AFFINE MODELS FOR MOTION AND SHAPE RECOVERY

Chiou-Shann Fuh and Petros Maragos

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

Abstract

This paper presents an affine model for 3-D motion and shape recovery using two perspective views and their relative 2-D displacement field. The 2-D displacement vectors are estimated as parameters of a 2-D affine model that generalizes standard block matching by allowing affine shape deformations of image blocks and affine intensity transformations. The matching block size is effectively found via morphological size histograms. The parameters of the 3-D affine model are estimated using a least-squares algorithm that requires solving a system of linear equations with rank three. Some stabilization of the recovered motion parameters under noise is achieved through a simple form of MAP estimation. A multi-scale searching in the parameter space is also used to improve accuracy without high computational cost. Experiments on applying these affine models to various real world image sequences demonstrate that they can estimate dense displacement fields and recover motion parameters and object shape with relatively small errors.

1 Introduction

Visual motion analysis can provide rich information about the 3-D motion and surface structure of moving objects with many applications to vision-guided robots, video data compression, and remote sensing. There are two major problems in this area: the first is determining 2-D motion displacement fields from time sequences of intensity images. The second problem is to recover the motion parameters (3-D translations and rotations) and the surface structure (object depth relative to camera or retina) by using the estimated displacement field. There has been numerous previous and important work on visual motion analysis as summarized in [2, 13, 19].

The major approaches to estimating 2-D displacement vectors for corresponding pixels in two time-consecutive image frames can be classified as gradient-based methods, correspondence of motion tokens, and block matching methods. The gradient methods are based on constraints or relationships among the image spatial and temporal derivatives, e.g. [12]. A broad class of gradient methods are all the pixel-recursive algorithms, popular among video coding researchers [21, 22]. The correspondence methods consist of extracting important image features and tracking them over consecutive image frames. Examples of such features include isolated points, edges, and blobs [2, 3, 8, 24]. In block matching methods, blocks (i.e., subframes) in the previous image frame are matched with corresponding blocks in the current frame via criteria such as minimizing a mean squared (or absolute) error or maximizing a cross-correlation [14, 21]. The standard block matching does not perform well when the scenes undergo both shape deformations and illumination changes; thus various improved or generalized models have been proposed in [7, 10, 11, 15, 25]. Finally, there are also numerous approaches to 3-D motion and shape recovery. Most of them assume that 2-D velocity data (sparse or dense) have been obtained in advance. Examples of previous work related to our approach for 3-D motion recovery include [24, 26].

In this paper we present an integrated system to first determine 2-D motion displacement fields and then recover the 3-D motion parameters and surface structure. The unifying themes in our work are the use of affine models, both for 2-D and 3-D motion estimation, and of least-squares algorithms combined with a limited searching to estimate the parameters of these models. The usage of affine models has appeared in motion analysis and image processing in various useful ways [1, 5, 7, 11, 16, 18].

 0-8194-1018-7/92/$4.00

In Section 2.1 we review from [7] our 2-D affine model for estimating the displacement field in spatio-temporal image sequences, which allows for affine shape deformations of corresponding spatial regions and for affine transformations of image intensity range. The model parameters are found by using a least-squares algorithm. (In a related work [11] an adaptive least-squares correlation was proposed which allowed for local geometrical image deformations and intensity corrections (additive bias only) and a gradient descent algorithm was used to find model parameters.) In [7] we experimentally demonstrated that our affine block matching algorithm performs better in estimating displacements than standard block matching and gradient methods, especially for long-range motion with possible changes in scene illumination. In Section 2.2 we further refine our affine matching algorithm by using morphological size histograms to find an effective matching block size that, for each image frame pair, can be chosen to match the various characteristic object sizes present in the image frame and thus minimize displacement estimation errors.

In Section 3 we present a 3-D affine model that uses a least-squares algorithm to recover the 3-D rigid body motion parameters and surface structure based on two perspective views and given the 2-D displacement data estimated by the 2-D affine block matching. Our approach not only uses the redundancy inherent in the over-determined linear system to combat noise, but also uses MAP estimation to include prior information and to stabilize the parameters. Although the 3-D affine model is the same as used in [26], our approach for finding its parameters has the attractive feature of using a system of linear equations that has only rank three. In addition, our algorithm performs a multi-scale search of the discretized and bounded parameter space to avoid high computational cost and to achieve better accuracy. In the time domain, the recovered motion parameters can be smoothed by vector median filtering to reduce the noise when the motion remains constant or varies smoothly.

The proposed affine models are applied to time sequences of real world images and are shown to give displacement vectors, motion parameters, and surface structure with a small relative error.

2 Affine Block Matching Model

2.1 2-D Affine Model and a Least-Squared Algorithm

This section reviews a 2-D affine model and its associated least-squared algorithm for image matching and motion detection [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. Let a planar region R be the projection of the moving object at time $t = t_1$. At a future time $t = t_2$, R will correspond to another region R' with deformed shape due to foreshortening or rotations of the object surface regions as viewed at two different time instances. 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

$$Mp + d = \begin{bmatrix} s_x \cos\theta_x & -s_y \sin\theta_y \\ s_x \sin\theta_x & s_y \cos\theta_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix} \tag{1}$$

The vector $d = (d_x, d_y)$ accounts for spatial translations, whereas the 2×2 real matrix M accounts for rotations and scalings. 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. Translation, rotation, and scaling are region deformations that often occur in a moving image sequence. 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.

Given $I(p, t)$ 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

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

Table 1: *Displacement estimation errors with respect to block size (in pixels)*

$B \times B$	1×1	3×3	5×5	7×7	11×11	15×15	19×19	23×23
d_x error	46.4	21.3	7.1	1.6	0.5	0.3	0.3	0.3
d_y error	45.3	33.5	6.1	0.9	0.5	0.3	0.3	0.3

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 of 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.

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, we proposed in [7] 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 $\frac{\partial E}{\partial r} = 0$ and $\frac{\partial E}{\partial c} = 0$. Solving these two linear equations yields the optimal $r^\star$ and $c^\star$ as functions of M and d. Replacing the optimal $r^\star, c^\star$ into E yields a modified error functional $E^\star(M, d)$. Now by discretizing the 6-D parameter space M, d and exhaustively searching a bounded region we find the optimal $M^\star, d^\star$ that minimize $E^\star(M, d)$. The translation is restricted to be L pixels in each direction, i.e., $|d_x|, |d_y| \leq L$, and the region R at $t = t_1$ is assumed a square of $B \times B$ pixels. After having found the optimal $M^\star$ and $d^\star$, we can obtain the optimal $r^\star$ and $c^\star$ [7].

Figures 1(a),(b) show an original "Poster" image and a synthetically transformed image according to the affine model with a global translation of $d = (5, 5)$ pixels, rotation by $\theta = 6^o$, scaling $s_x = s_y = 1.2$, intensity ratio $r = 0.7$, and intensity bias $c = 20$. The center of the synthesized rotation and scaling is at the global center of the image. Figure 1(c) shows the displacement field estimated via the affine matching algorithm. In this experiment the searching range for the scaling was $s_x = s_y \in [0.8, 1.2]$ and for the rotation $\theta_x = \theta_y \in [-6^o, 6^o]$; also we had set $B = 19$ and $L = 40$.

2.2 Block Size Selection for 2-D Affine Matching

The selection of the block size B is important because if B is too small there is insufficient information in the analysis region to determine the affine model parameters and hence mismatches can occur. If the block size is too large, the matching is unnecessarily computationally expensive and the affine model cannot resolve small objects undergoing disparate motions within the region. As an example, the whole image in Fig. 1(a) is an affine transformation of the image in Fig. 1(b). As the block size increases, the block contains more information for determining the affine model parameters thus the error in d_x and d_y decreases. Table 1 and Figure 1(d) show that as the block size increases the number of mismatches decreases and vice-versa.

The size and shape of the objects in the image are natural criteria for the selection of the optimal block size B. Our approach is to obtain a binarized version X for the gray-level image frame and determine an optimum block size based on the shapes and sizes of the binary objects in X. The *morphological shape-size histogram* [17, 23], based on multiscale openings/closings and granulometries [20] and also called 'pattern spectrum' in [17], offers a good description of the shape and size information of the objects in the binary image X and is defined as follows:

$$\begin{aligned} SH_X(+n) &= A[X \circ nS] - A[X \circ (n+1)S], \quad n \geq 0 \\ SH_X(-n) &= A[X \bullet nS] - A[X \bullet (n-1)S], \quad n \geq 1 \end{aligned} \tag{3}$$

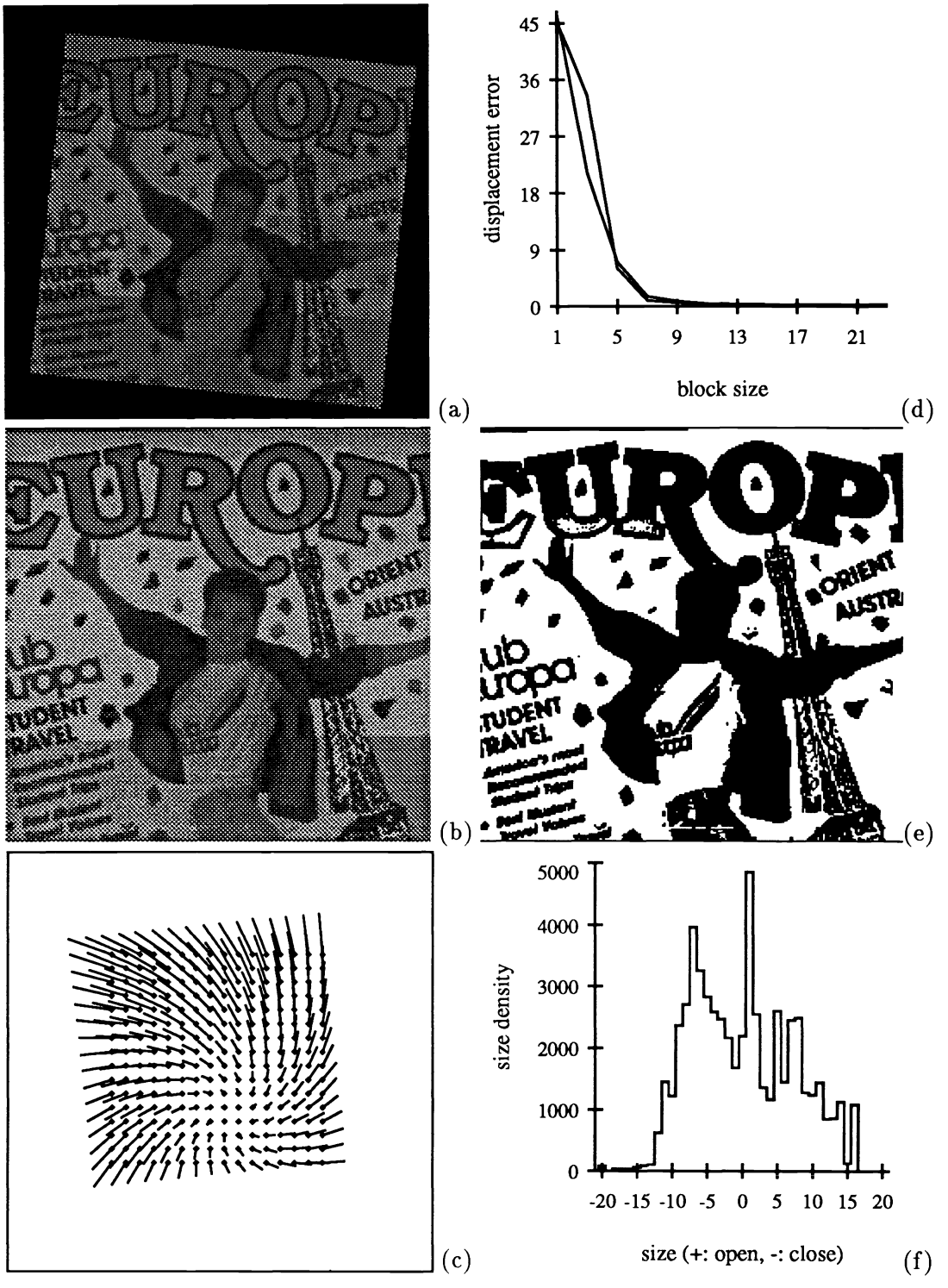

Figure 1: *Selection of block size B. (a) An affine transformed version of the image in (b) with translation* $d = (5, 5)$, *rotation* $\theta = 6^o$, *scaling* $s = 1.2$, *intensity ratio* $r = 0.7$, *and intensity bias* $c = 20$. *(b) The original "Poster" image. (242×242 pixels, 8-bit/pixel). (c) Result of matching (a) and (b) where block size is* 19×19. *(d) Errors of* d_x *and* d_y *(in pixels) with respect to varying block size. (e) Binarized image of (b). (f) Size histogram of (e) using a* 3×3 *square structuring element.*

where $A[\cdot]$ denotes area, and $X \circ nS$ and $X \bullet nS$ denote opening and closing of X by a structuring element S of size n. Large isolated spikes or narrow peaks in the size histogram, located at some positive (or negative) size n, indicate the existence of separate objects or protrusions in the foreground (or background) of the image X at that size n. In our experiments we use square analysis regions for image matching, and so we fix S to be a 3×3-pixel square.

We convert[1] a gray-tone image frame into a binary image X by thresholding at the median of the intensity values, so that to obtain approximately equal numbers of dark and bright pixels. Note that the opening and closing are dual operations on bright and dark pixels, and hence the size histogram will be more symmetrical if the binary image has approximately equal numbers of dark and bright pixels. The binary image thus generated is shown in Figure 1.(e) and its size histogram is shown in Figure 1.(f).

By using the size histogram and a heuristic rule for the selection of block size B we can avoid expensive multi-scale analysis to choose an "optimal" block size B_{opt} that minimizes the average displacement error. Since we have six parameters in our 2-D affine model, $(r, c, \theta, s, d_x, d_y)$ the block size B_{opt} cannot be less than a minimum size in order to have enough information in the analysis region; experimentally we found this minimum to be about 11. After some experimentation on various images, we found strong correlation between n_{max} and optimal block size B_{opt} where n_{max} is the size at which the size histogram assumes its maximum value over all sizes ≥ 11. As an example, Table 1 shows that the estimation errors in the displacements d_x and d_y (between the images in Figs. 1(a),(b)) achieve an asymptotic value of 0.3 pixels when $B \geq 15$. From the size histogram, the size which is not less than the minimum and which gives the maximum value of the size histogram is 7. Therefore, since the structuring element is a 3×3 square, the most common pattern size is $n_{max} = 2 \times 7 + 1 = 15$, which coincides with the optimum block size. Despite their strong correlation, an exact relationship between B_{opt} and n_{max} is difficult to find. In practice, we propose the following general heuristic rule for block size selection: $B_{opt} \approx n_{max} + 4$. We add this small constant (4) to n_{max} because the most common patterns will be smaller than the corresponding analysis region R and lie entirely inside R. Thus, for the example of Fig. 1 we finally selected $B = 19$. We have applied this heuristic rule to various images to approximately select the optimal block size B_{opt} and found that it performs well.

Overall, we have applied the affine block matching algorithm to various indoor and outdoor image sequences and the experimental results showed the algorithm is robust and gives dense and reliable displacement fields.

3 3-D Motion and Shape Recovery

After the 2-D displacement vector field is estimated, the next step is to use it to recover the rigid-body motion parameters and object shape. This section gives the details and experimental results of recovering 3-D motion parameters and surface structure under perspective projection via a 3-D affine model whose parameters are found using a least-squares algorithm.

3.1 3-D Affine Model and Least-Squares Algorithm

Assume a perspective projection where the origin is the center of projection and the image plane is the $Z = 1$ plane, as shown in Figure 2. Let (X, Y, Z) and (X', Y', Z') be the 3-D world coordinates of a point on objects before and after rigid motion. Let (x, y) and (x', y') be the coordinates of the projections of the

[1] We did not use any edge operator to convert a gray-tone image into a binary image, because a good edge detection requires pre-smoothing the image and the size of the smoothing kernel affects the size histogram.

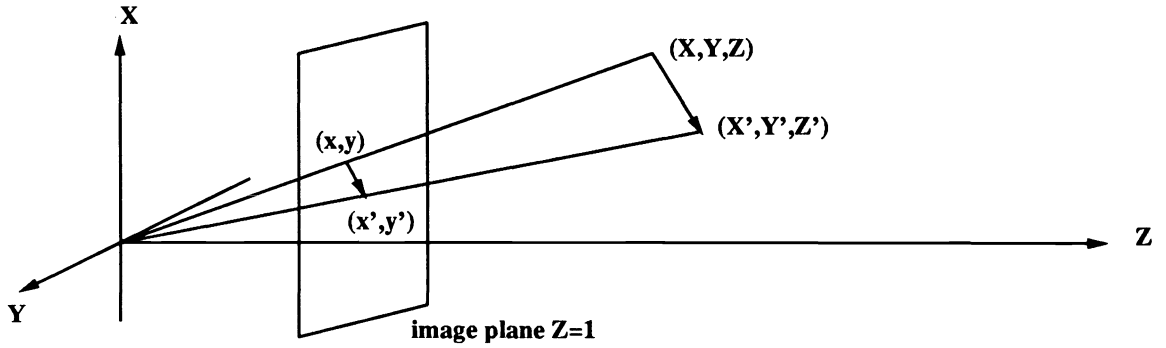

Figure 2: *Camera setup and the perspective projection.*

point on the 2-D image plane before and after the motion; thus we have:

$$x = \frac{X}{Z} \qquad x' = \frac{X'}{Z'} \qquad y = \frac{Y}{Z} \qquad y' = \frac{Y'}{Z'} \tag{4}$$

Rigid motion includes rotation by angles $\theta_x, \theta_z, \theta_y$ around their respective axes X, Z, Y in the given order (other orders can be solved similarly) and followed by translation (T_x, T_y, T_z), where the subscript denotes the corresponding axis along which the translation component is measured. Thus we have the 3-D affine model:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} C_y & 0 & S_y \\ 0 & 1 & 0 \\ -S_y & 0 & C_y \end{bmatrix} \begin{bmatrix} C_z & -S_z & 0 \\ S_z & C_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_x & -S_x \\ 0 & S_x & C_x \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{5}$$

$$\begin{aligned} X' &= C_z C_y X + (S_x S_y - C_x C_y S_z) Y + (C_x S_y + S_x C_y S_z) Z + T_x \\ Y' &= S_z X + C_x C_z Y - S_x C_z Z + T_y \\ Z' &= -S_y C_z X + (C_y S_x + C_x S_y S_z) Y + (C_x C_y - S_x S_y S_z) Z + T_z \end{aligned} \tag{6}$$

where $C_x = \cos\theta_x$, $C_y = \cos\theta_y$, $C_z = \cos\theta_z$, $S_x = \sin\theta_x$, $S_y = \sin\theta_y$, $S_z = \sin\theta_z$.

We assume that the angles of rotation are sufficiently small such that to a first-order approximation:

$$\cos\theta_x \approx 1, \quad \cos\theta_y \approx 1, \quad \cos\theta_z \approx 1, \quad \sin\theta_x \approx \theta_x, \quad \sin\theta_y \approx \theta_y, \quad \sin\theta_z \approx \theta_z \tag{7}$$

$$\sin\theta_x \sin\theta_y \approx 0, \quad \sin\theta_y \sin\theta_z \approx 0, \quad \sin\theta_z \sin\theta_x \approx 0 \tag{8}$$

For example, if $(-10^o \leq \theta_x, \theta_y, \theta_z \leq 10^o)$ the errors in $\cos\theta \approx 1$ and $\sin\theta \approx \theta$ are at most 2% and 1%, respectively. Under this small angle assumption, Eq. (6) becomes

$$\begin{aligned} X' &= X + \theta_y Z - \theta_z Y + T_x \\ Y' &= Y + \theta_z X - \theta_x Z + T_y \\ Z' &= Z + \theta_x Y - \theta_y X + T_z \end{aligned} \tag{9}$$

If we divide X' and Y' by Z' in Eq. (9), we obtain

$$x' = \frac{X'}{Z'} = \frac{X + \theta_y Z - \theta_z Y + T_x}{Z + \theta_x Y - \theta_y X + T_z} = \frac{x + \theta_y - \theta_z y + \frac{T_x}{Z}}{1 + \theta_x y - \theta_y x + \frac{T_z}{Z}} \tag{10}$$

$$y' = \frac{Y'}{Z'} = \frac{Y + \theta_z X - \theta_x Z + T_y}{Z + \theta_x Y - \theta_y X + T_z} = \frac{y + \theta_z x - \theta_x + \frac{T_y}{Z}}{1 + \theta_x y - \theta_y x + \frac{T_z}{Z}} \tag{11}$$

Cancelling Z from the above two equations, assuming $T_z \neq 0$, dividing both sides with T_z, and letting $L = \frac{T_x}{T_z}, M = \frac{T_y}{T_z}$, we have:

$$\theta_x(y'yL + L - x' - x'yM) + \theta_y(-xy'L + xx'M + M - y') + \theta_z(xx' - xL - yM + yy')$$
$$= Mx' - Mx + xy' - Ly' + yL - x'y \tag{12}$$

Here the known data are the n corresponding beginning points (x, y) and ending points (x', y') and the unknowns are the five motion parameters $(L, M, \theta_x, \theta_y, \theta_z)$. We further constrain the range of L and M by assuming that $-10.0 \leq L, M \leq 10.0$, which corresponds to assuming T_x and T_y are not more than an order of magnitude larger than T_z. Thus we search a discretized and bounded parameter space of $(L, M) \in [-10, 10]^2$ with step size of 0.05 in each direction. For each (L, M), we set up an overdetermined system of equations

$$\underset{(n \times 3)}{\Psi} \quad \underset{(3 \times 1)}{\alpha} \quad = \quad \underset{(n \times 1)}{\beta} \tag{13}$$

where Ψ and β consist of n rows of

$$(y_i'y_iL + L - x_i' - x_i'y_iM, -x_iy_i'L + x_ix_i'M + M - y_i', x_ix_i' - x_iL - y_iM + y_iy_i'), \tag{14}$$

$$(Mx_i' - Mx_i + x_iy_i' - Ly_i' + y_iL - x_i'y_i), \qquad 1 \leq i \leq n \tag{15}$$

and $\alpha = (\theta_x, \theta_y, \theta_z)^T$, where $(\cdot)^T$ denotes vector transpose. For each pair of translation parameters (L, M), we can solve Eq. (12) for a *least-squares solution* of corresponding rotation parameters $(\theta_x, \theta_y, \theta_z)$ as follows:

$$\alpha_{LS} = (\Psi^T\Psi)^{-1}\Psi^T\beta \tag{16}$$

The quintuple $(L, M, \theta_x, \theta_y, \theta_z)$ which minimizes the squared error $(\Psi\alpha - \beta)^T(\Psi\alpha - \beta)$ is the set of recovered motion parameters.

3.2 MAP Estimation

This section explains how our 3-D affine model can include statistical assumptions to include prior information and thus "stabilize" the recovered motion parameters. Assume that the overall effect of displacement estimation errors is to have the error model

$$\beta = \Psi\alpha + \epsilon \tag{17}$$

where $\epsilon = (\epsilon_1, ..., \epsilon_n)^T$ and the random variables ϵ_i are zero-mean independent, and normally distributed with identical variance σ_β^2.

First, if we assume that α is deterministic, its *maximum likelihood (ML)* estimate

$$\alpha_{ML} = \arg\max_{\alpha} P(\beta|\alpha) \tag{18}$$

makes use of whatever information we have about the distribution of the observations (displacement vectors). This ML estimate is equal to [4]:

$$\alpha_{ML} = (\frac{1}{\sigma_\beta^2}\Psi^T\Psi)^{-1}\Psi^T\frac{1}{\sigma_\beta^2}\beta = (\Psi^T\Psi)^{-1}\Psi^T\beta \tag{19}$$

Thus the maximum likelihood estimate is the same as the ordinary least-squares estimate under the above error assumptions.

Further statistical information can be utilized to improve the motion parameter estimates. Assuming now that α is random, by using Bayes' formula

$$P(\alpha|\beta) = \frac{P(\beta|\alpha)P(\alpha)}{P(\beta)} \tag{20}$$

it follows that the *maximum a posteriori (MAP)* estimate for α is

$$\alpha_{MAP} = \arg\max_{\alpha} P(\alpha|\beta) = \arg\max_{\alpha} P(\beta|\alpha)P(\alpha) \tag{21}$$

maximimizes the product of the likelihood and the prior. Since the camera field of view is small in real life, rotation angles are usually small; otherwise, objects will be out of view. We further assume as prior information that $\theta_x, \theta_y, \theta_z$ are independently and normally distributed with zero mean and identical variance σ_α^2. This assumption yields [4]:

$$\alpha_{MAP} = (\frac{1}{\sigma_\beta^2}\Psi^T\Psi + \frac{1}{\sigma_\alpha^2}I)^{-1}\frac{1}{\sigma_\beta^2}\Psi^T\beta = (\Psi^T\Psi + \frac{\sigma_\beta^2}{\sigma_\alpha^2}I)^{-1}\Psi^T\beta \tag{22}$$

The *confidence factor* $\sigma_\beta^2/\sigma_\alpha^2$ reflects the confidence of the prior information relative to that of displacement vectors. The larger $\sigma_\beta^2/\sigma_\alpha^2$, the more confidence about the prior information; on the other hand, if $\sigma_\beta^2/\sigma_\alpha^2$ is small we are more confident in the displacement vectors. Note that if $\sigma_\beta^2/\sigma_\alpha^2 = 0$, then the least-squares estimate, ML estimate, and MAP estimate become the same. The advantage of MAP estimators is that they can include prior information and are flexible because the confidence level can be controlled and hence the solutions can be "stabilized" when the matrix Ψ is ill-conditioned due to noise. The disadvantage is that when the mean values of the parameters assumed by the prior information are different from the actual values (e.g. nonzero rotation angles) and there is no noise in the displacement vectors (e.g. in synthetic simulations), the MAP estimates are shifted toward those mean values (i.e., toward zero rotation angles).

Synthetic simulations [9] show that when no noise is added and $\sigma_\beta^2/\sigma_\alpha^2 = 0$, the recovered motion parameters depend only on displacement vectors. In this case there is almost no error in recovered motion parameters; a small error occurs only because we search a bounded and discrete space for the translational direction $(T_x/T_z, T_y/T_z, 1)$. In our synthetic simulations, the noise added to the beginning points (x, y) and ending points (x', y') was white Gaussian noise. If the synthetic rotation angles are the same as the mean rotation angles assumed by the prior information ($\theta_x = 0^o, \theta_y = 0^o, \theta_z = 0^o$), increasing $\sigma_\beta^2/\sigma_\alpha^2$ always improves the motion parameter estimates. When the synthetic rotation angles are nonzero, as the confidence factor $\sigma_\beta^2/\sigma_\alpha^2$ increases, we are more confident in the prior information, thus the average error of the motion parameter estimates increases. Similar conclusions are achieved when the noise level is low, such as, $SNR \geq 50$dB. Hence, synthetic simulations indicate that more confidence should be on displacement vectors when no or low noise is present.

When the noise in displacement vectors increases, more confidence should be put on the prior information to stabilize the estimates. In [9] it was found via simulations that the optimal confidence factor increases as the noise increases, for cases where the signal-to-noise ratio was ≤ 40 dB. However, the relationship between these two amounts of increase is difficult to quantify and depends on the actual parameter values. Various simulations show that MAP estimation indeed improves motion parameter estimates compared to least-squares estimates or maximum likelihood estimates when there is noise in the displacement vectors.

3.3 Multi-Scale Parameter Searching and Time-Domain Smoothing

In this section we discuss how multi-scale searching of motion parameter space can improve accuracy and how time-domain smoothing of recovered motion parameters can reduce the noise. Since the velocity

equation is valid only instantaneously, each snapshot of scenes shows rigid body motion and is described more accurately by Eq. (5). The first-order approximation estimate of motion parameters $(L, M, \theta_x, \theta_y, \theta_z)$ is computed as described in Sections 3.1 and 3.2 and is used as the *initial estimate*. More accurate motion parameter estimates can be achieved by further refining this initial estimate through multi-scale searching (i.e., locally searching) the bounded and discretized motion parameter space around the initial estimate in a finer scale. This is explained next.

We return to the true motion equations of rigid body, define the error term, and locally search the bounded and discretized motion parameter space around the initial estimate in a finer scale. Using Eq. (6) and dividing X' and Y' by Z' yields

$$x' = \frac{X'}{Z'} = \frac{C_z C_y x + (S_x S_y - C_x C_y S_z) y + (C_x S_y + S_x C_y S_z) + \frac{T_x}{Z}}{-S_y C_z x + (C_y S_x + C_x S_y S_z) y + (C_x C_y - S_x S_y S_z) + \frac{T_z}{Z}} \tag{23}$$

$$y' = \frac{Y'}{Z'} = \frac{S_z x + C_x C_z y - S_x C_z + \frac{T_y}{Z}}{-S_y C_z x + (C_y S_x + C_x S_y S_z) y + (C_x C_y - S_x S_y S_z) + \frac{T_z}{Z}} \tag{24}$$

By cancelling Z from the above two equations, assuming $T_z \neq 0$, dividing both sides with T_z, and letting $L = \frac{T_x}{T_z}, M = \frac{T_y}{T_z}$, we define the error, for each corresponding pair,

$$\begin{aligned} Error(L, M, \theta_x, \theta_y, \theta_z) &= (C_y C_z + L S_y C_z) x y' + (S_x S_y - C_x C_y S_z - L C_x S_y S_z - L S_x C_y) y y' \\ &+ (C_x S_y + S_x C_y S_z - L C_x C_y + L S_x S_y S_z) y' - (M S_y C_z + S_z) x x' \\ &+ (M C_x S_y S_z + M S_x C_y - C_x C_z) y x' + (M C_x C_y - M S_x S_y S_z + S_x C_z) x' \\ &+ (L S_z - M C_y C_z) x + (L C_x C_z - M S_x S_y + M C_x C_y S_z) y - (M C_x S_y + M S_x C_y S_z + L S_x C_z) \end{aligned} \tag{25}$$

Ideally (in the noise-free case) $Error = 0$. But in practical experiments $Error \neq 0$, and we find the optimal $(L, M, \theta_x, \theta_y, \theta_z)$ that minimize $\sum (Error)^2$ over all corresponding pairs. The multi-scale searching is done by locally searching around the initial estimates in a finer scale. We search the discretized and bounded parameter space of $[\theta_x - 1^o, \theta_x + 1^o], [\theta_y - 1^o, \theta_y + 1^o], [\theta_z - 1^o, \theta_z + 1^o]$ with step size of 0.1^o and $[L - 0.05, L + 0.05], [M - 0.05, M + 0.05]$ with step size of 0.005. The quintuple $(L, M, \theta_x, \theta_y, \theta_z)$ which yields the minimum sum of squares of $Error$ is the set of recovered motion parameters. The multi-scale searching improves the accuracy of motion parameter estimates and avoids high computational cost since searching the complete motion parameter space with such a fine scale would be computationally expensive.

After multi-scale searching to compute more accurate motion parameters, we can substitute them back to Eq. (23) or (24) to compute $\frac{Z}{T_z}$, i.e. the depth of the object surface up to a scaling factor by:

$$\frac{Z}{T_z} = \frac{x' - L}{S_y C_z x x' - (C_x S_y S_z + S_x C_y) x' y - (C_x C_y - S_x S_y S_z) x' + C_y C_z x + (S_x S_y - C_x C_y S_z) y + \xi} \tag{26}$$

$$\frac{Z}{T_z} = \frac{y' - M}{S_y C_z x y' - (C_x S_y S_z + S_x C_y) y' y - (C_x C_y - S_x S_y S_z) y' + S_z x + C_x C_z y - S_x C_z} \tag{27}$$

where $\xi = (C_x S_y + S_x C_y S_z)$. The choice of which above equation or combination of them to use depends on the numerical considerations and motion. For example, when T_y is dominant (the motion is mainly horizontal translation), Eq. (27) is better than Eq. (26) because the situation is similar to stereo vision to recover object shape, where y' and y carry depth information but x' and x are almost constant. Similarly, when T_x is dominant (the motion is mainly vertical translation), Eq. (26) is better than Eq. (27).

Although the least-squares algorithm with MAP estimation and multi-scale searching has been found to be robust in many cases, the motion and shape recovery of real world images is sometimes sensitive to noise and the estimated motion parameters have errors due to the ambiguity that very differnt motion can induce similar displacement fields. We treat the errors in the recovered motion parameters as noise and

additional improvement can be achieved by smoothing the motion parameters in the time domain when the motion remains constant or varies smoothly between image frames. We choose median filtering because of its relative robustness compared to a linear averager. Thus the smoothed motion parameter θ_x at time j is the scalar median of the $2m+1$ estimates of θ_x centered at time j:

$$\theta_x(j) = med\{\theta_x(i) : i = j-m, j-m+1, ..., j, ..., j+m\} \tag{28}$$

We have found this time-domain median smoothing to perform well in reducing errors of estimated motion parameters, as shown in experiments presented in Section 3.4.

3.4 Experiments and Discussion

It is well known that different motions can induce similar displacement vector fields; thus motion and shape recovery algorithms rely on the consistency of d_x and d_y to clarify the ambiguity. To smooth[2] the estimated displacement field and eliminate some errors, we introduce a *nonlinear outlier removal filter* which leaves the displacement vector unchanged if it "agrees" with more than $\frac{1}{3}$ of its neighbors and removes the displacement vector if it "agrees" with fewer than $\frac{1}{3}$ of its neighboring displacement vectors. We say that a displacement vector $d_i = \{d_{x,i}, d_{y,i}\}$ "agrees" with its neighbor $d_j = \{d_{x,j}, d_{y,j}\}$ if and only if

$$|d_{x,i} - d_{x,j}| < 0.1 \cdot max(|d_{x,i}|, |d_{y,i}|) \text{ and } |d_{y,i} - d_{y,j}| < 0.1 \cdot max(|d_{x,i}|, |d_{y,i}|). \tag{29}$$

The two sides of an object with large depth difference can have very different displacement vector patterns; we choose "$\frac{1}{3}$" because if "$\frac{1}{3}$" of the neighbors are consistent then the displacement vectors of both sides stay unchanged. The proportional parameter, "0.1", constrains how stringently two displacement vectors must "agree". Both parameters can be changed depending on image sequence and applications. The nonlinear outlier removal filter has been demonstrated experimentally to be suitable for motion and shape recovery on various real world image sequences.

Figure 3 shows three frames from 6-frame toy truck image sequence with no rotation ($\theta_x = \theta_y = \theta_z = 0^o$) and an equal amount of translation ($T_x = T_y = T_z = -5$mm $\Rightarrow T_x/T_z = T_y/T_z = 1$) between each image frame. Here, camera yaw is θ_x; pitch is θ_y; roll is θ_z, all in degrees. Translation T_x points upward; T_y points rightward; T_z points toward the objects. The lower left truck is the closest (170mm away), the lower right truck is at middle (220mm away), and the upper tractor truck is the farthest (360mm away). We use the 2-D displacement vectors estimated by the 2-D affine model because the estimates are dense and accurate as shown in Figure 3(d). As shown in Figure 3(e), the nonlinear outlier removal algorithm performs well to remove the mismatches around occlusion boundaries. We use $\sigma_\beta^2/\sigma_\alpha^2 = 0.01$ in the MAP estimation because the displacement vector field has low noise after nonlinear outlier removal. Table 2 shows the recovered motion parameters of the image sequence. The rotation angles are almost zero (compared to 40 degrees of FOV) and translation direction $(T_x/T_z, T_y/T_z, 1)$ has at most 20% error. Because the motion is constant, we can apply the time-domain median smoothing on motion parameters and have $\theta_x = 0.349^o, \theta_y = -0.305^o, \theta_z = 0.009^o, T_x/T_z = 0.950, T_y/T_z = 0.950$, and it shows an improvement over most individual estimates. We use the above motion parameters to compute the object shape in the form of depth map. The average error for the depth map in Figure 3(f) was 15%. There is one depth estimate at each center of 19×19 block and these centers are 7 pixels apart horizontally and vertically. We repeat the depth estimate for the 7×7 pixels around the block center. The two black stripes on the right of the range image are not errors but indicate there is no depth information because the mismatches caused by occlusion boundaries are removed by nonlinear outlier removal.

[2] An alternative smoothing of the displacement vectors, we have also used component-wise median filtering. However, we found that the small variations introduced to d_x and d_y by vector median smoothing can affect the accuracy of the 3-D motion and shape recovery algorithm.

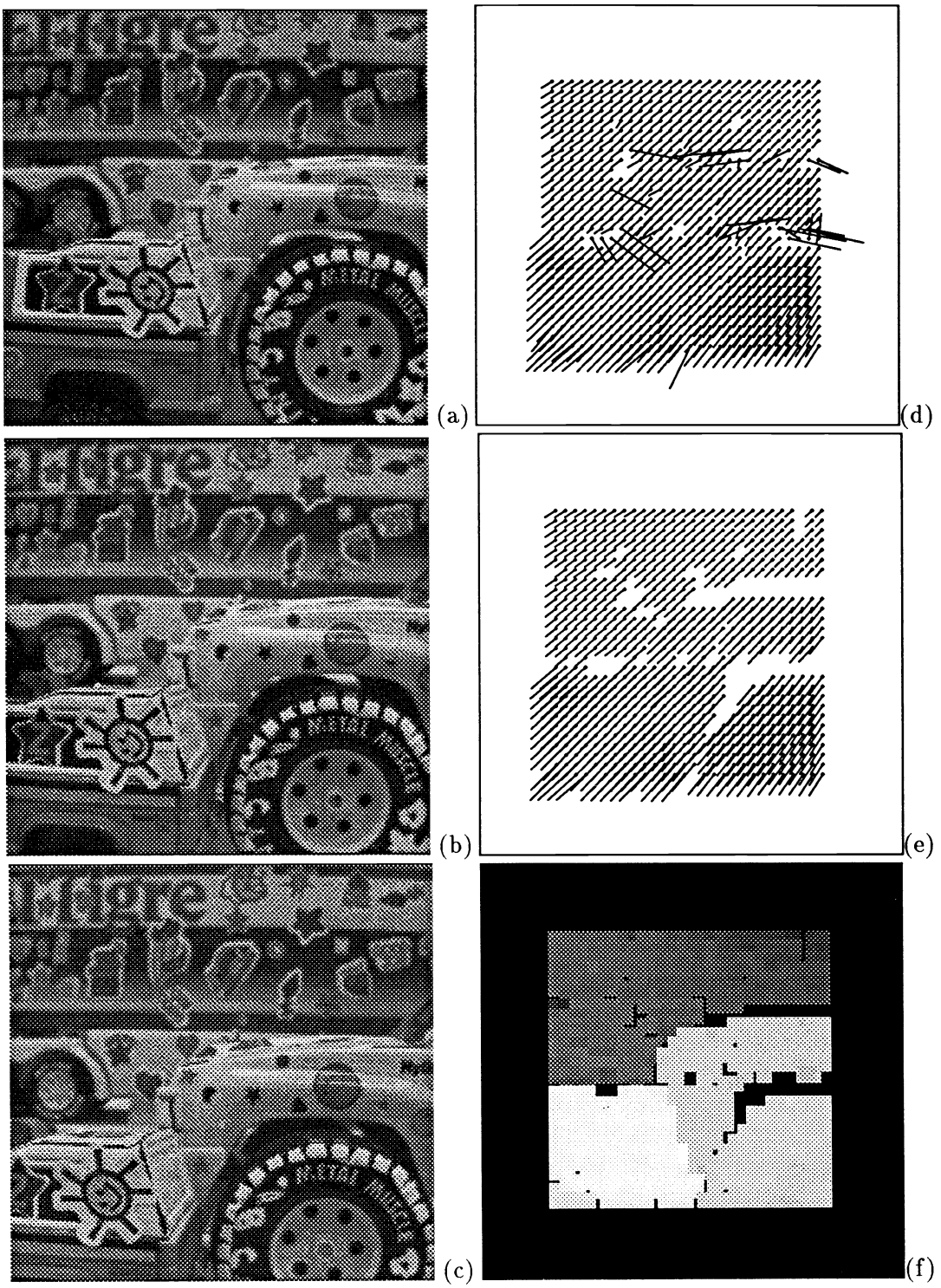

Figure 3 : Toy truck image sequence, $\theta_x = \theta_y = \theta_z = 0^o, T_x = T_y = T_z = -5$mm. (a) Frame 3 ($386 \times 386$ pixels, 8 bit/pixel) (b) Frame 4 (c) Frame 5 of the image sequence. (d) Result of 2-D affine block matching of (a) and (b). (e) Result of nonlinear outlier removal on (d). (f) Range image of recovered object depth of (a). (The brighter the closer; the darker the farther away).

Table 2 : *Recovered motion parameters of the toy truck image sequence. The measured values are* $\theta_x = \theta_y = \theta_z = 0^o$ *and* $L = M = 1$.

frames	θ_x	θ_y	θ_z	$L = T_x/T_z$	$M = T_y/T_z$
1,2	0.037	-0.008	0.007	1.200	1.200
2,3	0.180	-0.133	0.009	1.100	1.100
3,4	0.349	-0.305	0.013	0.950	0.950
4,5	0.469	-0.406	0.007	0.900	0.900
5,6	0.453	-0.396	0.011	0.900	0.900

Table 3 : *Measured and recovered motion parameters of the mountain image sequence. (The field of view is approximately* 50^o*.)*

frames	data	θ_x	θ_y	θ_z	$L = T_x/T_z$	$M = T_y/T_z$
12,13	measured	2.181	0.192	-2.137	-0.258	0.000
	recovered	2.513	0.094	-0.819	-0.320	0.000
13,14	measured	3.417	4.603	-5.477	-0.254	0.000
	recovered	4.927	4.978	-3.492	-0.255	0.070
14,15	measured	2.357	-2.620	-1.549	-0.170	0.000
	recovered	2.223	-3.024	-0.947	-0.235	0.045

Figure 4 shows three frames from a 21-frame mountain image sequence. As shown in this figure, the non-linear outlier removal algorithm performs well to remove mismatches around occlusion (the boundary between mountain top and cloud). We use $\sigma_\beta^2/\sigma_\alpha^2 = 0.01$ in MAP estimation because the displacement vector field has low noise after nonlinear outlier removal. Table 3 shows the typical measured and recovered motion parameters. The rotation angles have on average 15% error, L has 20% average error, and M is almost zero. The following are several possible causes for the large estimation errors. This is a "move and shoot" image sequence; the vehicle does not stop to stabilize and the road surface is unpaved. The motions between image frames are quite abrupt and time-domain smoothing of motion parameters is not suitable. The translation is also mainly along optical axis, so the depth estimates are more sensitive to noise. We suspect the cloud moves relative to the mountain thus this relative motion violates rigid body constraint. The relative motion might cause the cloud to appear closer than the mountain as shown in the range image.

4 Conclusion

We presented a visual motion analysis system which includes a 2-D affine model to determine 2-D motion displacement fields and a 3-D affine model to recover the 3-D motion parameters and surface structure under perspective projection. The parameters of both affine models are found using least-squares algorithms and a limited searching in a bounded parameter space. In the 3-D affine motion and shape recovery algorithm, a simple form of MAP estimation was added to stabilize the recovered motion parameters in the presence of noise in the displacement vector field. Multi-scale searching improves accuracy without high computational cost. Time-domain smoothing improves motion parameter estimates when the motion remains constant or

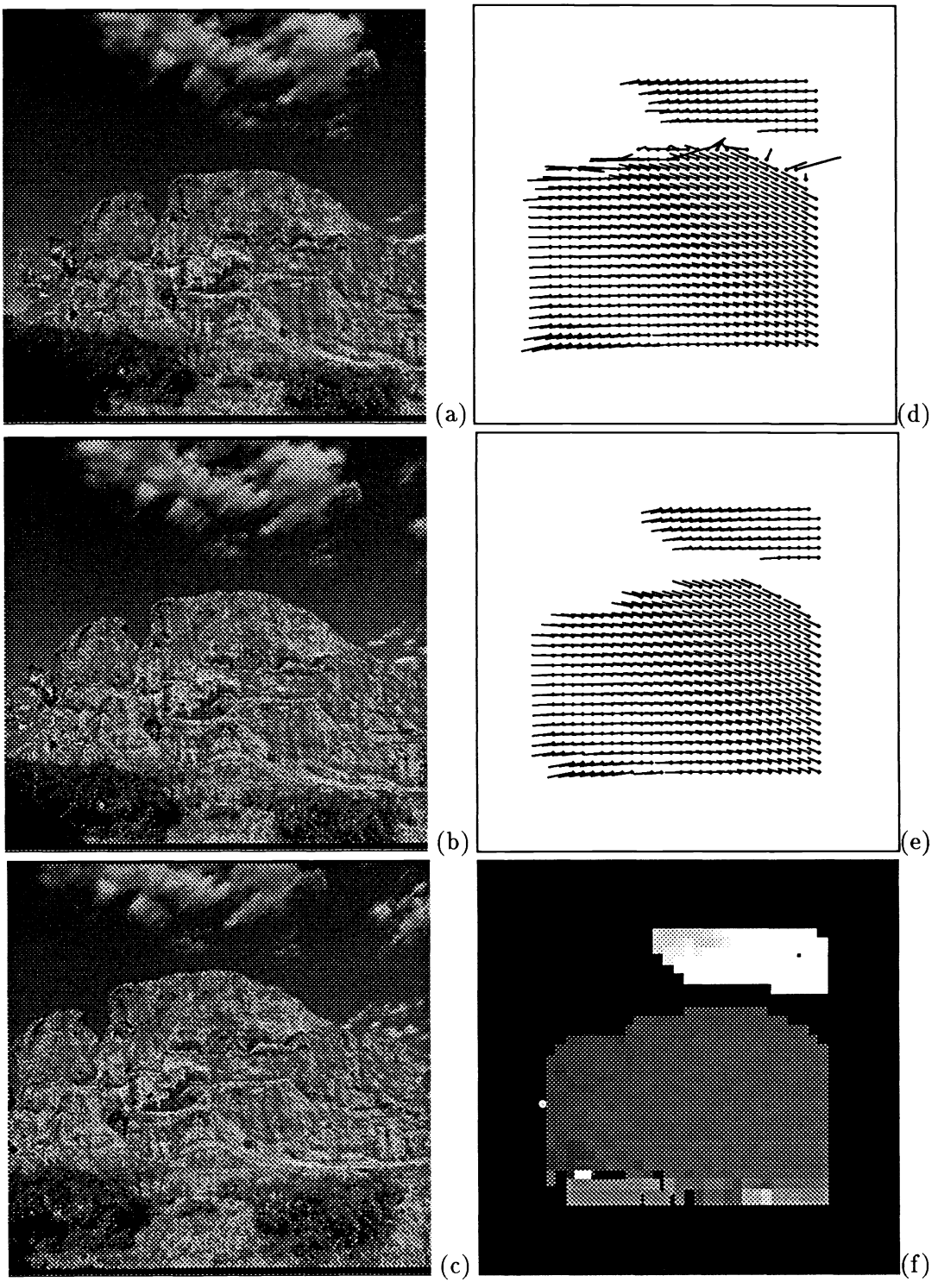

Figure 4 : A mountain image sequence from the University of Massachusetts at Amherst motion data set [6]. (a) Frame 12 (386 × 386 pixels, 8 bit/pixel) (b) Frame 13 (c) Frame 14 of the image sequence. (d) Result of 2-D affine block matching of (a) and (b). (e) Result of nonlinear outlier removal on (d). (f) Range image of recovered object depth of (a).

varies slowly. Many synthetic simulations as well as experiments on real world image sequences indicate that the proposed affine models and related algorithms are effective and can robustly recover motion parameters and object shape with relatively small errors.

Acknowledgements

This research work was supported by a National Science Foundation Presidential Young Investigator Award under NSF Grant MIP-86-58150 with matching funds from DEC and Xerox, by TASC, and in part by the ARO Grant DAALO3-86-K-0171 to the Brown-Harvard-MIT Center for Intelligent Control Systems.

References

[1] Y.S. Abu-Mostafa and D. Psaltis, "Image Normalization by Complex Moments," *IEEE Trans. on Patt. Anal. Mach. Intell.*, vol. 7, pp. 46-55, Jan. 1985.

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

[3] S.T. Barnard and W.B. Thompson, "Disparity Analysis in Images," *IEEE Trans. on Patt. Anal. Mach. Intell.,* vol. 2, pp. 333-340, 1980.

[4] J.V. Beck and K.J. Arnold, *Parameter Estimation in Engineering and Science,* J. Wiley & Sons, New York, 1977.

[5] R. Brockett, "Gramians, Generalized Inverses, and the Least-Squares Approximation of Optical Flow", *J. Visual Commun. Image Repres.*, 1, pp.3-11, Sep. 1990.

[6] R. Dutta, R. Manmatha, L.R. Williams, and E.M. Riseman, "A Data Set for Quantitative Motion Analysis," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* pp. 159-164, San Diego, June 1989.

[7] C.S. Fuh and P. Maragos, "Motion Displacement Estimation Using an Affine Model for Image Matching," *Optical Engineering,* vol. 30, pp. 881-887, July 1991.

[8] C.S. Fuh, P. Maragos, and L. Vincent, "Region-Based Approaches to Visual Motion Correspondence," submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence.* Also *Technical Report 91-18, Harvard Robotics Lab.,* Nov. 1991.

[9] C.S. Fuh, "Visual Motion Analysis: Estimating and Interpreting Displacement Fields" Ph.D. thesis, Division of Applied Sciences, Harvard University, 1992.

[10] M. Gilge, "Motion estimation by scene adaptive block matching (SABM) and illumination correction," *Image Processing Algorithms and Techniques*, Proc. SPIE vol. 1244, pp.355-366, 1990.

[11] A.W. Gruen and E.P. Baltsavias, "Adaptive Least Squares Correlation with Geometrical Constraints," *Computer Vision for Robots,* Proc. SPIE, vol. 595, pp. 72-82, 1985.

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

[13] T.S. Huang and R.Y. Tsai, "Image Sequence Analysis: Motion Estimation," in *Image Sequence Analysis,* T.S. Huang, Ed., Springer-Verlag, 1981.

[14] J.R. Jain and A.K. Jain, "Displacement Measurement and Its Application in Interframe Coding," *IEEE Trans. Commun.,* COM-29, pp. 1799-1808, Dec. 1981.

[15] D.S. Kalivas, A.A. Sawchuk, and R. Chellappa, "Segmentation and 2-D Motion Estimation of Noisy Image Sequences," in *Proc. IEEE Int'l. Conf. Acoust., Speech, Signal Process.,* pp. 1076-1079, New York, Apr. 1988.

[16] M. Lee, "Recovering the Affine Transformation of Images by Using Moments and Invariant Axes," *Image Understanding and Machine Vision,* Technical Digest Series, vol. 14, pp. 2-5, Washington, D.C.: Opt. Soc. Amer., 1989.

[17] P. Maragos, "Pattern Spectrum and Multiscale Shape Representation," *IEEE Trans. on Patt. Anal. Mach. Intell.,* vol. 11, pp. 701-716, July 1989.

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

[19] D. Marr, *Vision*, W.H. Freeman & Co., San Francisco, 1982.

[20] G. Matheron, *Random Sets and Integral Geometry*, Acad. Press, New York, 1975.

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

[22] A.N. Netravali and J.D. Robbins, "Motion Compensated Television Coding– Part I," *Bell Syst. Tech. J.,* vol. 58, pp. 631-670, March 1979.

[23] J. Serra, *Image Analysis and Mathematical Morphology,* Academic Press, London, 1982.

[24] R.Y. Tsai and T.S. Huang, "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces," *IEEE Trans. Pattern Anal. Mach. Intellig.*, vol. 6, pp. 13-27, Jan. 1984.

[25] K.H. Tzou, T.R. Hsing, and N.A. Daly, "Block-Recursive Matching Algorithm(BRMA) for Displacement Estimation of Video Images," *Proc. IEEE Intl. Conf. Acoust., Speech, Signal Process.,* pp. 359-362, Tampa, March 1985.

[26] J. Weng, T.S. Huang, and N. Ahuja, "Motion and Structure from Two Perspective Views: Algorithms, Error Analysis, and Error Estimation," *IEEE Trans. on Patt. Anal. Mach. Intell.,* vol. 11, pp. 451-476, May 1989.