solved iteratively by gradient steepest descent in an 8-D parameter space, this approach cannot guarantee convergence to a global minimum. Alternatively, in our work we propose the following algorithm that provides a closed-form solution for the optimal r, c and iteratively searches a quantized parameter space for the optimal M, d. We find first the optimal r, c by setting $\partial E / \partial r = 0$ and $\partial E / \partial c = 0$. This yields two linear equations in r, c which can be easily solved to find the optimal r^* and c^* as functions of M and d:

$$r^* = \left[A \sum I_1 I_2 - \sum I_1 \sum I_2\right] / \left[A \sum I_2^2 - (\sum I_2)^2\right]$$
$$c^* = \left[\sum I_1 \sum I_2^2 - \sum I_2 \sum I_1 I_2\right] / \left[A \sum I_2^2 - (\sum I_2)^2\right]$$

where $I_1 = I(p, t_1)$, $I_2 = I(Mp + d, t_2)$, $\sum = \sum_{p \in R}$, and A is the area of the region R. Replacing the optimal r^*, c^* into E yields the error functional

$$E^*(M, d) = E(M, d, r^*, c^*) = \sum I_1^2 - r^* \sum I_1 I_2 - c^* \sum I_1$$
$$= \sum I_1^2 - \underbrace{\frac{A(\sum I_1 I_2)^2 + \sum I_2^2 (\sum I_1)^2 - 2 \sum I_1 \sum I_2 \sum I_1 I_2}{A \sum I_2^2 - (\sum I_2)^2}}_{K(M, d)}$$

Since the term $\sum_p I_1^2$ is independent of M, d, minimizing $E^*(M, d)$ is equivalent to maximizing $K(M, d)$. The function $K(M, d)$ consists of several correlation terms. Now by discretizing the 6-D parameter space M, d and exhaustively searching a bounded region we find the optimal M, d that maximize $K(M, d)$. (The 2-D parameter subspace d is inherently discrete because it represents integer pixel coordinates.)

In our experiments we assume that M performs a uniform rotation by $\theta = \theta_x = \theta_y$ and uniform scaling by $s = s_x = s_y$, and d is constrained to be within an $L \times L$ window around p, where $L/2$ is the maximum expected displacement in each direction. Thus we search in a finite discrete 4-D parameter space s, θ, d. Figure 1 reports several motion detection experiments where the motion is a short- or long-range translation or rotation and the scene lighting changes. These experiments show that our affine matching algorithm performs better than other standard approaches such as block matching, gradient methods [4] or feature-tracking methods [3]. However, the superior performance of our affine model comes at a higher computational complexity.

Although the affine matching algorithm usually yields robust displacement estimates, there may be a few mismatches which we view as noise. In this case additional improvement can be achieved by smoothing the displacement vector field. We exclude the use of linear filtering (e.g., local averaging) because linear smoothing filters have the well-known tendency to blur and shift sharp discontinuities in signals. Sharp discontinuities in the displacement field may indicate object boundaries and, hence, must be preserved. Instead we choose spatio-temporal vector median filtering because the scalar median filter can eliminate outliers while preserving abrupt edges. Vector median filtering is defined to be the x, y componentwise median filtering: $med\{d_i\} = (med\{d_{x,i}\}, med\{d_{y,i}\})$, where d_i, $i = 1, 2, ..., n$, are the displacement vectors in a spatio-temporal cube surrounding the center of region R and time t_1. We have found this vector median to perform well in smoothing velocity fields.

Our affine matching algorithm performs well not only on rigid objects but also on nonrigid objects, such as moving clouds where the interframe changes of object shapes could be very large; Figure 2 shows an example.

3 Recovery of 3-D Motion and Depth

The displacement vectors from the affine matching algorithm can be applied to recover the rigid body motion parameters (including 3-D rotations and translations) and surface structure (the depth of the objects with respect to the camera) from two perspective views. Let (X, Y, Z) and (X', Y', Z') be the coordinates of a point on a rigid object before and after rotation $(\theta_x, \theta_y, \theta_z)$ and translation (T_x, T_y, T_z) and (x, y) and (x', y') be the coordinates of the projection of the point on the image plane before and after rigid motion. Similarly, we can assume the objects in the scene are stationary and the camera undergoes translation and rotation. Assume a perspective projection where the origin is the center of projection and the image plane is the $z = 1$ plane. So all the distance units are expressed in terms of the distance from projection center to image plane. Thus we have $x = \frac{X}{Z}, y = \frac{Y}{Z}$ and $x' = \frac{X'}{Z'}, y' = \frac{Y'}{Z'}$. Rigid body motions can always be represented by a rotation followed by a translation. (Here the rotation is in the order $\theta_x, \theta_z, \theta_y$, but other orders can be solved similarly.) The following abbreviations are used: $C_x = cos\theta_x, S_x = sin\theta_x, C_y = cos\theta_y, S_y = sin\theta_y, C_z = cos\theta_z, S_z = sin\theta_z$.

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} C_y & 0 & S_y \\ 0 & 1 & 0 \\ -S_y & 0 & C_y \end{bmatrix} \begin{bmatrix} C_z & -S_z & 0 \\ S_z & C_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_x & -S_x \\ 0 & S_x & C_x \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

Expanding the above equations and dividing both numerators and denominators by Z yields

$$x' = \frac{C_z C_y x + (S_x S_y - C_x C_y S_z) y + (C_x S_y + S_x C_y S_z + T_x/Z)}{-S_y C_z x + (C_y S_x + C_x S_y S_z) y + (C_x C_y - S_x S_y S_z + T_z/Z)} \quad (1)$$

$$y' = \frac{S_z x + C_x C_z y - S_x C_z + T_y/Z}{-S_y C_z x + (C_y S_x + C_x S_y S_z) y + (C_x C_y - S_x S_y S_z + T_z/Z)} \quad (2)$$

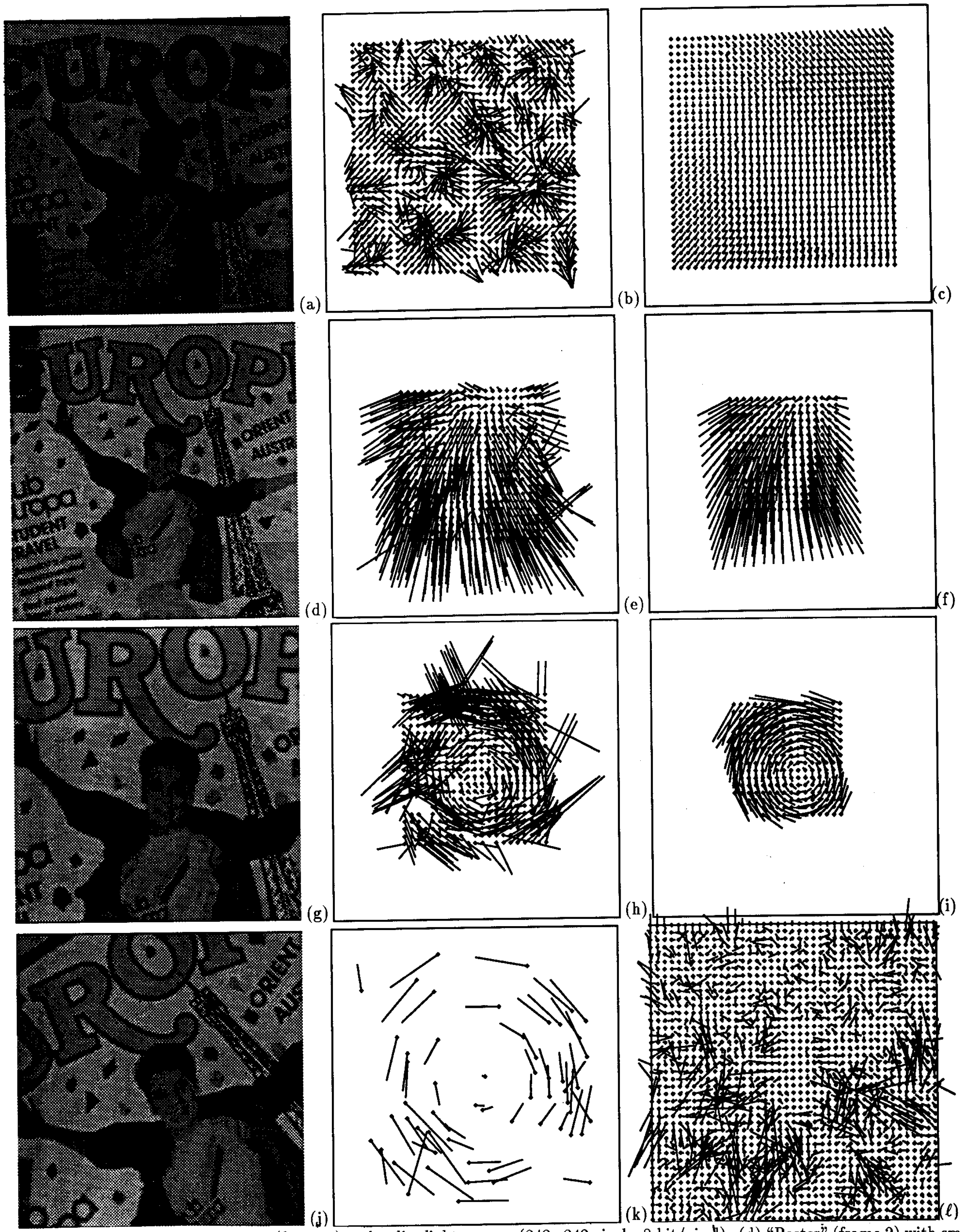

Figure 1: (a) "Poster" image sequence (frame 1) under dim light sources (242×242 pixels, 8-bit/pixel). (d) "Poster" (frame 2) with small rotation and under much brighter light sources. Displacement vectors between images 1a and 1d using (b) standard block matching and (c) the affine matching algorithm. (g) "Poster" (frame 3) after camera moved closer to the object. Displacement vectors between images 1d and 1g using (e) standard block matching and (f) the affine matching algorithm. (j) "Poster" (frame 4) after a 23° counterclockwise rotation. Displacement vectors between images 1g and 1j using: (h) standard block matching, (i) the affine matching algorithm, (k) a feature-based correspondence algorithm [3], and (ℓ) a gradient-based optical flow algorithm [4].

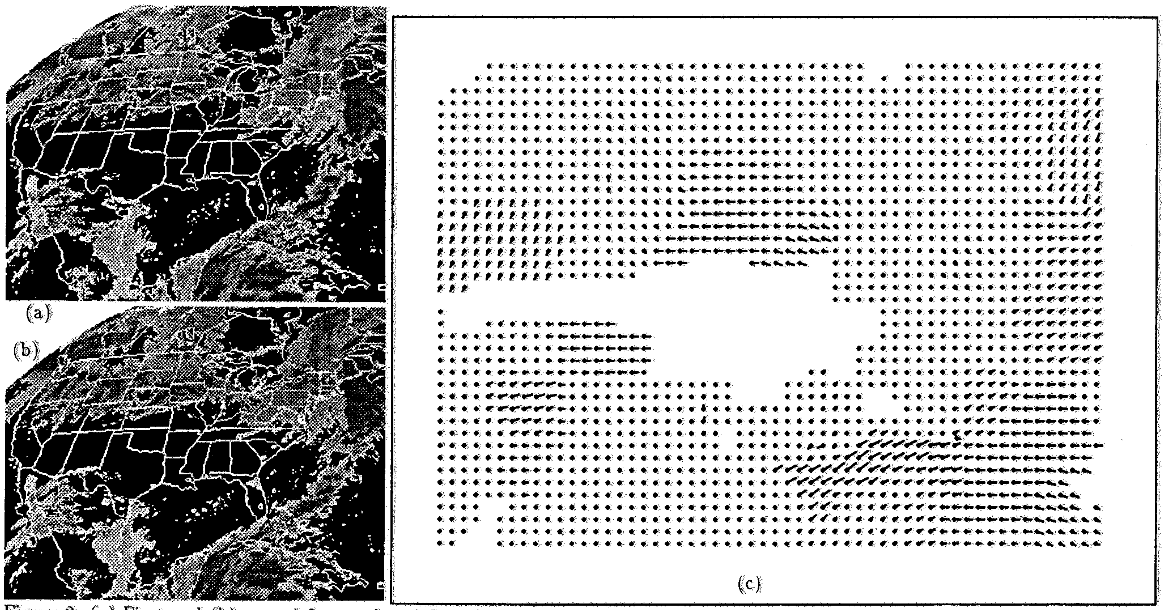

Figure 2: (a) First and (b) second frame of an infrared cloud image sequence (240×320 pixels, 4-bit/pixel where pixel intensity is the cloud top altitude). (c) Displacement vectors from affine matching algorithm, smoothed by a spatio-temporal vector median filter.

Assuming $T_x = T_y = 0, T_z \neq 0$ (other cases for nonzero translation in only one axis can be solved similarly) and canceling T_z/Z gives $S_xC_zx' - S_zxx' + C_yC_zxy' + (S_xS_y - C_xC_yS_z)yy' + (C_xS_y + S_xC_yS_z)y' - C_xC_zyx' = 0$. We can write this as

$$yx' = Ax' + Bxx' + Dxy' + Eyy' + Fy'$$

where $A = S_x/C_x, B = -S_z/C_xC_z, D = C_y/C_x,$

$$E = \frac{S_xS_y - C_xC_yS_z}{C_xC_z}, \quad F = \frac{C_xS_y + S_xC_yS_z}{C_xC_z}.$$

Now we set up an overdetermined system of equations $\Psi\alpha = \beta$ where n is the number of correspondence pairs, $\Psi_{n\times5}$ and $\beta_{n\times1}$ consists of n rows of $(x_i', x_ix_i', x_iy_i', y_iy_i', y_i')$ and $(y_ix_i'), 1 \leq i \leq n$ respectively and $\alpha_{5\times1} = (A, B, D, E, F)$. By using the pseudo-inverse approach, we can find a least-squares solution of $\Psi\alpha = \beta$ and thus obtain the parameters A, B, D, E, F. Then we can find the rotation angles by $\theta_x = tan^{-1}A, \theta_z = tan^{-1}(-BC_x)$,

$$\theta_y = sin^{-1}(\frac{FC_xC_z}{\sqrt{C_x^2 + S_x^2S_z^2}}) - tan^{-1}(\frac{S_xS_z}{C_x}).$$

Depth up to a scaling factor $\frac{Z}{T_z}$ can be computed by substituting rotation angles back into (2) (or (1)) thus

$$\frac{Z}{T_z} = y'/[S_zx + C_xC_zy - S_xC_z$$
$$+S_yC_zxy' - (C_yS_x + C_xS_yS_z)yy' - (C_xC_y - S_xS_yS_z)y']$$

We next present the results from this approach to 3-D motion recovery for a simulated example of 8 object points.

object points (X,Y,Z) : $(92, 18, 84), (41, 64, 48), (9, 9, 50),$
$(9, 50, 9), (50, 9, 9), (26, 35, 49), (32, 7, 10), (62, 16, 36)$
applied rotation: $\theta_x = -1.0^o, \theta_y = 2.0^o, \theta_z = -3.0^o$
applied translation: $T_x = T_y = 0.0, T_z = 6.0$
—recovered parameters—
(A,B,D,E,F) : $(-0.01746, 0.05242, 0.99954, 0.05177, 0.03586)$
rotation: $\theta_x = -1.00^o, \theta_y = 2.00^o, \theta_z = -3.00^o$
depth $\frac{Z}{T_z}$: $(14.005, 8.00, 8.334, 1.500, 1.500, 8.167, 1.667, 6.001)$

References

[1] J.K. Aggarwal and N. Nandhakumar, "On the computation of Motion from Sequences of Images–A Review", *Proc. IEEE, 76*, pp.917-935, Aug. 1988.

[2] J. Biemond, J.N. Driessen, A.M. Geurtz, and D.E. Boekee, "A Pel-Recursive Wiener-Based Algorithm for the Simultaneous Estimation of Rotation and Translation," *Proc. SPIE 1001: Visual Comm. Image Process.*, pp. 917-924, 1988.

[3] C.S. Fuh and P. Maragos, "Region-Based Optical Flow Estimation", *Proc. IEEE Conf. CVPR*, San Diego, Jun 1989.

[4] B.K.P. Horn, and B.G. Schunck, "Determining Optical Flow", *Artificial Intelligence*, 17:185-203, Aug. 1981.

[5] D.S. Kalivas, A.A. Sawchuk, and R. Chellappa, "Segmentation and 2-D Motion Estimation of Noisy Image Sequences," *Proc. IEEE ICASSP'88*, pp. 1076-1079, NY, April 1988.

[6] P. Maragos, "Affine Morphology and Affine Signal Models", *Proc. SPIE 1350: Image Algebra and Morphological Image Processing*, pp.31-43, 1990.

[7] H.G. Musmann, P. Pirsch, and H-J Grallert, "Advances in Picture Coding," *Proc. IEEE*, vol. 73, pp. 523-548, 1985.

[8] A.N. Netravali and J.D. Robbins, "Motion compensated television coding– Part I," *Bell Syst. Tech. J.*, 58, 1979.

[9] R.Y. Tsai and T.S. Huang, "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces", *IEEE T-PAMI-6*, Jan. 1984.

[10] K.H. Tzou, T.R. Hsing, and N.A. Daly, "Block-Recursive Matching Algorithm(BRMA) for Displacement Estimation of Video Images," *Proc. IEEE ICASSP'85*, Tampa, 1985.