Proc. Natl. Sci. Counc. ROC(A)
Vol. 24, No. 6, 2000. pp. 450-462

Spatio-Temporal Feature Tracking and Multi-target Motion Shape Analysis

JINSHIUH TAUR

Department of Electrical Engineering
National Chung-Hsing University
Taichung, Taiwan, R.O.C.

(Received October 25, 1999; Accepted April 21, 2000)

ABSTRACT

Three-dimension shape reconstruction is one of the important research areas in object recognition and image understanding. A structure-from-motion problem as originally proposed by C. Tomasi and T. Kanade in 1992 has attracted a lot of attention. It is based on the singular value decomposition (SVD) approach. In this paper, it is extended to cope with the multi-target case. That is, given a sequence of 2-D video images of multiple moving targets, the goal is to compute the 3-D motion of the targets and reconstruct their 3-D shapes. This is further extended to the multi-camera-multi-target problem. First, a robust algorithm which enhances the reliability of the block matching techniques is proposed for fast tracking of feature points in a sequence of images. Then the feature points are mapped onto their corresponding objects using an algebraic method based on the subspace clustering method and principal singular vector (PSV). Thereafter, the motion and shape may be estimated from a matrix factorization using SVD. We demonstrate the effectiveness of the algorithms in tracking and reconstruction of the shape information using both artificially created data and a real image sequence in somewhat controlled environments.

Key Words: singular value decomposition, feature tracking, shape reconstruction from motion, image understanding

I. Introduction

In the structure-from-motion problem proposed by Tomasi and Kanade (1992), the shape (the relative position of feature points) on "one" moving object can be obtained from a sequence of the images using the singular value decomposition (SVD) approach. The generalization to the paraperspective projection case was introduced in Poelman and Kanade (1997). Morita and Kanade (1997) proposed a sequential factorization method to obtain shape and motion information in real-time applications. Recently, Marugame *et al*. (1999) proposed a framework to recover the structure of an object using a scaled orthographic and perspective views. In these studies, there is only one moving object in the image sequence. In this paper, we consider a generalized situation when there are multiple moving objects in the image sequence. The image sequences from several cameras are also discussed. The study is based on the assumption of orthographic views.

The multi-target motion-shape-estimation (MSE) problem is: *Given a sequence of 2-D video images of multiple moving targets, the goal is to compute the 3-D motion of the targets and reconstruct their 3-D shapes*. This can be further extended to multi-camera-multi-target MSE, with potential application to the 3-D occlusion problem. After collecting feature points (FPs) which are sequentially tracked by a video system, the SVD may be applied to a measurement matrix formed by the FPs. The distribution of singular values will first reveal the information about the number of objects at hand. Then, using an algebraic method based on the subspace clustering method and Principal Singular Vector (PSV) analysis, the FPs may be mapped onto their corresponding objects. Thereafter, the motion and shape may be estimated from a matrix factorization using SVD.

Our method hinges upon the numerical effectiveness and stability of SVD factorization. Also, a robust algorithm is proposed for tracking feature points in a sequence of images. Block matching techniques provide a reliable method for estimating the motion of an object. However, implicit in this method is the assumption that distinctive features, e.g., corners, can be located unambiguously between frames for each block. The correlation between features in image blocks provides the basis for estimating the quality of the match and the motion of the object. We propose a technique that enhances the reliability of the block matching techniques. This method can improve the algorithmic robustness for a broad class of tracking scenarios. We will demonstrate the effectiveness of the algo-

rithms in tracking and reconstruction of the shape information using both artificially created data and a real image sequence in somewhat controlled environments.

This research can serve as a basis for many potential applications. For example, in parking lot (or airport) surveillance applications, it can be used to separate and then recognize different moving targets, so the type and speed of an object can be obtained. Also, it can be utilized to determine the camera motion (reflected in the motion of the background) and the motion of the targets.

This paper is organized as follows: Section II provides the mathematical background and describes some previous works. A discussion of feature point tracking is provided in Section III. Section IV presents the shape from motion method based on the split-and-merge subspace clustering algorithm and principal singular vector analysis. Section V concludes the paper.

II. Background and Previous Works

1. Feature Point Tracking

Robust selection and tracking of feature points is a crucial preprocessing step in this application as well as in many other surveillance applications. For tracking of some moving objects in a video sequence, selection of robust feature(s) of the objects in the initial image is essential. By a robust feature we mean a feature that can be easily detected and accurately located in successive images in the sequence. Many feature selection algorithms have been proposed in the literature. Moravec (1981) proposed an interest operator which can find corner points in an image. Thorpe (1984) improved Moravec's interest operator by first finding edge direction within the window and then computing the directional variance perpendicular to the edge direction. This local variance can be used to select robust feature points. Several other algorithms have been proposed to detect corners. For example, some corner detection operators based on surface fitting were studied in Kitchen and Rosenfeld (1982), and a robust corner detection using curvature scale space was proposed in Mokhtarian and Suomela (1998). Tomasi and Kanade (1991) proposed a elegant feature selection operator that uses a tracking equation to determine if a particular point in an image sequence is trackable.

In this paper, we assume that the feature points are available. Therefore, we concentrate here on fast tracking of the feature points in the image sequence. Block matching techniques for feature point tracking estimate the new position of a feature point when the matching error between blocks is minimized. If the feature point of an object is located on the boundary of the object, and if the background changes between frames, then the block matching algorithm is influenced by the changing background. This motivates the use of a weighting mask to emphasize the object instead of treating the background and the object equally. In Section III, we propose a technique to enhance the reliability of the block matching techniques.

2. SVD Analysis of a Single Moving Target

In this section, SVD is adopted to analyze a set of feature points from a sequence of images in order to recover the shape of a moving object. This is called an MSE problem. Figure 1 shows a coordinated world model for a video-camera imaging system with a target. Here, for convenience of notation, we assume that the rotation center coincides with the (local) coordinate system pertaining to Target A. Let $\boldsymbol{a}$ denote the position vector of one of the P feature points of the target. Moreover, we assume an orthographic projection of the feature points onto the image plane. Projecting this feature point onto the camera's image plane, we have the x-coordinate's value as

$$w_a^i(f) = \boldsymbol{i}\boldsymbol{R}_a(f)\boldsymbol{a} + \boldsymbol{i}\boldsymbol{t}_a(f), \quad \textbf{(1)}$$

where $\boldsymbol{i}$ denotes the vector [1 0 0] and $\boldsymbol{R}_a(f)$, $\boldsymbol{t}_a(f)$, respectively, denote the rotational matrix and the translational vector for frame f. By means of a simplified notation, we obtain

$$w_a^i(f) = \boldsymbol{R}_a^i(f)\boldsymbol{a} + t_a^i(f). \quad \textbf{(2)}$$

Now we can construct an expanded matrix $\boldsymbol{W}_a^i$ by expanding along two directions. (1) Along the horizontal direction, we gather all the feature vectors of Target A. (2)

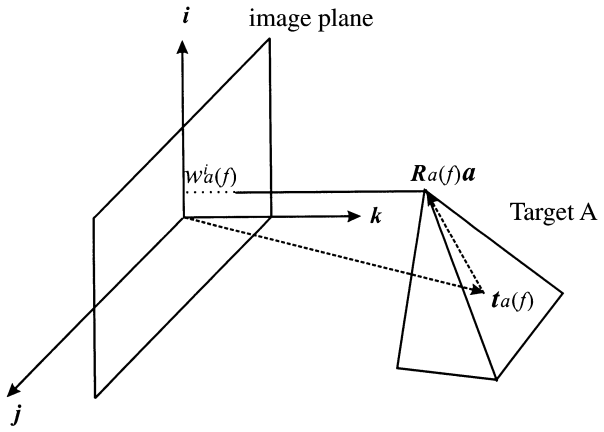

Fig. 1. A coordinated world model for a video-camera imaging system with a target, where $\boldsymbol{a}$ denotes the position vector of one of the P feature points of the target.

Along the vertical direction, we cascade the i-coordinate projected feature positions at time f =1, 2, …, F. This yields a matrix:

$$\boldsymbol{W}_a^i = \boldsymbol{R}_a^i \boldsymbol{S}_a + \boldsymbol{T}_a^i \boldsymbol{E}_a, \tag{3}$$

where

$$\boldsymbol{E}_a = [1\ 1\ \ldots\ 1] \tag{4}$$

is a $1 \times P$ vector.

The dimensions of the $\boldsymbol{W}_a^i$, $\boldsymbol{R}_a^i$, $\boldsymbol{T}_a^i$, and $\boldsymbol{S}_a$ matrices are $F \times P$, $F \times 3$, $F \times 1$ and $3 \times P$, respectively. The shape matrix $\boldsymbol{S}_a = [\boldsymbol{a}(1) \,|\, \boldsymbol{a}(2) \,|\, \cdots \,|\, \boldsymbol{a}(P)]$ is formed from all the P column feature vectors belonging to Target A. The matrix $\boldsymbol{R}_a^i$ is a cascadation of the rotational matrices of different times, i.e.,

$$\boldsymbol{R}_a^i = \left[R_a^i(1)^T \,\middle|\, R_a^i(2)^T \,\middle|\, \sqcup \,\middle|\, R_a^i(F)^T \right]^T.$$

Similarly,

$$\boldsymbol{T}_a^i = \left[t_a^i(1) \middle| t_a^i(2) \middle| \sqcup \middle| t_a^i(F) \right]^T.$$

For the y-coordinate in the image plane, we have

$$w_a^j(f) = \boldsymbol{j}\boldsymbol{R}_a(f)\boldsymbol{a} + \boldsymbol{j}\boldsymbol{t}_a(f) \tag{5}$$

$$= R_a^j(f)\boldsymbol{a} + t_a^j(f), \tag{6}$$

where $\boldsymbol{j}$ denotes the vector [0 1 0]. For the y-axis, we have another expanded matrix:

$$\boldsymbol{W}_a^j = \boldsymbol{R}_a^j \boldsymbol{S}_a + \boldsymbol{T}_a^j \boldsymbol{E}_a. \tag{7}$$

Now let us stack the x- and y- image measurement matrices:

$$\boldsymbol{W}_a = \begin{bmatrix} \boldsymbol{W}_a^i \\ \boldsymbol{W}_a^j \end{bmatrix} = \boldsymbol{R}_a \boldsymbol{S}_a + \boldsymbol{T}_a \boldsymbol{E}_a, \tag{8}$$

where

$$\boldsymbol{R}_a = \begin{bmatrix} \boldsymbol{R}_a^i \\ \boldsymbol{R}_a^j \end{bmatrix} \text{ and } \boldsymbol{T}_a = \begin{bmatrix} \boldsymbol{T}_a^i \\ \boldsymbol{T}_a^j \end{bmatrix}.$$

The dimensions of the matrices $\boldsymbol{W}_a$, $\boldsymbol{R}_a$ and $\boldsymbol{T}_a$ are $2F \times P$, $2F \times 3$ and $2F \times 1$, respectively.

Furthermore, without loss of generality, we can always set the rotation center to be the (moving) centroid of the feature points on the target. (Consequently, for the single and multi-target cases, the rotation center of each object coincides with its individual mass center.) From now on, our discussion will be based on this representation; i.e., the origin of the (local) coordinate system is at the center of mass of the feature points of the target. Under this coordinate system, $\bar{\boldsymbol{S}}_a = 0$, where $\bar{\boldsymbol{S}}_a$ denotes the average of $\boldsymbol{S}_a$. This in turn implies that the average (or the sum) of the columns of $\boldsymbol{S}_a$ will be equal to zero. Therefore, it is obvious that the rows of the matrix $\boldsymbol{S}_a$ in Eq. (8) are orthogonal to $\boldsymbol{E}_a$, i.e.,

$$\boldsymbol{S}_a \boldsymbol{E}_a^T = 0.$$

In the following, this property will be exploited to separate the rotational component from the translational component.

For the single target case, the measurement matrix

$$\boldsymbol{W} = \boldsymbol{W}_a = [\boldsymbol{T}_a | \boldsymbol{R}_a] \begin{bmatrix} \boldsymbol{E}_a \\ \boldsymbol{S}_a \end{bmatrix} \tag{9}$$

$$\equiv \boldsymbol{M}\boldsymbol{S}, \tag{10}$$

where $\boldsymbol{M}_{2F\times 4}$ and $\boldsymbol{S}_{4\times P}$ are defined in an obvious way. The measurement matrix $\boldsymbol{W}$ is usually corrupted by noise, whose existence is due to the measurement noises, other acquisition errors, or the assumption of orthographic projection. The SVD technique permits a large number of points and frames (possibly corrupted by noise) to be processed in a computationally efficient and numerically stable way:

$$\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}^{1/2}\boldsymbol{\Sigma}^{1/2}\boldsymbol{V}.$$

If the SNR is sufficiently large, then the number of predominant singular values in $\boldsymbol{\Sigma}$ should be less than or equal to 4; i.e., the dimensions of $\boldsymbol{U}$ and $\boldsymbol{V}$ should be consistent with those of $\boldsymbol{M}$ and $\boldsymbol{S}$. Note also that

$$\boldsymbol{W} = \boldsymbol{M}\boldsymbol{Q}^{-1}\boldsymbol{Q}\boldsymbol{S}$$

for some invertible matrix $\boldsymbol{Q}$. Therefore, the matrices $\boldsymbol{S}$ and $\boldsymbol{V}$ are related by a transformation matrix $\boldsymbol{Q}$:

$$\boldsymbol{S} = \boldsymbol{Q}^{-1}\boldsymbol{\Sigma}^{1/2}\boldsymbol{V}.$$

It follows that

$$\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}^{1/2}\boldsymbol{Q}.$$

In the single target case, the translational component can be removed by subtracting the (horizontal) average of the measurement matrix (which will not be so easy for multiple target case):

$$\tilde{\boldsymbol{W}} = \boldsymbol{W} - \boldsymbol{W}\boldsymbol{E}_a^T \boldsymbol{E}_a / P \tag{11}$$

$$= \boldsymbol{R}_a\boldsymbol{S}_a + \boldsymbol{T}_a\boldsymbol{E}_a - \boldsymbol{R}_a\bar{\boldsymbol{S}}_a - \boldsymbol{T}_a\bar{\boldsymbol{E}}_a \quad \textbf{(12)}$$

$$= \boldsymbol{R}_a\left[\boldsymbol{S}_a - \bar{\boldsymbol{S}}_a\right], \quad \textbf{(13)}$$

where $\bar{\boldsymbol{S}}_a = \boldsymbol{S}_a\boldsymbol{E}_a^T\boldsymbol{E}_a/P = 0$ and $\bar{\boldsymbol{E}}_a$ denotes the average of $\boldsymbol{E}_a$. Since the elements of $\boldsymbol{E}_a$ are all identical, the translational component $\boldsymbol{T}_a\boldsymbol{E}_a$ is exactly cancelled by its average. This yields a very simple formulation:

$$\tilde{\boldsymbol{W}} = \boldsymbol{R}_a\boldsymbol{S}_a. \quad \textbf{(14)}$$

An important rank property suggested by Eq. (14) is closely related to the fact that three pictures of four points of a rigid body determine structure and motion under orthography (Ullman, 1979). This property was expressed as a rank theorem by Tomasi and Kanade (1992).

Theorem II.1 (Rank Theorem for Noise-free Measurement Matrix). The matrix $\tilde{\boldsymbol{W}}$ given in Eq. (14) has a generic rank of 3.

The rank theorem captures the nature of the redundancy that exists in an image sequence. This implies that the best possible shape and rotation estimate can be obtained by considering only the three greatest singular values of $\tilde{\boldsymbol{W}}$, together with the corresponding left and right eigenvectors. Based on this, Tomasi and Kanade (1992) developed a robust SVD-based factorization method, which takes advantage of numerically well behaved SVD routines. Tomasi and Kanade (1992) proposed a scheme to recover $\boldsymbol{Q}$, based on known constraints on $\boldsymbol{R}_a$, and to subsequently find an exact solution of the motion and shape matrices $\boldsymbol{R}_a$ and $\boldsymbol{S}_a$.

III. Feature Point Tracking Using Weighted Masks

Robust tracking of feature points is a crucial preprocessing step in this application as well as in many other surveillance applications. The tracking results are usually influenced by the following factors:

(1) Focus of the camera: If the objects are widely separated, all the objects can not be in good focus at the same time. This may affect the accuracy of tracking, especially for complex objects.

(2) Type and speed of motion: The image of the object near the feature point can change a lot during fast rotation. In this case, the feature point will probably drift away from the correct position.

(3) Background: If the selected feature point moves close to the boundary of the object, some of the background will be included in the computation of the current position of the feature point.

(4) Perspective distortion: When the objects get close to the camera, the distortion in perspective can be serious enough to bias the tracking accuracy.

In the following, we will propose a technique for tracking feature points which enhances the reliability of the block matching technique. The goal is to emphasize the object region instead of treating the background and the object equally (Taur, 1995). This is useful when the feature point is close to the boundary of the object. To locate the position with the minimum matching error, the matching errors of the pixels in the block are weighted by a mask obtained from the estimated quality of matching and the motion of the feature point. We will demonstrate that this algorithm yields robustness in a scenario of tracking cars in a parking lot and in an image sequence containing two simple moving objects.

Weighted Block Matching. In order to reduce the influence of the background in the matching process, we estimate which elements in the matching window belong to the object under consideration and emphasize these elements by using a weighting mask. The weighting mask is updated frame by frame according to the following equation:

$$W_m^{(i)} = W^{(i-1)} * \alpha_1 + M_1, \quad \textbf{(15)}$$

where α_1 is the forgetting factor and the motion mask M_1 is computed as follows. First the difference between image blocks from the current image and the reference image is computed at the position of the feature point on the reference image. Let B denote the block region containing the feature point:

$$I_d = abs(I_r(x,y) - I_c(x,y)),\ (x,y) \in B.$$

Then I_d is thresholded. That is, if the difference is larger than a certain amount, then the corresponding element in the motion mask will be set to one, which means it is very likely that the element is near a moving object. Then dangling points and small holes in the mask are removed. $W_m^{(i)}$ is used to compute the best match. Once the best match is found, the new weighting mask is computed as follows:

$$W^{(i)} = W_m^{(i)} * \alpha_2 + M_2, \quad \textbf{(16)}$$

where α_2 is a forgetting factor.

The match mask M_2 is computed by setting a threshold for the difference between the current image and the reference image at the positions of the feature point. That is, assume that the image block B_m in the current image matches best block B in the reference image under the mask $W_m^{(i)}$. The difference between the positions of B_m and B is described by an offset vector (d_x, d_y). Then we have

$$I_d = abs(I_r(x,y) - I_c(x+d_x, y+d_y)),$$

$$(x,y) \in B \text{ and } (x+d_x, y+d_y) \in B_m.$$

Then dangling points and small holes in the mask are removed. If the difference is smaller than a certain amount, then the corresponding element in the match mask will be set to one, which indicates a good match. If the background is changing a lot, the matching window will have a good match with the object and a poor match with the background. Therefore, M_2 will have ones in the region of the object.

In the simulation, we used an image sequence of a moving car in a parking lot. The lower right corner of the windshield of the moving car was selected as a test feature point. (If the window of a feature point was totally on the body of the car, e.g., the left corner, then the tracking task would be easier.) The window size was 13, and $\alpha_1 = \alpha_2 = 0.6$. If simple block matching was used, the feature points would not be tracked correctly due to the changing background. After the weighting mask was applied, the feature points could be tracked. The tracking results are shown in Fig. 2. For the purpose of illustration, Fig. 2(a) and (b) show the magnified pictures around the moving car in Fig. 2(c) and (d) with one feature point (indicated with white dots). Note that the background and the size and orientation of the car change considerably in different frames, yet the algorithm is able to track the feature point.

In Figs. 3 and 4, we show some typical masks in different scenarios. They show motion masks and matching masks from the upper left to the lower right corresponding to the following situations: (1) a car is moving fast on a uniform background, (2) a car is moving slowly on a uniform background, (3) a car is moving fast on a changing background, and (4) a car is moving slowly on a changing background. From Figs. 3 and 4, we can see that the mask covers the object region in most of the cases. In the case where the car is moving fast on a uniform background (the leftmost figures), the matching is still valid since the background is uniform.

The same tracking algorithm was applied to the

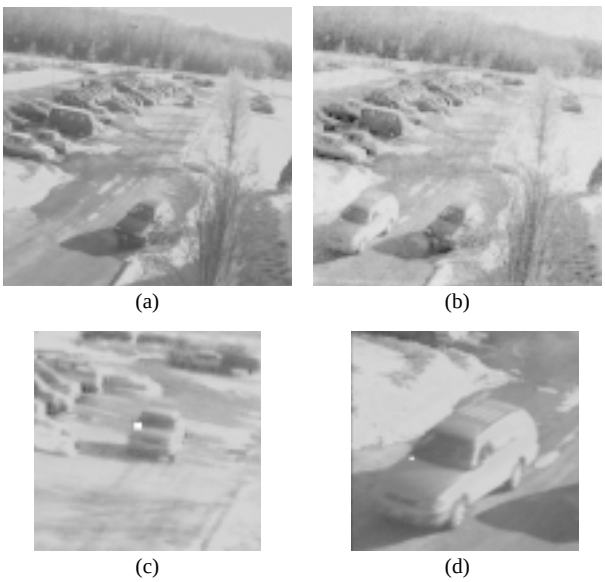

Fig. 2. (a) Starting position. (b) Ending position. (c) and (d) are magnified images of the car.

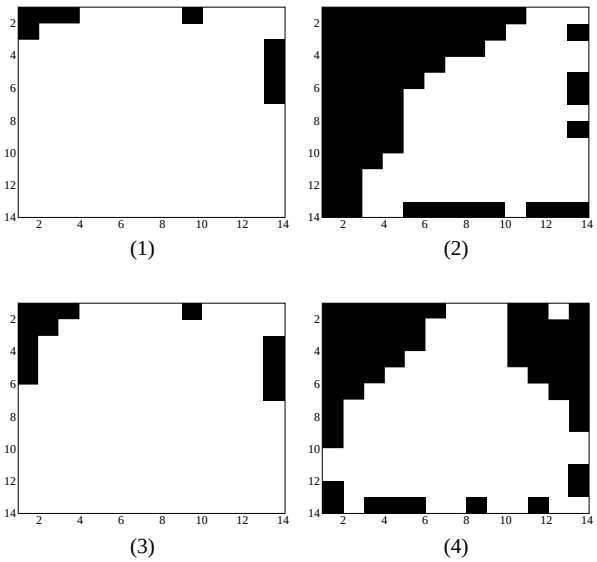

Fig. 3. Motion masks of the feature point in different situations. The white and black elements denote 1's and 0's, respectively.

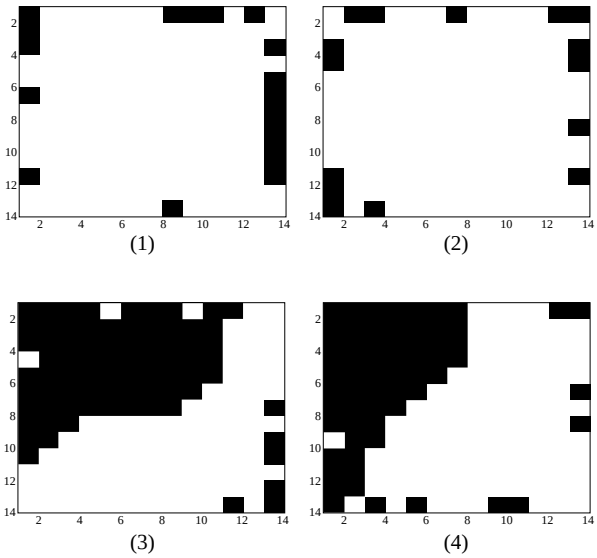

Fig. 4. Match masks of the same feature point in Fig. 3 in different situations. The white and black elements denote 1's and 0's, respectively.

image sequence recorded in a laboratory (sequence LAB-2OBJ) with the following equipment. The system for the experiments is depicted in Fig. 5, which shows a personal computer and a video grabbing system. The video grabbing system is composed of three parts: a personal animation recorder, transducer, and high speed hard disk. It can record the video signals onto a hard disk in real time. The resolution is 480×740 and 256 gray levels per pixel. The source can be any S-Video or NTSC signal. The signal from the camera is shown on a video monitor. The video signals are compressed before they are stored onto a hard disk. Therefore, the image quality is degraded a little bit. In the experiment, there were totally 150 frames in the sequence. One solid object and one hollow object were moving on a surface. The gray levels in the background were changing gradually. There were also abrupt changes of gray levels in the background. Nine feature points were selected manually. (In order to recover the shape and the motion information, at least four feature points were required for each object.) The tracking results are shown in Figs. 6 – 8. The circles indicate the feature points. We can see that the positions of the feature points remain reasonably accurate although the sizes and orientations of the objects change a lot.

IV. Multi-Camera-Multi-Target Motion-Shape Analysis

In this section, we consider the structure-from-motion problem in the situation where there are multiple cameras and multiple moving objects in the image sequences. That is, given long sequences of images, we wish to construct a three-dimensional model of the moving targets.

1. SVD for the Multi-Target MSE Problem

For the single-object MSE problem, let us define $\boldsymbol{M} = [\boldsymbol{T}_a \,|\, \boldsymbol{R}_a]$ and $\boldsymbol{S} = [\boldsymbol{E}_a / \boldsymbol{S}_a]$ using the notation given in Section II.2. Then the measurement matrix can be expressed as $\boldsymbol{W} = \boldsymbol{W}_a = \boldsymbol{MS}$. This factorization suggests that the matrix $\boldsymbol{W}$ has a generic rank of 4. If we further subtract the mass center from $\boldsymbol{W}$, the resulting matrix

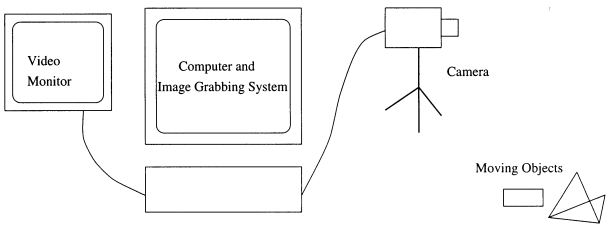

Fig. 5. Set-up of the equipment for image sequence grabbing.

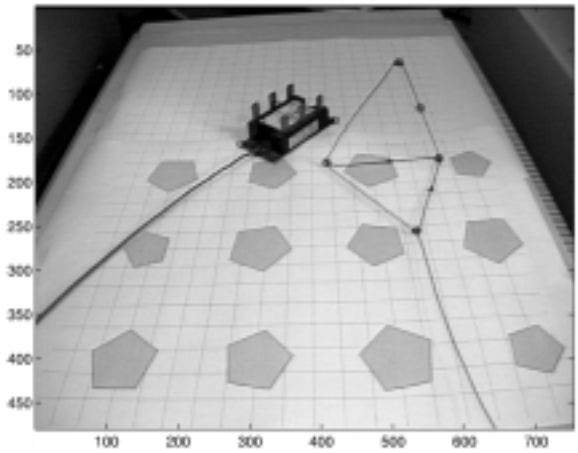

Fig. 6. The first frame of LAB2OBJ in the experiment. The circle shows the selected feature points.

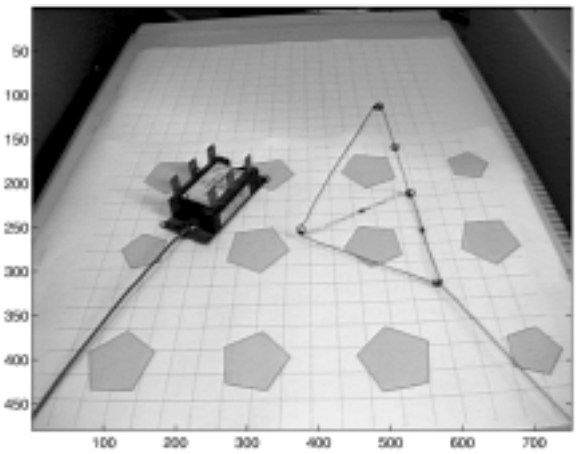

Fig. 7. The 75th frame of LAB2OBJ in the experiment. The circle shows the tracked feature points.

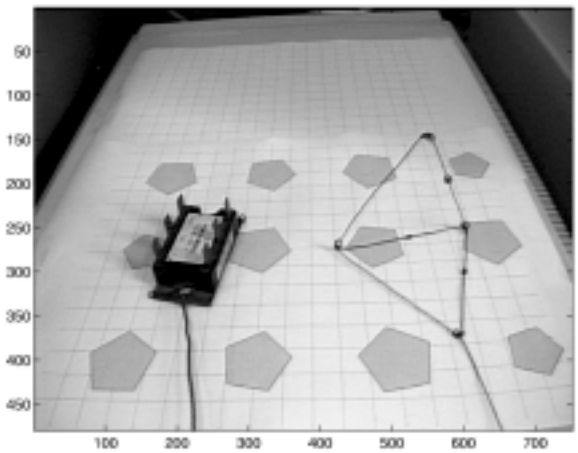

Fig. 8. The last frame of sequence LAB2OBJ. The circle shows the tracked feature points.

will have a generic rank of 3 (Tomasi and Kanade, 1992).

For the multi-object MSE problem, a new challenge arises since we have to distinguish the feature points of adjacent objects so that they can be correctly classified into their corresponding objects. Let us, for simplicity, concentrate on two targets, A and B, with their shapes denoted by $\boldsymbol{S}_a$ and $\boldsymbol{S}_b$.

Like the single-target case, the measurement matrices for A and B can be derived as follows: $\boldsymbol{W}_a = \boldsymbol{R}_a\boldsymbol{S}_a + \boldsymbol{T}_a\boldsymbol{E}_a$ and $\boldsymbol{W}_b = \boldsymbol{R}_b\boldsymbol{S}_b + \boldsymbol{T}_b\boldsymbol{E}_b$. Concatenating these matrices, we have

$$\boldsymbol{W} = \left[\boldsymbol{W}_a | \boldsymbol{W}_b\right] \tag{17}$$

$$= \left[\boldsymbol{T}_a | \boldsymbol{R}_a | \boldsymbol{T}_b | \boldsymbol{R}_b\right] \left[\begin{array}{c|c} \boldsymbol{E}_a & 0 \\ \hline \boldsymbol{S}_a & 0 \\ \hline 0 & \boldsymbol{E}_b \\ \hline 0 & \boldsymbol{S}_b \end{array}\right] \equiv \boldsymbol{MS}. \tag{18}$$

Let P_a and P_b denote the number of feature points in Target A and Target B, respectively. Then the sizes of $\boldsymbol{W}$, $\boldsymbol{M}$ and $\boldsymbol{S}$ are $2F \times (P_a + P_b)$, $2F \times 8$ and $8 \times (P_a + P_b)$, respectively. The rank of a noise-free non-degenerate measurement matrix $\boldsymbol{W}$ should be exactly 8. If there are k targets in the image sequence, the rank of a noise-free measurement matrix $\boldsymbol{W}$ should be exactly $4 \times k$.

A. Main Theorem for Reclustering of Feature Points

The representation in Eq. (18) assumes that the columns from A and B are already separated in correct clusters. This may not be the case in reality. When two targets are close to each other, the FPs may not be pre-aligned in the correct order. Therefore, we have to first recluster the FPs such that the FPs for A are separated from those for B. Furthermore, it is realistic to assume that the measurement matrix $\boldsymbol{W}$ is corrupted by noise and, thus, has full rank.

Computing the SVD of the measurement matrix

$$\boldsymbol{W} = \overline{\boldsymbol{U}}\overline{\boldsymbol{\Sigma}}\overline{\boldsymbol{V}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V} + \boldsymbol{U}'\boldsymbol{\Sigma}'\boldsymbol{U}'$$

and removing the "noise" singular values $\boldsymbol{\Sigma}'$, we have

$$\boldsymbol{W} \approx \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{\Sigma}^{1/2}\boldsymbol{\Sigma}^{1/2}\boldsymbol{V}. \tag{19}$$

Let us assume that the SNR is sufficiently large (ideally noise-free), and that each (rigid-body) object consists of at least 4 or more linear independent FPs and has total freedom of 3-D (rotational and translational) motion. Then the following theorem is valid:

Theorem IV.1 (Subspace Rank Property). Compute the SVD of $\boldsymbol{W}$ and obtain $\boldsymbol{U}$, $\boldsymbol{\Sigma}$, $\boldsymbol{V}$ as given in Eq. (19).

(1) *Total Rank*: The total number of (numerically) non-zero singular values in $\boldsymbol{\Sigma}$ (i.e., those attributed to the objects) will be $4k$, where k is the number of objects. Here an object is, by definition, a rigid body.

(2) *Inclusive Rank Property*: If the column vectors of the matrix $\boldsymbol{V}$ (or, equivalently, $\boldsymbol{\Sigma}^{1/2}\boldsymbol{V}$) are correctly grouped into k clusters, each corresponding to one object, and if the correctly permuted matrix is re-written as

$$\boldsymbol{V} \equiv [\boldsymbol{V}_a \,|\, \boldsymbol{V}_b], \tag{20}$$

then $\boldsymbol{V}_a$ and $\boldsymbol{V}_b$ have (generically) a rank of 4.

(3) *Exclusive Rank Property*: Due to the mutual orthogonality property, any mixture of column vectors from different objects will generally cause the submatrix (comprising of columns from more than one object) to exceed rank 4. In other words, no column in $\boldsymbol{V}_a$ may fall in the span of the submatrix of $\boldsymbol{V}_b$, and vice versa. Generally, any mixture of (5 or more) columns from $\boldsymbol{V}_a$ and $\boldsymbol{V}_b$ will cause the rank to exceed 4. This property may be exploited to prevent over-subscribing of alien columns into an object.

(4) *Uniqueness*: The *inclusive* and *exclusive* rank properties together guarantee the uniqueness of the solution.

Proof. The theorem can be proved by inspecting Eq. (18). In particular, we note that $\boldsymbol{\Sigma}^{1/2}V = \boldsymbol{QS}$ for some nonsingular matrix $\boldsymbol{Q}$. Therefore, $\boldsymbol{V}_a = \boldsymbol{Q}[\boldsymbol{E}_a^T | \boldsymbol{S}_a^T | 0 | 0]^T$. Since $\boldsymbol{Q}$ is nonsingular, $\boldsymbol{V}_a$ must have rank 4. Equation (18) also indicates the mutual rank independency between $\boldsymbol{V}_a$ and $\boldsymbol{V}_b$, thus verifying Part (3). Part (4) follows naturally Parts (2) and (3). □

B. Subspace Clustering Problem

The rank theorem prescribes the common bound shared by FPs from the same (rigid) object. This leads to a general algebraic framework formulated in the so-called a subspace clustering problem.

Definition IV.1 (Subspace Clustering Problem). Given a set of feature vectors $\boldsymbol{V} = \{\upsilon_i\}$, the problem is to find all the (rank-r) objects in $\boldsymbol{V}$ by identifying their corresponding subsets of feature vectors. Here a rank-r object is defined as a subset of $\boldsymbol{V}$ which forms a rank-r subspace. (For example, when applied to the multi-target MSE problem, $r = 4$.)

Algorithm IV.1 (Subspace Clustering Method). For the noise-free case, the following steps may be adopted:

(1) Determine a pool of basis vectors $\boldsymbol{B}$ as a maximally linearly independent subset of $\boldsymbol{V}$. Generally, $\boldsymbol{B}$ should contain exactly $k \times r$ basis vectors.

(2) A subset of r basis vectors in $\boldsymbol{B}$ will be incorporated into a partnership if there exists at least one vector in $\boldsymbol{V}$, but not in $\boldsymbol{B}$, which falls in the span of the subset. The justification for forming such a partnership is that, due to the *exclusive rank property*, if $r + 1$ vectors fall in a span of rank-r subspace, then they can not possibly be from a mixture of two objects; i.e., they belong to the same object. (For notational convenience, the r basis vectors shall be called major members in the partnership.)

(3) Attract other minor members to join the partnership. By the *inclusive rank property*, a vector is elected to membership if and only if it falls in the span of the r basis vectors (i.e., major members).

(4) Continue the process until all the memberships of the k objects (i.e., partnerships) are identified.

A Clustering Example (Noise-Free Case). For simplicity, the object rank is now set to be $r = 2$. Given a set of vectors $\{\upsilon_i, i = 1, 2, \ldots\} = \{A_1, B_1, B_2, C_1, A_2, A_3, C_2, B_3, A_4, B_4, C_3, \ldots\}$, from objects A, B and C, the clustering process can be shown as follows:

Vector	Dependence	Basis Pool	Partnership
1	No	1,	-
2	No	1,2,	-
3	No	1,2,3,	-
4	No	1,2,3,4,	-
5	No	1,2,3,4,5	-
6	Yes (So vector #6 is excluded from the pool.)		
7	No	1,2,3,4,5,7	-
Basis pool is complete and new members are now added:			
6	Yes(on 1,5)	(1,5),2,3,4,7	(1,5 | 6)
8	Yes(on 2,3)	(1,5),(2,3),4,7	(2,3 | 8)
9	Yes(on 1,5)	(1,5),(2,3),4,7	(1,5 | 6,9)
10	Yes(on 2,3)	(1,5),(2,3),4,7	(2,3 | 8,10)
11	Yes(on 4,7)	(1,5),(2,3),(4,7)	(4,7 | 11)

The final clustering result is that the vectors (1,5, 6,9, …) form one object (say, A), (2,3, 8,10, …) form another object (B), and (4,7, 11, ...) form yet a third object (C).

C. Discussion

In real-time applications, recursive extraction of principal components using the adaptive algorithm becomes important. For example, a parallel processing neural model, Adaptive Principal component EXtraction (APEX), may provide a very attractive implementation (Kung and Diamantaras, 1990). Moreover, Morita and Kanade (1997) proposed a sequential factorization method using QR factorization to obtain the principal singular vectors with one target for real-time processing. The computation complexity is reduced to $O(P^2)$, where P is the number of feature points. For the multi-target case, the same approach can be used to obtain $\boldsymbol{V}$. Then the subspace clustering algorithm can be adopted to identify the corresponding object for the feature points. For each vector $\boldsymbol{\upsilon}_i$ not in the set of the basis vectors $\boldsymbol{B}$, we have to check which vectors in $\boldsymbol{B}$ can span $\boldsymbol{\upsilon}_i$. QR factorization can again be used for this purpose. However, if P is large, the computation time may exceed the limit for real-time applications.

In addition to the computation requirement, several other factors need to be taken into account in practical (real-time) applications. For example, when the object is moving, the feature points will sometimes be occluded and reappear later. The chang in the number of feature points and the correspondence problem will make the MSE more difficult. When the number of feature points becomes too small, a scheme to automatically select the feature points is required. Moreover, if the motion models of the object degenerated then the subspace rank property will not hold. An example is the case in which the object is moving in only one dimension. (We can move the camera to make the relative motion non-degenerate if there is only one target.) Another degenerate situation happens when more than one object is moving with very similar translation and rotation models. Therefore, we sometimes have to solve the MSE problem using a batch-type approach. We will focus on these research topics in the future.

2. Multi-Camera-Multi-Target MSE

In many application domains, images taken using multiple cameras can offer vital information. However, one must keep FPs from different objects from being mixed. In this case, the **subspace clustering method** offers a simple solution.

Without loss of generality, let us assume that the first camera is (as before) located at the origin [0,0,0] with the direction of the imaging plane defined by its normal vector [0,0,1]. A second camera, located at a new location, $\boldsymbol{m} = [m_i, m_j, m_k]$, has its own image plane defined by a new normal vector $\boldsymbol{k}_3 = [k_3^1, k_3^2, k_3^3]$ (Fig. 9). Since $\boldsymbol{m}$ is known and remains constant, its shift effect can be removed by first pre-shifting the FPs recorded by the second camera. Therefore, without loss of generality, we simply pre-align the FPs of the second camera so that in the following derivation, we consider in effect $\boldsymbol{m} = 0$.

There exists a common viewing angle from the two cameras since two image planes (assumed to be non-paral-

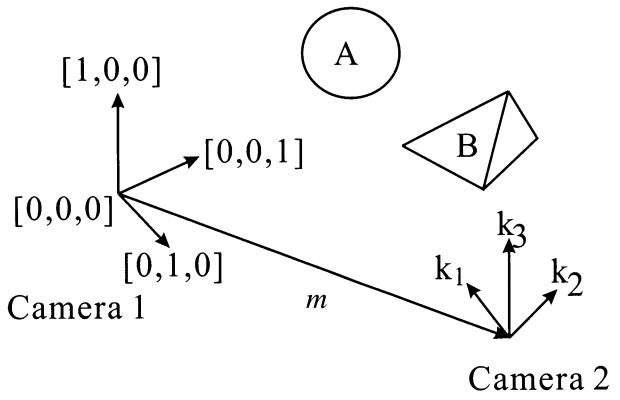

Fig. 9. Coordinate system for two cameras with multiple moving targets.

lel) must intersect on one line, which is orthogonal to both of the normal vectors. Let the direction of the line be denoted as $\boldsymbol{l}$. It is obvious that

$$\boldsymbol{l}^T[0,\ 0,\ 1] = 0 \text{ and } \boldsymbol{l}^T[k_3^1,\ k_3^2,\ k_3^3] = 0.$$

This yields a solution:

$$\boldsymbol{l} = [k_3^2,\ -k_3^1,\ 0].$$

Just as in Eq. (1), with orthographic projection onto the line $\boldsymbol{l}$, the FP is recorded as

$$w_a^l(f) = \boldsymbol{l}R_a(f)\boldsymbol{a} + \boldsymbol{l}t_a(f), \tag{21}$$

and the measurement matrix for the first camera is

$$\boldsymbol{W}_a^l = \boldsymbol{R}_a^l\boldsymbol{S}_a + \boldsymbol{T}_a^l\boldsymbol{E}_a. \tag{22}$$

Similarly, for the second camera, we have another matrix:

$$\overline{\boldsymbol{W}}_a^l = \boldsymbol{R}_a^l\overline{\boldsymbol{S}}_a + \boldsymbol{T}_a^l\overline{\boldsymbol{E}}_a. \tag{23}$$

Assuming that there are two objects (A and B), the total measurement matrix becomes

$$\left[\boldsymbol{W}_a^l \middle| \overline{\boldsymbol{W}}_a^l \middle| \boldsymbol{W}_b^l \middle| \overline{\boldsymbol{W}}_b^l\right] = \left[\boldsymbol{T}_a^l \middle| \boldsymbol{R}_a^l \middle| \boldsymbol{T}_b^l \middle| \boldsymbol{R}_b^l\right] \begin{bmatrix} \boldsymbol{E}_a \middle| \overline{\boldsymbol{E}}_a & 0 \\ \hline \boldsymbol{S}_a \middle| \overline{\boldsymbol{S}}_a & 0 \\ \hline 0 & \boldsymbol{E}_b \middle| \overline{\boldsymbol{E}}_b \\ \hline 0 & \boldsymbol{S}_b \middle| \overline{\boldsymbol{S}}_b \end{bmatrix}.$$

Inspection shows clearly that all the multiple-object rank properties in Theorem IV.1 and the same subspace clustering method remain applicable. Therefore, we can identify the set of FPs in the same object using different cameras.

Note also that when more than two cameras are employed, we can cluster the FPs through fusion of images from any pair of cameras. The final shapes of individual objects can be constructed based on the results of all the image pairs obtained using multiple cameras.

3. Simulation Results

Example IV.1 (Four Moving Targets). The targets considered in the simulation consisted of two cylinders, one block and one pyramid. There were 20 feature points on the cylinders and the block, and 10 points on the pyramid. The order of the feature points was randomly permuted. During the duration of 50 frames, all the targets rotated independently. One frame of the orthographic projection of the four objects is shown in Fig. 10(a), in which the FPs of the objects can not be separated easily, at least not by

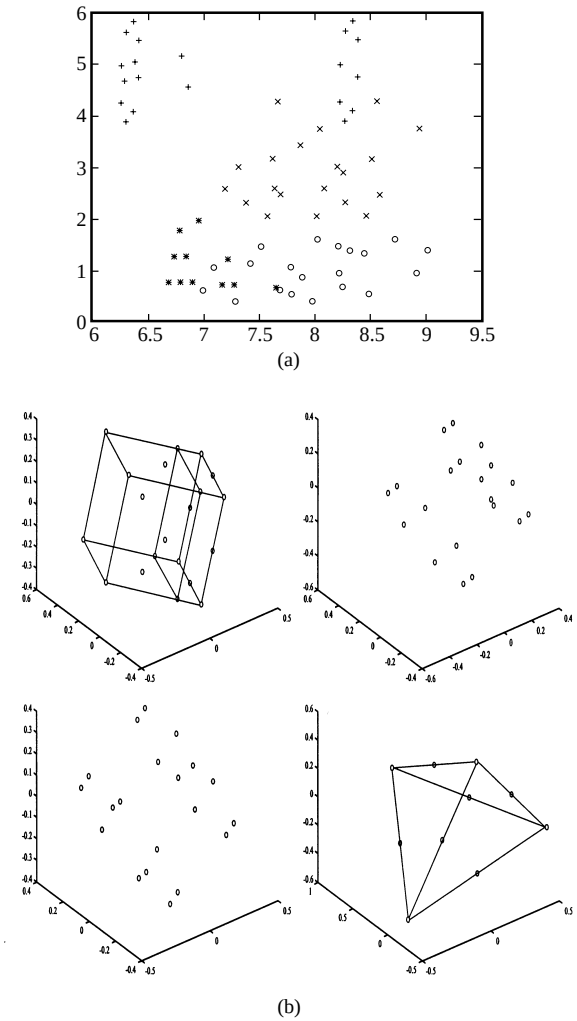

Fig. 10. A multiple target experiment. (a) The feature points of four targets, and (b) the reconstructed 3-D shapes.

conventional clustering algorithms. The rank property can be used to estimate the number of objects. As shown in Fig. 11, the singular values of the measurement matrix indicate a substantial drop in the 17th singular value. Therefore, the rank is 16, and the number of objects is 4, just as predicted. (More details can be found in Kung *et al*. (1994).) By using the subspace clustering method, we obtained four different groups of column vectors $[\boldsymbol{V}_a \boldsymbol{V}_b \boldsymbol{V}_c \boldsymbol{V}_d]$. The translation-rotation decomposition was used to obtain the shapes, which are shown in Fig. 10(b).

Example IV.2 (Two-Camera-Two-Target Case). Experiments were performed on a 2-camera-2-targets motion-shape problem. Two objects (one cube and one pyramid) were used. The SVD approach was successfully applied to separate the two targets and reconstruct their shapes. Figure 12(a) shows the frames taken with the two different cameras. Their intersection vector $\boldsymbol{l}$ is (–1, –2, 0). Figure 12(b) depicts the reconstructed objects obtained following fusion of the image sequences obtained using both cameras.

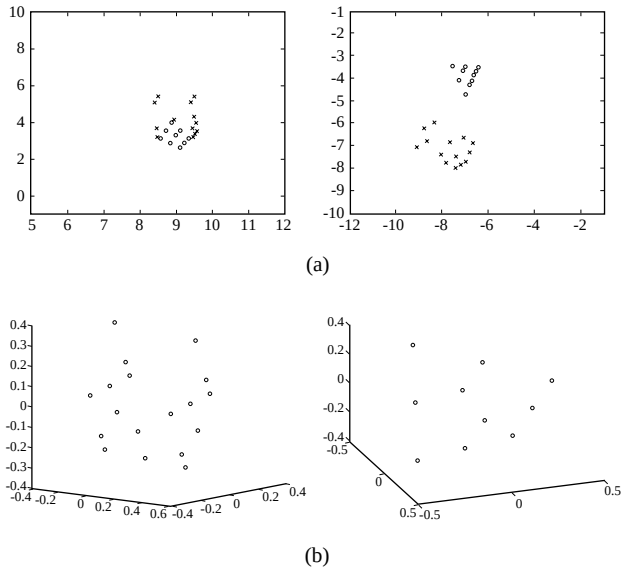

Fig. 12. A two-camera-two-target experiment. (a) The feature points obtained using two cameras, and (b) the reconstructed 3-D shapes.

We observed that the 2-camera case was more sensitive to noise than the 1-camera case. We adopted a smaller singular value threshold to find major basis members and adopted a much larger threshold to find new (minor) members. Note also that when the rotation axis of the object constantly coincided with the intersection vector of the two camera planes, then 2-camera fusion was insufficient to reconstruct the shape. (This is a minor concern since this is a very degenerate situation.)

Example IV.3. The results of the experiment on two moving objects in the sequence LAB2OBJ are shown in Fig. 13. We can observe that the basic relationships among the feature positions have been extracted.

4. Numerical Considerations

To improve the numerical behavior, the basis vectors should be numerically as nonsingular as possible. Here the "numerical nonsingularity" is measured using the smallest singular value associated with the basis vectors. This results in a more stable linear dependency check.

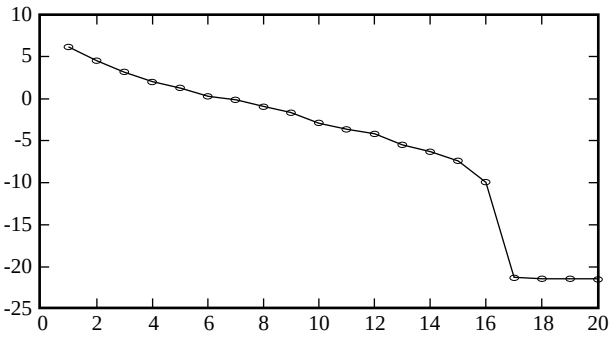

Fig. 11. The semi-log plot of the singular values in the simulation.

A Confidence Measure. In a noisy situation, a confidence measure for linear dependency may be very useful. The membership check should take into account the confidence measure. In a practical, noisy situation, it is more meaningful to ask: "*Is there an approximate linear dependency, and if so, how close is it*?" The answer is complex and hinges upon the confidence criterion adopted. One popular approach is to have $\boldsymbol{A}$ perturbed by a perturbation matrix $\boldsymbol{\Delta}$ so that linear dependency will exist. Suppose that the SVD of $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \Sigma_{i=1}^{m+1} \boldsymbol{u}_i \sigma_i \boldsymbol{\upsilon}_i^T$. Then, by setting $\boldsymbol{\Delta} = -\boldsymbol{u}_{m+1}\sigma_{m+1}\boldsymbol{\upsilon}_{m+1}^T$, we need

$$[\boldsymbol{A} + \boldsymbol{\Delta}] = \sum_{i=1}^{m} u_i \sigma_i \upsilon_i^T \qquad \textbf{(24)}$$

to have rank deficiency. This further implies that $[\boldsymbol{A} + \boldsymbol{\Delta}]$ is the closest approximation of $\boldsymbol{A}$ with rank no more than m. From Eq. (24), we note that $[\boldsymbol{A} + \boldsymbol{\Delta}]\boldsymbol{\upsilon}_{m+1} = 0$, so the "best" normalized null-space solution is simply $\boldsymbol{x} = \boldsymbol{\upsilon}_{m+1}$. In summary:

(1) The last singular vector of $\boldsymbol{A}$, $\boldsymbol{\upsilon}_{m+1}$, reveals the most likely linear dependency existing in $\boldsymbol{A}$.

(2) The last singular value σ_{m+1} gives a quantitative measure of the confidence of such a linear dependency. (The smaller σ_{m+1} is, the higher is the confidence since it is closer to linear dependency.)

It is not necessary to identify all the objects at one time; they may be identified sequentially. This is impor-

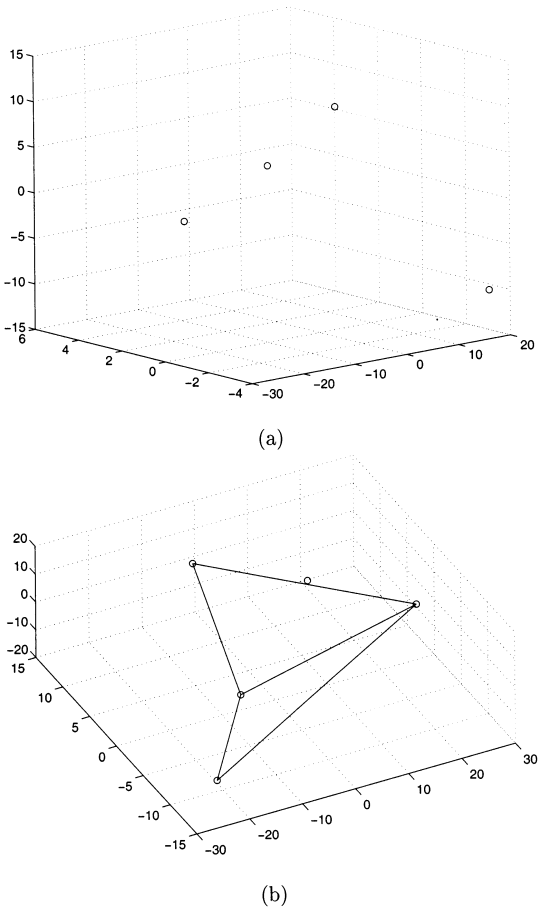

Fig. 13. The shape information of the objects.

tant since the smallest singular values associated with the basis vectors usually decrease (rapidly) with an increasing number of objects (or basis vectors). Therefore, when the number of objects is very large, it may be difficult to form a complete set of basis vectors with a decent smallest singular value. It is then advisable to use only a partial basis set which offers a better and more comfortable "numerical nonsingularity." As long as the partial basis set contains the r basis vectors needed for at least one object, all the (minor) members of that object may be identified afterwards. The members of the first object may be removed from the set $\boldsymbol{V}$ before the search process for the second object is started.

A. Split-and-Merge Procedure for Noisy Data

When the noise level is high (or the tracking is not accurate), it is not easy to determine the number of objects from the rank property. In Fig. 14, the singular values of a set of feature points from four different objects are depicted. From the rank property, $\boldsymbol{W}$ should be a matrix of rank 12. However, this is not obvious in the figure. Moreover, it is sometimes difficult to cluster the feature points into a correct group in one shot. In this case, a split-and-merge procedure can be applied. In this method, perhaps only some of the FPs of the objects are separated from the set of points in each subspace clustering iteration under a certain confidence measure. After the splitting process is completed, we try to merge the detected groups gradually. For each possible combination of two groups of FPs, the confidence measure of the combined group is computed. If the best confidence measure is larger than a threshold, the corresponding groups are merged. The same procedure is repeated unitl the best confidence measure is smaller than the threshold.

B. Principal Singular Vector (PSV) Analysis

In this section, we will describe how principal singular vector analysis can be used to separate feature points objects (Kung *et al*., 1996). It is found that the PSV's have a very good noise-tolerance property.

The **PSV Clustering Method** consists of the following main steps:

(1) The first k PSV's from $\boldsymbol{V}$ form a $k \times P$ matrix, where k is the number of objects and P is the number of feature points.
(2) Cluster the columns into k different groups.
(3) For each group of feature points, compute the motion and shape information using the algorithm described in Section II.2.

This procedure is based on the fact that the translation components of the objects often dominate the rotational component when the SVD of $\boldsymbol{W}$ in Eq. (18) is computed. If the translational components of each object are quite different from each other, the feature points can be separated by using clustering algorithms. One example of

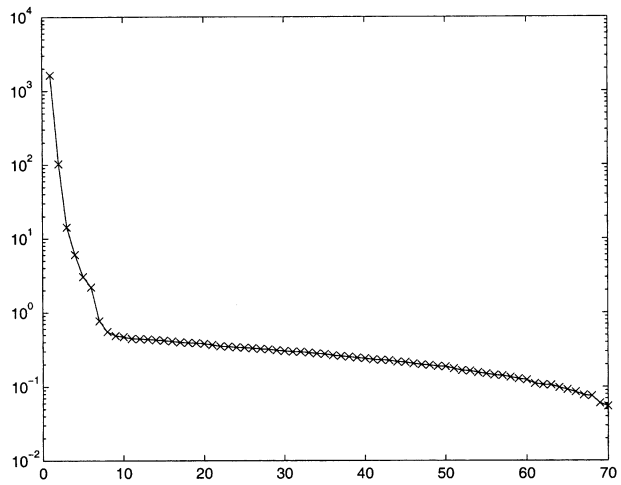

Fig. 14. The semi-log plot of the singular values of a noisy feature point.

the case is shown in Fig. 15 for a set of feature points from three different objects. We can observe that the clusters are far away from each other, and that noise, inaccuracy of the tracker, or the perspective distortion hardly affect the clustering. However, the assumption of dominance is sometimes not correct. For example, the tracking results of the sequence LAB2OBJ introduced in the previous section are shown in Fig. 16. Correct clustering is not guaranteed. Also, the number of objects is sometimes difficult to obtain in noisy environments.

Combined Procedure. The noise tolerance in the split-and-merge approach is not as good as that in the PSV approach especially when the number of objects is large. However if the dominance property does not hold, the PSV approach can not be applied. Therefore a combination approach offers the advantages of both methods. First, the PSV approach can be adopted to separate as many clusters as possible. Then, the split-and-merge method can be used to cluster the rest of the feature points. Since the number of objects is smaller, the split-merge method can achieve better performance.

V. Conclusion

Robust selection and tracking of feature points is a crucial preprocessing step. However, the tracking results are often influenced by the following factors: the focus of the camera, the type and speed of motion, the background image, and distortion in perspective. Thus, it is very difficult to design a universal tracker which can work well in all kinds of environments. In this paper, a weighted block matching approach has been proposed to improve the robustness of block matching techniques used for tracking. Weighting masks are adopted to emphasize reliable region in the matching procedure. We have described the work done by Tomasi and Kanade (1992), who focused on the single target MSE problem. Then, we extended the formulation to cope with the multi-camera-multi-target situation. A combination procedure based on principal singular vector analysis and split-and-merge subspace clustering algorithms has been proposed to solve this problem. This research can serve as a basis for many potential applications, such as surveillance applications. Our approach can also be used to separate and then recognize different moving parts, which may be useful in the coding of video sequences.

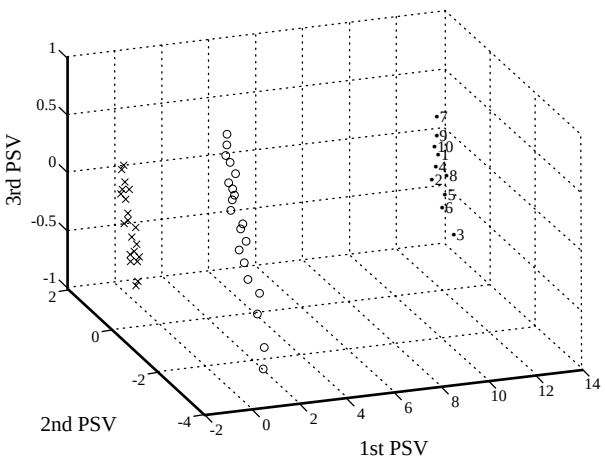

Fig. 15. This figure shows that PSV's can be used to separate objects. In the figure, the 'O's, 'X's, and numerals indicate different objects.

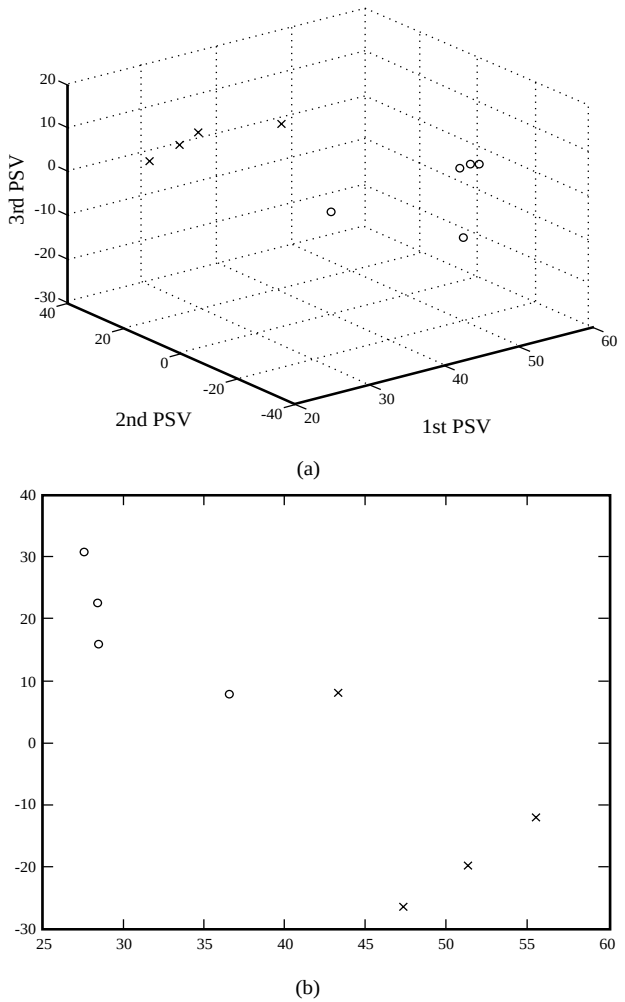

Fig. 16. This figure shows that clusters sometimes are not so obviously separable. In (a), three PSV's are shown. The first two PSV's are plotted in (b).

References

Kitchen, L. and A. Rosenfeld (1982) Gray-level corner detection. *Pattern*

Recognition Letters, **1**, 95-102.

Kung, S. Y. and K. I. Diamantaras (1990) A neural network learning algorithm for Adaptive Principal Component EXtraction (APEX). *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 861-864, Albuquerque, NM, U.S.A.

Kung, S. Y., Y. T. Lin, and Y. K. Chen (1996) Motion-based segmentation by principal singular vectors (PSV) clustering method. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 3410-3413, Atlanta, GA, U.S.A.

Kung, S. Y., J. S. Taur, and M. Y. Chiu (1994) Application of SVD networks to multi-object motion-shape analysis. In: *Neural Networks for Signal Processing, IV, Proceedings of IEEE Workshop*, pp. 413-422, Ermioni, Greece.

Marugame, A., J. Katto, and M. Ohta (1999) Structure recovery with multiple cameras from scaled orthographic and perspective views. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **21**(7), 628-633.

Mokhtarian, F. and R. Suomela (1998) Robust image corner detection through curvature scale space. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **20**(12), 1376-1381.

Moravec, H. P. (1981) Rover visual obstacle avoidance. *Proc. of the 7th International Conference on Artificial Intelligence*, pp. 785-790, Vancouver, BC, Canada.

Morita, T. and T. Kanade (1997) A sequential factorization method for recovering shape and motion from image streams. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **19**(8), 858-867.

Poelman, C. J. and T. Kanade (1997) A paraperspective factorization method for shape and motion recovery. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **19**(3), 206-218.

Taur, J. S. (1995) Feature selection and tracking for video signal. *Proceeding, Workshop on Computer Application*, pp. 53-58, Nantao, Taiwan, R.O.C.

Thorpe, C. E. (1984) *Fido: Vision and Navigation for a Robot Rover*. Ph.D. Dissertation (CMU-CS-84-168). CMU, Pittsburgh, PA, U.S.A.

Tomasi, C. and T. Kanade (1991) *Shape and Motion from Image Streams: a Factorization Method – Part 3. Detection and Tracking of Points Features*. Technical Report CMU-CS-91-132, CMU, Pittsburgh, PA, U.S.A.

Tomasi, C. and T. Kanade (1992) Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, **9**(2), 330-334.

Ullman, S. (1979) *The Interpretation of Visual Motion*. MIT Press, Cambridge, MA, U.S.A.

時序影像特徵點追蹤與多目標之行動形體分析

陶金旭

國立中興大學電機工程學系

摘　要

三維形體重建在影像分析及物體辨認的研究領域中是一重要的課題。Kanade 提出一個從視訊信號分析出單一物體形體的演算法並受到相當的重視。此一演算法主要是以奇異值分析法（Singular Value Decomposition）爲基礎。在本論文中，我們主要研究多目標及多攝影機的情形。也就是說，給定一系列內含多個移動物體的二維視訊影像，目的爲計算各物體的三維移動資訊以及他們的三維形狀。同時也討論在多個攝影機時的情況。首先，我們設計一特徵點追蹤的演算法來找出特徵點在一連串的影像中相對應的座標位置。此一演算法可加強區域比對的可靠度。接著使用主要奇異向量分析法（Principal Singular Vector Analysis）以及次空間分類法（Subspace Clustering Method）將特徵點分類到相對應的物體上。然後各物體的形體及運動軌跡將可由矩陣分解法加以求出。在此論文中，我們將列出人工合成的資料以及由攝影機取得的實際影像的實驗結果。