JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 12, 1-23 (1996)

DOI: 10.6688/JISE.1996.12.1.1

Least-Squares Algorithms for Motion and Shape Recovery Under Perspective Projection

CHIOU-SHANN FUH[†] AND PETROS MARAGOS[‡]

[†] *Department of Computer Science and Information Engineering*
National Taiwan University
Taipei, Taiwan, R.O.C.
[‡] *School of Electrical Engineering*
Georgia Institute of Technology
Atlanta, GA 30332, USA

This paper presents an algorithm 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 rigid body motion 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 maximum a posteriori estimation. A multi-scale search in the parameter space is also used to improve accuracy without high computational cost. Experiments on applying this algorithm to various real world image sequences demonstrate that it can estimate dense displacement fields and recover motion parameters and object shape with relatively small errors.

Keywords: computer vision, motion analysis, correspondence.

1. INTRODUCTION

1.1 Background

Visual motion analysis can provide rich information about the 3-D motion and surface structure of moving objects with many applications on vision-guided robots, video data compression, and remote sensing. There are two major problems in this area: the first is determination of 2-D motion displacement fields from time sequences of intensity images. The second problem is to recover the motion parameters, which include 3-D translations and rotations, and the surface structure in terms of object depth relative to the camera or retina by using the

Received August 18, 1994; revised May 19, 1995.
Communicated by Zen Chen.

estimated displacement field. There has been much previous and important work on visual motion analysis as summarized in [1, 15, 21, 23].

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. [14]. Tomasi and Kanade [28] proposed a point feature detection and tracking algorithm which uses Taylor expansion and the linearization method and assumes that the displacement is much smaller than the window size. A broad class of gradient methods is all the pixel-recursive algorithms, popular among video coding researchers [12, 24, 25]. 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 [1, 2, 6, 29]. 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 [16, 24]. 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 [5, 9, 10, 17, 30].

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. Tsai and Huang [29, 32] used seven correspondence point pairs to determine 3-D motion parameters of curved surfaces from 2-D perspective views. Heeger and Jepson [13] proposed a method for computing 3-D motion and depth, but they used the image velocity equations of rigid body and velocity equations which are only the first-order approximation of rigid body motion captured in any two image frames. Longuet-Higgins and Prazdny [19] showed that an observer can, in principle, determine the structure of a rigid scene and his direction of motion relative to it from the instantaneous retinal velocity field. Waxman and Ullman [31] introduced a new representation of a local image flow in terms of the image velocities, strain rates, spin, and image gradients of the strain rate and spin, evaluated along the line of sight to a moving surface. Harris [11] explored structure-from-motion algorithms based on matched point-like features under orthographic projection for use in analyzing image motion from small rigid moving objects. Tomasi and Kanade [28] achieved very good results when they assumed orthographic projections and used a stream of real world images with many points on each image. They decomposed the coordinate matrix directly into motion parameter and object surface structure matrices without resorting to depth as an intermediate step. Quan and Mohr [26] recovered the object surface structure from motion for linear features through referenced points.

1.2 Organization

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 under perspective projection. Our strength is that 3-D motion and shape recovery uses real 2-D displacement fields from real world images while most 2-D algorithms generate displacement fields which are never really tested in subsequent 3-D motion and shape recovery and most 3-D algorithms simply input synthetic displacement field. Orthographic projection has the property that the projection of the centroid of the object points is the centroid of the projections of the object points and, thus, can be used to separate translation from rotation. However, orthographic projection is not a good model in the real world because foreshortening is an important clue to motion recovery. Perspective projection is more realistic for real world applications but is more difficult to analyze.

In Section 2.1, we will review from [5] 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 the image intensity range. The model parameters are found by using a least-squares algorithm. (In a related work [10], 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 [5], we experimentally demonstrated that our affine block matching algorithm performs better in estimating displacements than do standard block matching and gradient methods, especially for long-range motion with possible changes in scene illumination. In Section 2.2, we will 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 will present an algorithm that uses a least-squares method to recover the 3-D rigid body motion parameters and surface structure based on two perspective views and the given 2-D displacement data estimated by 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 maximum a posteriori (MAP) estimation to include prior information and to stabilize the parameters. Although the rigid body motion equation is the same as that used in [32], 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 noise when the motion remains constant or varies smoothly.

Our contribution includes a novel least-squares algorithm to estimate the 2-D displacement field, a new approach to guide selection of the block size, and an original least-squares algorithm with MAP estimation and multi-scale searching for 3-D motion and shape recovery. The proposed algorithm is applied to time sequences of real world images and is shown to give displacement vectors, motion parameters, and surface structure with 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 [5]. 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 times 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

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

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. Although we expect a certain variation of the affinities over the area of the image of an object such as a sphere, we hope that the 2-D affine model, where the spatial transformation may or may not be coupled with the intensity transformation, will be a reasonable approximation in that situation.

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 [5] 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^* and c^* as functional of M and d. Replacing the optimal r^*, c^* into E yields a modified error functional $E^*(M, d)$. Now, by discretizing the 6-D parameter space M, d and exhaustively searching a bounded region, we find the optimal M^*, d^* that minimize $E^*(M, d)$. 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 to be a square of $B \times B$ pixels. In our implementation, we assume $s_x = s_y$ and $\theta_x = \theta_y$, so the computational complexity is $O(L^2 \times B^2 \times S \times \Theta)$, where S and Θ are the numbers of search samples in scaling and rotation angles, respectively. After having found the optimal M^* and d^*, we can obtain the optimal r^* and c^* [5].

Figs. 1(a), (b) show an original "Poster" image and a synthetically trans-

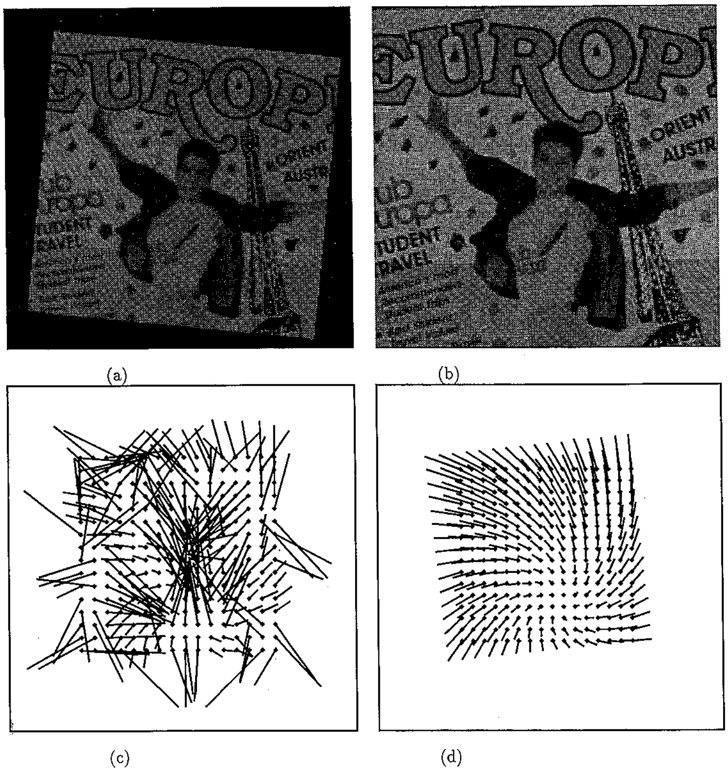


Fig. 1. Simulation result of the affine block model. (a) An affine transformed version of the image in (b) with translation $d = (5,5)$, rotation $\theta = 6°$, scaling $s = 1.2$, intensity ratio $r = 0.7$, and intensity bias $c = 20$. (b) The original "Poster" image. (242x242 pixels, 8-bit/pixel). (c) Displacement vectors between the images in (a) and (b) obtained from standard block matching. (d) Displacement vectors from the affine matching algorithm ($L = 40$ pixels).

formed image according to the affine model with a global translation of $d = (5,5)$ pixels, rotation by $\theta = 6°$, 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. Fig. 1(c) shows the displacement field estimated via the standard block matching where the error $\Sigma_{p \in R} | I(p, t_1) - I(p + d, t_2) |^2$ is minimized. The standard block matching performs poorly because it assumes only translation and has difficulty dealing with scaling or rotation. The standard block matching also assumes constant intensity and results in poor matching under intensity scaling and intensity bias. Fig. 1(d) 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 $\theta_x = \theta_y \in [-6°, 6°]$; also we had set $B = 19$ and $L = 40$. The experiment clearly shows the superior performance of the affine model for image matching to the standard block matching; however, the superior performance comes at higher computational cost.

Our affine matching algorithm not only performs well on rigid objects undergoing short- or long-range motion and/or changes in scene lighting, but also has satisfactory performance on nonrigid objects such as moving clouds or hurricane where the interframe changes of object shapes could be very large. Figs. 2(a) and (b) show two time frames from a satellite infrared hurricane image sequence where the intensity represents the altitude of the cloud top. Fig. 2(c) shows the centers of the matching blocks and the scale of the block size. In this experiment, there are no displacement estimates for blocks whose standard deviations of intensity are less than 5 because in such cases there is insufficient texture information in the analysis region to perform a successful matching. There are also no estimates for blocks which correspond to multiple blocks in the next image frame with the same minimum matching error. Fig. 2(d) shows the motion displacement field d that results by applying the above affine matching algorithm and smoothing the raw estimates by using a spatio-temporal vector median filter [7]. The motion is quite rapid and inhomogeneous across the image.

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; 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. Fig. 3 shows the relationship between the block size and the performance of the 2-D affine matching. Since the whole image in Fig. 1(a) is an affine transformation of the image in Fig. 1(b), Figs. 3(a), (b), and (c) illustratively show that 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 Fig. 3(d) numerically 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

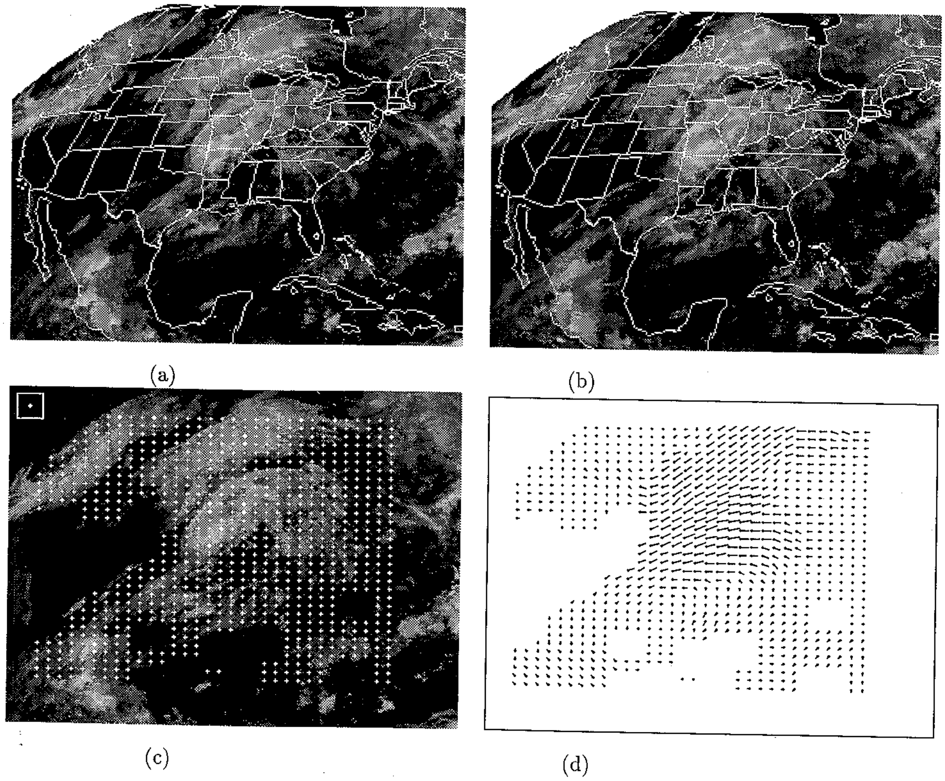


Fig. 2. Affine block model on a hurricane image sequence. (a) First frame of a satellite infrared cloud image sequence from a hurricane (240x320 pixels, 4-bit/pixel). (b) Second frame of the hurricane sequence (30 minutes between frames). (c) The scale of size and centers of the blocks used for affine matching. The block size is 19x19 as shown on upper left corner with the block center. The horizontal and vertical spacings are 8 and 6 pixels, respectively. (d) Displacement vectors (magnified 1.5 times) from the affine matching algorithm, smoothed by a vector median filter. (B = 19, L = 15 pixels.)

based on the shapes and sizes of the binary objects in X. The *morphological shape-size histogram* [20, 27], based on multiscale openings/closings and granulometries [22] and also called the 'pattern spectrum' in [20], offers a good description of the shape and size information of the objects in the binary image X and is defined as follows:

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

where $A[\cdot]$ denotes the area, and $X \circ nS$ and $X \bullet nS$ denote the opening and closing of X by a structuring element S of size n. The opening of image X by the structuring element K is denoted by $X \circ K = (\mathrm{X} \ominus K) \oplus K$. The closing of image X by the structuring element K is denoted by $X \bullet K = (X \oplus K) \ominus K$. In turn, the dilation of the binary image X by the structuring element K is defined by

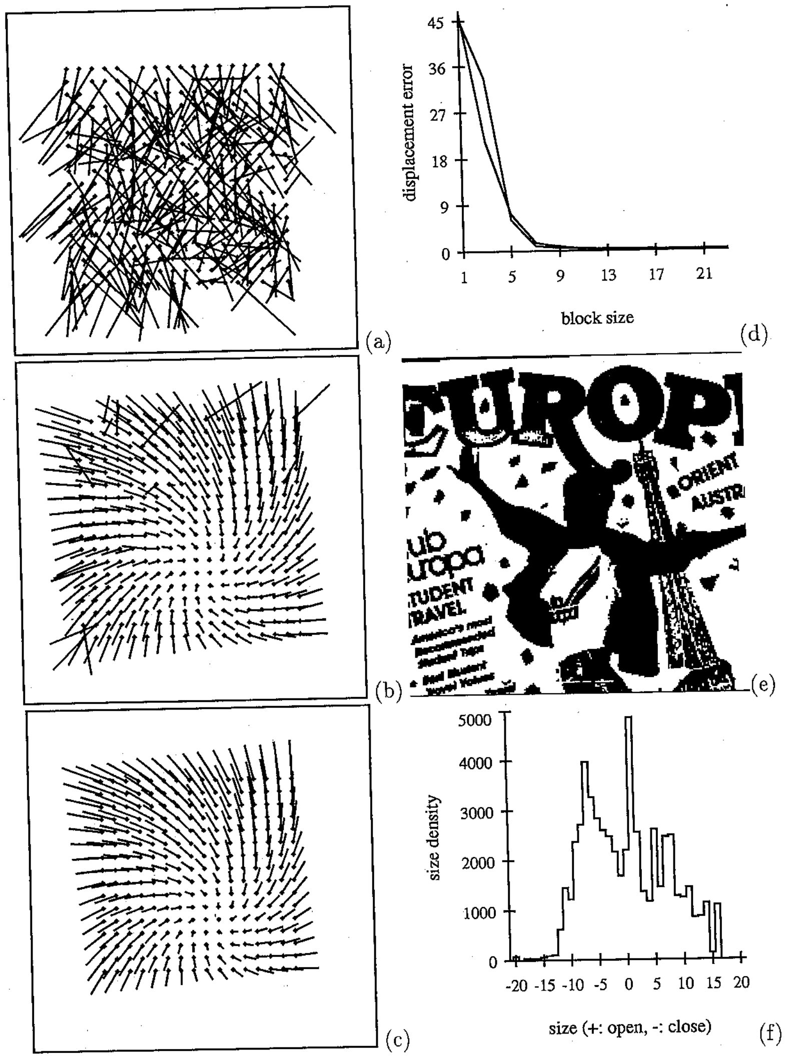


Fig 3. Selection of block size B. Result of matching Figs. 1(a) and 1(b) where the block size is: (a) 3×3, (b) 7×7, (c) 23×23. (d) Erros of d_x an d_y (in pixels) with respect to varying block size. (e) Binarized image of (b). (f) Size histogram of (e) using a 3×3 square structuring element.

$$X \oplus K = \{c \in E^N \,|\, c = x + k \text{ for some } x \in X \text{ and } k \in K\}, \tag{4}$$

where E^N is the Euclidean N-space. The erosion of the image X by the structuring element K is defined by

$$X \ominus K = \{c \in E^N \,|\, c + k \in X \text{ for every } k \in K\}, \tag{5}$$

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 used square analysis regions for image matching, so we fixed S to be a 3 × 3-pixel square.

We convert a gray-tone image frame into a binary image X by thresholding at the median of the intensity values so as to obtain approximately equal numbers of dark and bright pixels. Note that opening and closing are dual operations on bright and dark pixels; hence, the size the 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 Fig. 3(e), and its size histogram is shown in Fig. 3(f). Note that we did not use any edge operator to convert a gray-tone image into a binary image because good edge detection requires pre-smoothing of the image, and the size of the smoothing kernel affects the size histogram.

By using the size histogram and a heuristic rule for the selection of block size B, we can avoid expensive multi-scale analysis in choosing 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; through experiments on various image sequences of synthetic translation and rotation, we found a reasonable minimum to be about 11. After some experimentation on various images, we found strong correlation between n_{max} and the 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 in Fig. 3, 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. Our experiments on various images show strong correlation between n_{max} and B_{opt} but not a dead sure relationship; thus, n_{max} is a good heuristic indicator of B_{opt}. Experimental details about n_{max} and B_{opt} are abundant

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

B × 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

and would make this paper unduly long; thus, they will be presented in another paper now in preparation.

Overall, we have applied the affine block matching algorithm to various indoor and outdoor image sequences, and the experimental results show that 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 rigid body motion equation whose parameters are found using a least-squares algorithm, maximum a posteriori parameter estimation, and multi-scale parameter searching.

3.1 Rigid Body Motion 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 Fig. 4. 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 point on the 2-D image plane before and after the motion; thus we have

$$x=\frac{X}{Z}\ x'=\frac{X'}{Z'}\ y=\frac{Y}{Z}\ y'=\frac{Y'}{Z'}. \tag{6}$$

Rigid motion includes rotation by angles θ_x, θ_z, θ_y around their respective axes X, Z, Y in the given order (other orders can be solved similarly), 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 rigid body motion equation:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} C_y & 0 & S_y \\ 0 & 1 & 0 \\ -S_y & 0 & C_y \end{bmatrix} \begin{bmatrix} C_z & -S_z & 0 \\ S_z & C_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_x & -S_x \\ 0 & S_x & C_x \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{7}$$

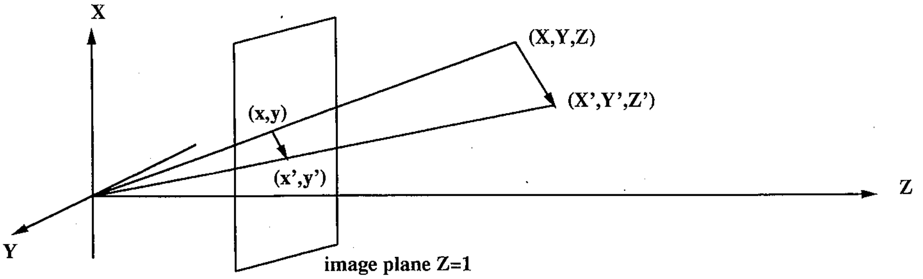


Fig. 4. Camera setup and the perspective projection.

$$
\begin{aligned}
X' &= C_zC_yX + (S_xS_y - C_xC_yS_z)Y + (C_xS_y + S_xC_yS_z)Z + T_x \\
Y' &= S_zX + C_xC_zY - S_xC_zZ + T_y \\
Z' &= -S_yC_zX + (C_yS_x + C_xS_yS_z)Y + (C_xC_y - S_xS_yS_z)Z + T_z,
\end{aligned}
\tag{8}
$$

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 as to arrive at a first-order approximation:

$$
\cos\theta_x \approx 1,\ \cos\theta_y \approx 1,\ \cos\theta_z \approx 1,\ \sin\theta_x \approx \theta_x,\ \sin\theta_y \approx \theta_y,\ \sin\theta_z \approx \theta_z \tag{9}
$$

$$
\sin\theta_x \sin\theta_y \approx 0,\ \sin\theta_y \sin\theta_z \approx 0,\ \sin\theta_z \sin\theta_x \approx 0. \tag{10}
$$

For example, if ($-10° \leq \theta_x, \theta_y, \theta_z \leq 10°$), 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. (8) becomes the velocity equation

$$
\begin{aligned}
X' &= X + \theta_yZ - \theta_zY + T_x \\
Y' &= Y + \theta_zX - \theta_xZ + T_y \\
Z' &= Z + \theta_xY - \theta_yX + T_z.
\end{aligned}
\tag{11}
$$

Note that Heeger and Jepson [13] computed 3-D motion and depth with this velocity equation, which is only a first-order approximation of the rigid body motion equation.

If we divide X' and Y' by Z' in Eq. (11), we obtain

$$
x' = \frac{X'}{Z'} = \frac{X + \theta_yZ - \theta_zY + T_x}{Z + \theta_xY - \theta_yX + T_z} = \frac{x + \theta_y - \theta_zy + \frac{T_x}{Z}}{1 + \theta_xy - \theta_yx + \frac{T_z}{Z}} \tag{12}
$$

$$
y' = \frac{Y'}{Z'} = \frac{Y + \theta_zX - \theta_xZ + T_y}{Z + \theta_xY - \theta_yX + T_z} = \frac{y + \theta_zx - \theta_x + \frac{T_y}{Z}}{1 + \theta_xy - \theta_yx + \frac{T_z}{Z}}. \tag{13}
$$

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

$$
\begin{aligned}
&\theta_x(y'yL + L - x' - x'yM) + \theta_y(-xy'L + xx'M + M - y') \\
&\quad + \theta_z(xx' - xL - yM + yy') \\
&= Mx' - Mx + xy' - Ly' + yL - x'y.
\end{aligned}
\tag{14}
$$

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 that 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 a step size of 0.05 in each direction. For each (L, M), we set up an overdetermined system of equations:

$$\underset{(n\times 3)}{\Psi} \quad \underset{(3\times 1)}{\alpha} = \underset{(n\times 1),}{\beta} \tag{15}$$

where Ψ and β consist of n rows of

$$(y_i' y_i L + L - x_i' - x_i' y_i M, -x_i y_i' L + x_i x_i' M + M - y_i', x_i x_i' - x_i L - y_i M + y_i y_i'), \tag{16}$$

$$(Mx_i' - Mx_i + x_i y_i' - Ly_i' + y_i L - x_i' y_i), \quad 1 \leq i \leq n \tag{17}$$

and $\alpha = (\theta_x, \theta_y, \theta_z)^T$, where $(\cdot)^T$ denotes the vector transpose. For each pair of translation parameters (L, M), we can solve Eq. (14) for a *least-squares solution* of the corresponding rotation parameters $(\theta_x, \theta_y, \theta_z)$ as follows:

$$\alpha_{LS} = (\Psi^T\Psi)^{-1}\Psi^T\beta. \tag{18}$$

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. This least-squares algorithm has a computational complexity of $O(n \times \#L \times \#M)$, where $\#L$ and $\#M$ are the numbers of search points in L and M, respectively.

3.2 MAP Estimation

This section explains how our algorithm can use statistical methods 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

$$\beta = \Psi\alpha + \epsilon, \tag{19}$$

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

$$\alpha_{ML} = \arg\max_\alpha P(\beta \mid \alpha) \tag{20}$$

makes use of whatever information we have about the distribution of the observations (displacement vectors). This ML estimate is equal to [3]:

$$\alpha_{ML} = (\frac{1}{\sigma_\beta^2} \Psi^T \Psi)^{-1} \Psi^T \frac{1}{\sigma_\beta^2} \beta = (\Psi^T \Psi)^{-1} \Psi^T \beta. \tag{21}$$

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,

$$P(\alpha|\beta)=\frac{P(\beta|\alpha)P(\alpha)}{P(\beta)}, \tag{22}$$

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

$$\alpha_{MAP}=\arg\max_{\alpha} P(\alpha|\beta)=\arg\max_{\alpha} P(\beta|\alpha)P(\alpha), \tag{23}$$

which maximizes 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 would be out of view. We further assume as prior information that θ_x, θ_y, θ_z are independently and normally distributed with zero mean and identical variance σ_α^2. This assumption yields [3]:

$$\alpha_{MAP}=(\frac{1}{\sigma_\beta^2}\Psi^T\Psi+\frac{1}{\sigma_\alpha^2}I)^{-1}\frac{1}{\sigma_\beta^2}\Psi^T\beta=(\Psi^T\Psi+\frac{\sigma_\beta^2}{\sigma_\alpha^2}I)^{-1}\Psi^T\beta. \tag{24}$$

The *confidence factor* $\sigma_\beta^2/\sigma_\alpha^2$ reflects the confidence of the prior information relative to that of the displacement vectors. The larger $\sigma_\beta^2/\sigma_\alpha^2$ is, the greater is the 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; 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). The MAP estimate has the same computational complexity as the least-squares algorithm in Section 3.1 because we only add the confidence factor to each diagonal element of $\Psi^T\Psi$ before matrix inversion.

Synthetic simulations [7, 8] 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 the 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 ($\theta_x = 0°$, $\theta_y = 0°$, $\theta_z = 0°$), 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 results are achieved when the noise level is low, such as, *SNR*[1] ≥ 50dB. Hence, synthetic simulations indicate that more confidence should be placed 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 [7, 8], it was found via simulations that the optimal confidence factor increases as the noise increases for cases where the signal-to-noise ratio was ≤ 40dB. 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.

The MAP estimate and parameter search reported here are heuristic because we only assume the most general case of white Gaussian noise on the elements of the linear systems of equations. If more statistical information about noise is available in advance, the information can be utilized in a similar way.

3.3 Multi-Scale Parameter Searching and Time-Domain Smoothing

In this section, we will discuss how multi-scale searching of motion parameter space can improve accuracy and how time-domain smoothing of recovered motion parameters can reduce noise. Since the velocity equation is valid only instantaneously, each snapshot of a scene shows rigid body motion and is described more accurately by Eq. (7). 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) of the bounded and discretized motion parameter space around the initial estimate on a finer scale. This will be explained next.

We next return to the true motion equations of a rigid body, define the error term, and locally search the bounded and discretized motion parameter space around the initial estimate on a finer scale. Using Eq. (8) and dividing X' and Y' by Z' yields

$$x' = \frac{X'}{Z'} = \frac{C_z C_y x + (S_x S_y - C_x C_y S_z) y + (C_x S_y + S_x C_y S_z) + \frac{T_x}{Z}}{-S_y C_z x + (C_y S_x + C_x S_y S_z) y + (C_x C_y - S_x S_y S_z) + \frac{T_z}{Z}} \tag{25}$$

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

By cancelling Z from the above two equations, assuming $T_z \neq 0$, dividing

[1] The noise added to the beginning points (x,y) and ending points (x',y') is white Gaussian noise, and the signal-to-noise ratio is defined as SNR = $20\log\frac{\sigma_{\text{noise}}}{\sigma(x, y, x', y')}$.

both sizes 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 as

$$
\begin{aligned}
Error(L, M, \theta_x, \theta_y, \theta_z) = {} & (C_yC_z + LS_yC_z)xy' + (S_xS_y - C_xC_yS_z - LC_xS_yS_z - LS_xC_y)yy' \\
& + (C_xS_y + S_xC_yS_z - LC_xC_y + LS_xS_yS_z)y' - (MS_yC_z + S_z)xx' \\
& + (MC_xS_yS_z + MS_xC_y - C_xC_z)yx' + (MC_xC_y - MS_xS_yS_z + S_xC_z)x' \\
& + (LS_z - MC_yC_z)x + (LC_xC_z - MS_xS_y + MC_xC_yS_z)y \\
& - (MC_xS_y + MS_xC_yS_z + LS_xC_z).
\end{aligned} \tag{27}
$$

Ideally (in the noise-free case) $Error = 0$. However, in practical experiments, $Error \neq 0$, and we find the optimal $(L, M, \theta_x, \theta_y, \theta_z)$ which minimize $\Sigma(Error)^2$ over all corresponding pairs. Multi-scale searching is done by locally searching around the initial estimates on a finer scale. We search the discretized and bounded parameter space of $[\theta_x - 1°, \theta_x +1°]$, $[\theta_y - 1°, \theta_y + 1°]$, $[\theta_z - 1°, \theta_z +1°]$ with a step size of 0.1° and search that of $[L - 0.05, L + 0.05]$, $[M - 0.05, M + 0.05]$ with a 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. Multi-scale searching improves the accuracy of the motion parameter estimates and avoids high computational cost since searching the complete motion parameter space with such a fine scale would be computationally expensive. This multi-scale searching has a computational complexity of $O(\#L \times \#M \times \Theta_x \times \Theta_y \times \Theta_z)$, where $\#L$, $\#M$, Θ_x, Θ_y, Θ_z are the numbers of fine scale search points in L, M, θ_x, θ_y, θ_z, respectively.

After multi-scale searching to compute more accurate motion parameters, we can substitute the parameters back into Eq. (25) or (26) to compute $\frac{Z}{T_z}$, i.e. the depth of the object surface up to a scaling factor by:

$$
\frac{Z}{T_z} = \frac{x' - L}{S_yC_zxx' - (C_xS_yS_z + S_xC_y)x'y - (C_xC_y - S_xS_yS_z)x' + C_yC_zx + (S_xS_y - C_xC_yS_z)y + \xi} \tag{28}
$$

$$
\frac{Z}{T_z} = \frac{y' - M}{S_yC_zxy' - (C_xS_yS_z + S_xC_y)y'y - (C_xC_y - S_xS_yS_z)y' + S_zx + C_xC_zy - S_xC_z}, \tag{29}
$$

where $\xi = (C_xS_y + S_xC_yS_z)$. The choice of which of the above equations or combination of equations to use depends on the numerical considerations and motion. For example, when T_y is dominant (the motion is mainly horizontal translation), Eq. (29) is better than Eq. (28) because the situation is similar to stereo vision in recovering an 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. (28) is better than Eq. (29).

If we know $T_z = 0$ in advance, we can still recover the motion parameters and object shape using a similar method only with a much simpler set of equations.

Although the least-squares algorithm with MAP estimation and multi-scale searching has been found to be robust in many cases, motion and shape recovery of real world images is sometimes sensitive to noise, and the estimated motion parameters have errors due to ambiguity where very different 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:

$$\theta_x(j) = med\{\theta_x(i) : i = j - m, j - m + 1, ..., j, ..., j + m\}. \tag{30}$$

We have found this time-domain median smoothing to perform well in reducing errors of estimated motion parameters, as shown in experiments which will be described 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 any ambiguity. In our algorithm, we use only displacement vectors and achieve satisfactory performance; however, the rotation, scaling, and intensity ratio and bias contain rich motion information and should further improve the performance. Thus, their use to avoid motion ambiguity is a good subject for future research. 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

$$|d_{x,i} - d_{x,j}| < 0.1 \cdot \max(|d_{x,i}|, |d_{y,i}|) \text{ and } |d_{y,i} - d_{y,j}| < 0.1 \cdot \max(|d_{x,i}|, |d_{y,j}|). \tag{31}$$

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 the 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.

Fig. 5 shows three frames from a 6-frame toy truck image sequence with

[2] As 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.

no rotation ($\theta_x = \theta_y = \theta_z = 0°$) 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, the camera yaw was θ_x, pitch is θ_y, and roll is θ_z, all in degrees. The translation direction was T_x points upward, T_y points rightward, and T_z points toward the objects. The lower left truck was the closest (170mm away), the lower right truck was in the middle (220mm away), and the upper tractor truck was the farthest (360mm away).

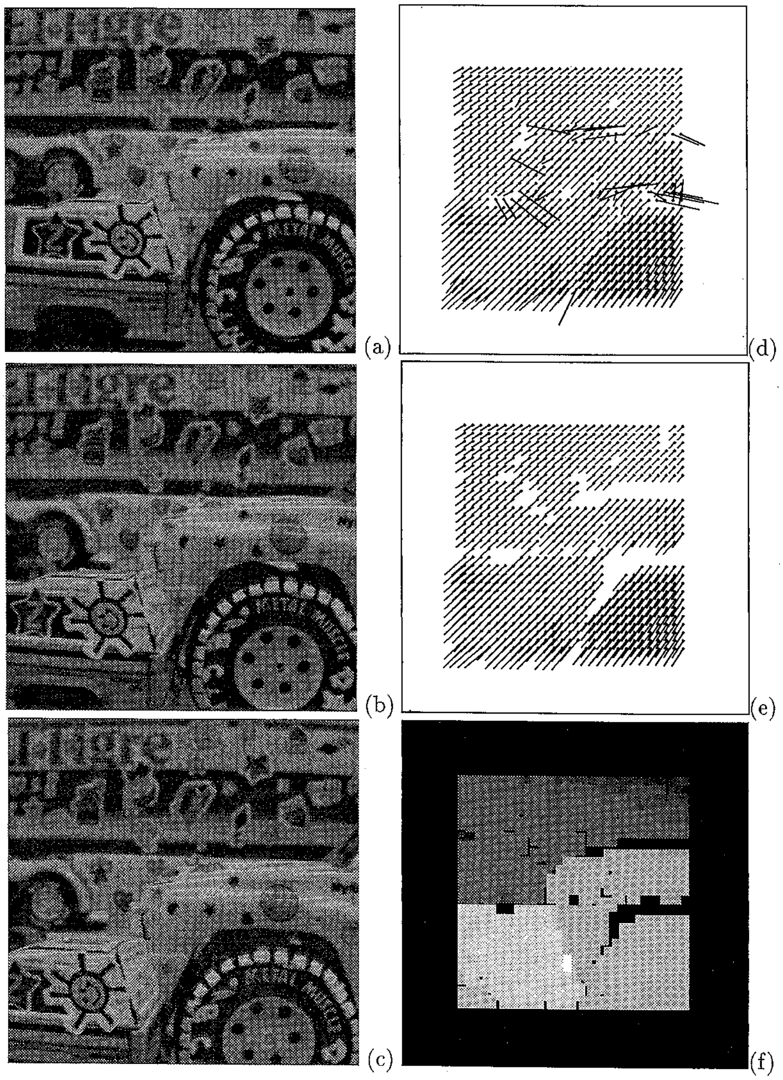


Fig. 5. Toy truck image sequence, $\theta_x = \theta_y = \theta_z = 0°$, $T_x = T_y = T_z = -5$mm. (a) Frame 3 (386×386 pixels, 8 bit/pixel). (b) Frame 4. (c) Frame 5 of the image sequence. (d) Result of 2-D affine block matching of (a) and (b). (e) Result of nonlinear outlier removal on (d). (f) Range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

We used the 2-D displacement vectors estimated by the 2-D affine model because the estimates were dense and accurate as shown in Fig. 5(d). As shown in Fig. 5(e), the nonlinear outlier removal algorithm performs well in removing the mismatches around the occlusion boundaries. We used $\sigma_\beta^2 / \sigma_\alpha^2 = 0.01$ 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 were almost zero (compared to 40 degrees of FOV), and the translation direction $(T_x/T_z,\ T_y/T_z\ ,\ 1)$ had at most 20% error. Because the motion was constant, we could apply time-domain median smoothing on motion parameters and have $\theta_x = 0.349°$, $\theta_y = -0.305°$, $\theta_z = 0.009°$, $T_x/T_z = 0.950$, and $T_y/T_z = 0.950$; this shows an improvement over most individual estimates. We used the above motion parameters to compute the object shape in the form of depth map. The average error for the depth map in Fig. 5(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 repeated the depth estimate for the 7 × 7 pixels around the block center. The two black stripes on the right side of the range image are not errors but indicate that there is no depth information because the mismatches caused by occlusion boundaries were removed by nonlinear outlier removal.

Fig. 6 shows three frames from a 21-frame mountain image sequence. As shown in this figure, the non-linear outlier removal algorithm performed well in removing mismatches around occlusion (the boundary between the mountain top and cloud). We used $\sigma_\beta^2 / \sigma_\alpha^2 = 0.01$ in the MAP estimation because the displacement vector field had 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 was a "move and shoot" image sequence; the vehicle did not stop to stabilize, and the road surface was unpaved. The motion between image frames was quite abrupt, and time-domain smoothing of motion parameters was not suitable. The translation was also mainly along the optical axis, so the depth estimates were more sensitive to noise. We suspect that the cloud moved relative to the mountain; thus, this relative motion violated the rigid body constraint. The relative motion might have caused the cloud to appear to be closer than the mountain as shown in the range image.

Table 2. Recovered motion parameters of the toy truck image squence. The measured values are $\theta_x = \theta_y = \theta_z = 0°$ 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 was approximately 50°).

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

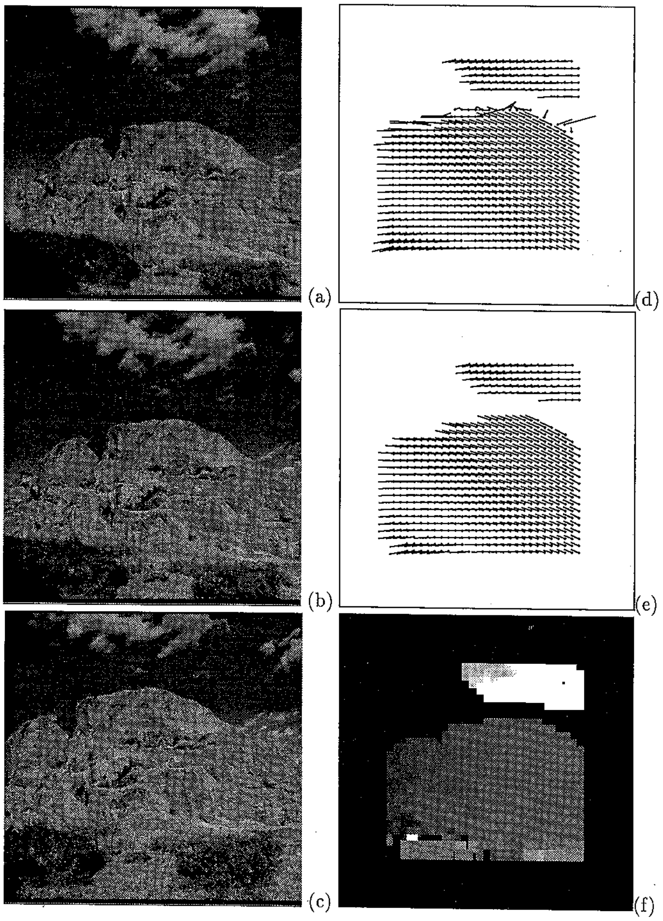


Fig. 6. A mountain image sequence from the University of Massachusetts at Amherst motion data set [4]. (a) Frame 12 (386x386 pixels, 8 bit/pixel). (b) Frame 13. (c) Frame 14 of the image sequence. (d) Result of 2-D affine block matching of (a) and (b). (e) Result of nonlinear outlier removal on (d). (f) Range image of the recovered object depth of (a).

4. CONCLUSION

We have presented a visual motion analysis system which includes a 2-D affine model to determine 2-D motion displacement fields and an algorithm to recover 3-D motion parameters and surface structure under perspective projection. The parameters of the 2-D affine model and velocity equation are found using least-squares algorithms and limited searching in a bounded parameter space. In the 3-D motion and shape recovery algorithm, a simple form of MAP estimation has been 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 varies slowly. The results of many synthetic simulations as well as experiments on real world image squences have indicated 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 ARO Grant DAALO3-86-K-0171 to the Brown-Harvard-MIT Center for Intelligent Control Systems. This research work was also supported in part by the National Science Council of the Republic of China under Grant NSC83-0408-E-002-010.

REFERENCES

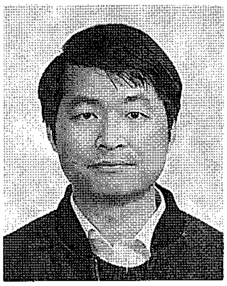
1. Aggarwal, J.K. and Nandhakumar, N., "On the computation of motion from sequences of images-a review," *Proc. IEEE*, Vol. 76, 1988, pp. 917-935.
2. Barnard, S.T. and Thompson, W.B., "Disparity analysis in images," *IEEE Trans. on Patt. Anal. Mach. Intell.*, Vol. 2, 1980, pp. 333-340.
3. Beck, J.V. and Arnold, K.J., *Parameter Estimation in Engineering and Science*, Wiley, New York, 1977.
4. Dutta, R., Manmatha, R., Williams, L.R. and Riseman, E.M., "A data set for quantitative motion analysis," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, 1989, pp. 159-164.
5. Fuh, C.S. and Maragos, P., "Motion displacement estimation using an affine model for image matching," *Optical Engineering*, Vol. 30, 1991, pp. 881-887.
6. Fuh, C.S., Maragos, P. and Vincent, L., "Region-based approaches to visual motion correspondence," Technical Report 91-18, Harvard Robotics Lab., 1991.
7. Fuh, C.S., "Visual motion analysis: estimating and interpreting displacement fields," Ph.D. Thesis, Division of Applied Sciences, Harvard University, 1992.
8. Fuh, C.S. and Maragos, P., "Affine models for motion and shape recovery,"

Proc. SPIE Visual Communications and Image Processing, Boston, 1992, pp. 120-134.

9. Gilge, M., "Motion estimation by scene adaptive block matching (SABM) and illumination correction," *Image Processing Algorithms and Techniques*, Proc. SPIE, Vol. 1244, 1990, pp. 355-366.
10. Gruen, A.W. and Baltsavias, E.P., "Adaptive least squares correlation with geometrical constraints," *Computer Vision for Robots*, Proc. SPIE, Vol. 595, 1985, pp. 72-82.
11. Harris, C.G., "Structure from motion under orthographic projection," *Proc. of European Conference on Computer Vision*, 1990, pp. 118-123.
12. Harris, C.G. and Stennett, C., "Rapid – a video-rate object tracker," *Proc. of British Machine Vision Conference*, Oxford, 1990, pp. 73-78.
13. Heeger, D.J. and Jepson, A., "Simple method for computing 3D motion and depth," *Proc. IEEE Intl. Conf. on Comp. Vis.*, 1990, pp. 96-100.
14. Horn, B.K.P. and Schunck, B.G., "Determining optical flow," *Artificial Intelligence*, Vol. 17, 1981, pp. 185-203.
15. Huang, T.S. and Tsai, R.Y., "Image sequence analysis: motion estimation," in Huang, T.S. (ed.), *Image Sequence Analysis*, Springer-Verlag, 1981.
16. Jain, J.R. and Jain, A.K., "Displacement measurement and its application in interframe coding," *IEEE Trans. Commun.*, Vol. 29, 1981, pp. 1799-1808.
17. Kalivas, D.S., Sawchuk, A.A. and Chellappa, R., "Segmentation and 2-D motion estimation of noisy image sequences," *Proc. IEEE Int'l. Conf. Acoust., Speech, Signal Process.*, New York, 1988, pp. 1076-1079.
18. Koenderink, J.J. and Van Doorn, A.J., "Affine structure from motion," *Journal of Optical Society of America A*, Vol. 8, 1991, pp. 377-385.
19. Longuet-Higgins, H.C. and Prazdny, K., "The Interpretation of a moving retinal image," *Proc. R. Soc. Lond. B*, Vol. 208, 1980, pp. 385-397.
20. Maragos, P., "Pattern spectrum and multiscale shape representation," *IEEE Trans. on Patt. Anal. Mach. Intell.*, Vol. 11, 1989, pp. 701-716.
21. Marr, D., *Vision*, W.H. Freeman, San Francisco, 1982.
22. Matheron, G., *Random Sets and Integral Geometry*, Acad. Press, New York, 1975.
23. Mundy, J.L. and Zisserman A. (ed.), *Geometric Invariance in Computer Vision*, MIT Press, Cambridge, Mass., 1992.
24. Musmann, H.G., Pirsch, P. and Grällert, H.-J., "Advances in picture coding," *Proc. IEEE*, Vol. 73, 1985, pp. 523-548.
25. Netravali, A.N. and Robbins, J.D., "Motion compensated television coding-part I," *Bell Syst. Tech. J.*, Vol. 58, 1979, pp. 631-670.
26. Quan, L. and Mohr, R., "Towards structure from motion for linear features through reference points," *Proc. IEEE Workshop on Visual Motion*, 1991.
27. Serra, J., *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.
28. Tomasi, C. and Kanade, T., "Shape and motion from image streams: a factorization method," Technical Report, School of Computer Science, Carnegie Mellon University, 1991.
29. Tsai, R.Y. and Huang, T.S., "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Trans. Pattern*

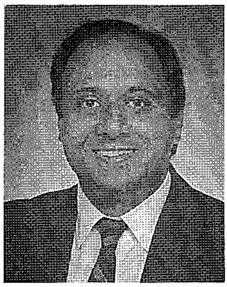
Anal. Mach. Intellig., Vol. 6, 1984, pp. 13-27.

30. Tzou, K.H., Hsing, T.R. and Daly, N.A., "Block-recursive matching algorithm (BRMA) for displacement estimation of video images," *Proc. IEEE Intl. Conf. Acoust., Speech, Signal Process.*, Tampa, 1985, pp. 359-362.
31. Waxman, A.M. and Ullman, S., "Surface structure and three-dimensional motion from image flow kinematics," *The International Journal of Robotics Research*, Vol. 4, 1985, pp. 72-94.
32. Weng, J., Huang, T.S. and Ahuja, N., "Motion and structure from two perspective views: algorithms, error analysis, and error estimation," *IEEE Trans. on Patt. Anal. Mach. Intell.*, Vol. 11, 1989, pp. 451-476.



Chiou-Shann Fuh (傅楸善) received the B.S. degree in information engineering from National Taiwan University, Taipei, Taiwan, in 1983, the M.S. degree in computer science from Pennsylvania State University, University Park, in 1987, and the Ph.D. degree in computer science from Harvard University, Cambridge, in 1992.

He was with AT&T Bell Laboratories and engaged in performance monitoring of switching networks from 1992 to 1993. Since 1993, he has been an Associate Professor in the Computer Science and Information Engineering Department at National Taiwan University, Taipei, Taiwan. His current research interests include digital image processing, computer vision, pattern recognition, and mathematical morphology.



Petros Maragos received the Diploma degree in electrical engineering from the National Technical University of Athens, Greece, in 1980, and the M.S.E.E. and Ph.D. degrees from the Georgia Institute of Technology, Atlanta, in 1982 and 1985.

In 1985, he joined the faculty of the Division of Applied Sciences at Harvard University, where he worked as Assistant (1985-1989) and Associate Professor (1989-1993) of Electrical Engineering. He has also been a consultant to Xerox in research on document image processing and to the Greek Institute for Language and Speech Processing. In 1993, he joined the faculty of the School of Electrical & Computer Engineering at Georgia Tech. His research and teaching activities have been in the general areas of signal processing, systems theory, communications, applied mathematics, and their application to image processing and computer vision, and computer speech processing and recognition.

Dr. Maragos has received several awards for his research work, including: a National Science Foundation Presidential Young Investigator Award (1987); the IEEE Signal Processing Society's 1988 paper Award for the paper 'Morphological

Filters'; the IEEE Signal processing Society's 1994 Senior Award as a corecipient; the 1995 IEEE Baker Prize Award for the paper 'Energy Separation in Signal Modulations with Application to Speech Analysis' as a co-recipient. He has also served as Associate Editor for IEEE Transactions on Signal and on Image Processing; general Chairman for the 1992 SPIE Conference on Visual Communications and Image Processing; President of the International Society of Mathematical Morphology.