PAPER

Multi-Stage Unsupervised Learning for Multi-Body Motion Segmentation

Yasuyuki SUGAYA[†] ***and*** **Kenichi KANATANI**[†], ***Members***

SUMMARY Many techniques have been proposed for segmenting feature point trajectories tracked through a video sequence into independent motions, but objects in the scene are usually assumed to undergo general 3-D motions. As a result, the segmentation accuracy considerably deteriorates in realistic video sequences in which object motions are nearly degenerate. In this paper, we propose a multi-stage unsupervised learning scheme first assuming degenerate motions and then assuming general 3-D motions and show by simulated and real video experiments that the segmentation accuracy significantly improves without compromising the accuracy for general 3-D motions.

key words: *motion segmentation, unsupervised learning, EM algorithm, affine camera model, degenerate motion*

1. Introduction

Segmenting feature point trajectories tracked through a video sequence into independent motions is the first step of many video processing applications. Already, many techniques have been proposed for this task.

Costeira and Kanade [1] proposed a segmentation algorithm based on the shape interaction matrix. Gear [3] used the reduced row echelon form and graph matching. Ichimura [4] used the discrimination criterion of Otsu [12]. He also used the QR decomposition [5]. Inoue and Urahama [6] introduced fuzzy clustering. Kanatani [8]–[10] incorporated model selection using the geometric AIC [7]. Wu et al. [19] introduced orthogonal subspace decomposition.

However, all these methods assume that objects in the scene undergo general 3-D motions relative to the camera. As a result, segmentation fails when the motions are degenerate, e.g., all the objects are simply translating independently. This type of degeneracy frequently occurs in real scenes.

At first sight, segmenting simple motions may seem easier than segmenting complicated motions. In reality, however, the opposite is the case, because complicated motions have sufficient cues for mutual discrimination. In fact, many methods that exhibit high accuracy for complicated simulations perform very poorly for real video sequences. To cope with this, we introduced a scheme for automatically selecting the best motion model using the geometric AIC [15], [16], but the improvement was very much limited.

In this paper, we introduce unsupervised learning [14] assuming degenerate motions followed by unsupervised learning assuming general 3-D motions and show that the segmentation accuracy significantly improves without compromising the accuracy for general 3-D motions.

In Sec. 2, we describe the geometric constraints that underlie our method. In Sec. 3, we introduce unsupervised learning of the non-Bayesian and Bayesian types. Our multi-stage learning scheme is described in Sec. 4. In Sec. 5, we show synthetic and real video examples. Section 6 concludes this paper.

2. Geometric Constraints

2.1 Trajectory of Feature Points

Suppose we track N feature points over M frames. Let $(x_{\kappa\alpha}, y_{\kappa\alpha})$ be the coordinates of the αth point in the κth frame. Stacking all the coordinates vertically, we represent the entire trajectory by the following $2M$-D *trajectory vector*:

$$\boldsymbol{p}_\alpha = (x_{1\alpha}\ y_{1\alpha}\ x_{2\alpha}\ y_{2\alpha} \cdots\ x_{M\alpha}\ y_{M\alpha})^\top. \qquad (1)$$

For convenience, we identify the frame number κ with "time" and refer to the κth frame as "time κ".

We regard the XYZ camera coordinate system as the world frame, relative to which multiple objects are moving. Consider a 3-D coordinate system fixed to one moving object, and let $\boldsymbol{t}_\kappa$ and $\{\boldsymbol{i}_\kappa, \boldsymbol{j}_\kappa, \boldsymbol{k}_\kappa\}$ be, respectively, its origin and basis vectors at time κ. If the αth point has coordinates $(a_\alpha, b_\alpha, c_\alpha)$ with respect to this coordinate system, its position with respect to the world frame at time κ is

$$\boldsymbol{r}_{\kappa\alpha} = \boldsymbol{t}_\kappa + a_\alpha \boldsymbol{i}_\kappa + b_\alpha \boldsymbol{j}_\kappa + c_\alpha \boldsymbol{k}_\kappa. \qquad (2)$$

2.2 Affine Camera Model

We assume an affine camera, which generalizes orthographic, weak perspective, and paraperspective projections [13]: the 3-D point $\boldsymbol{r}_{\kappa\alpha}$ is projected onto the image position

$$\begin{pmatrix} x_{\kappa\alpha} \\ y_{\kappa\alpha} \end{pmatrix} = \boldsymbol{A}_\kappa \boldsymbol{r}_{\kappa\alpha} + \boldsymbol{b}_\kappa, \qquad (3)$$

where $\boldsymbol{A}_\kappa$ and $\boldsymbol{b}_\kappa$ are, respectively, a 2×3 matrix and a 2-D vector determined by the position and orientation of the camera and its internal parameters at time κ. Substituting Eq. (2), we have

Manuscript received September 5, 2003.
Manuscript revised February 4, 2003.

[†]The authors are with the Department of Information Technology, Okayama University, Okayama-shi 700-8530 Japan.
a) E-mail: sugaya@suri.it.okayama-u.ac.jp

$$\begin{pmatrix} x_{\kappa\alpha} \\ y_{\kappa\alpha} \end{pmatrix} = \tilde{\boldsymbol{m}}_{0\kappa} + a_\alpha \tilde{\boldsymbol{m}}_{1\kappa} + b_\alpha \tilde{\boldsymbol{m}}_{2\kappa} + c_\alpha \tilde{\boldsymbol{m}}_{3\kappa}, \quad (4)$$

where $\tilde{\boldsymbol{m}}_{0\kappa}$, $\tilde{\boldsymbol{m}}_{1\kappa}$, $\tilde{\boldsymbol{m}}_{2\kappa}$, and $\tilde{\boldsymbol{m}}_{3\kappa}$ are 2-D vectors determined by the position and orientation of the camera and its internal parameters at time κ. From Eq. (4), the trajectory vector $\boldsymbol{p}_\alpha$ in Eq. (1) can be written in the form

$$\boldsymbol{p}_\alpha = \boldsymbol{m}_0 + a_\alpha \boldsymbol{m}_1 + b_\alpha \boldsymbol{m}_2 + c_\alpha \boldsymbol{m}_3, \quad (5)$$

where $\boldsymbol{m}_0$, $\boldsymbol{m}_1$, $\boldsymbol{m}_2$, and $\boldsymbol{m}_3$ are the $2M$-D vectors obtained by stacking $\tilde{\boldsymbol{m}}_{0\kappa}$, $\tilde{\boldsymbol{m}}_{1\kappa}$, $\tilde{\boldsymbol{m}}_{2\kappa}$, and $\tilde{\boldsymbol{m}}_{3\kappa}$ vertically over the M frames, respectively.

2.3 Constraints on Image Motion

Equation (5) implies that the trajectories of the feature points that belong to one object are constrained to be in the *4-D subspace* spanned by $\{\boldsymbol{m}_0, \boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3\}$ in $\mathcal{R}^{2M}$. It follows that multiple moving objects can be segmented into individual motions by separating the trajectories vectors $\{\boldsymbol{p}_\alpha\}$ into distinct 4-D subspaces. This is the principle of the method of *subspace separation* [8], [9].

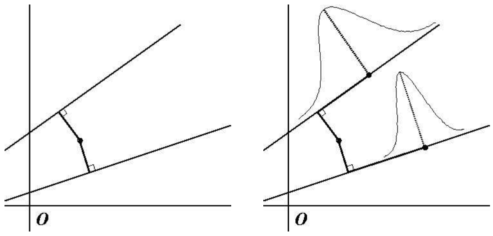
In addition, the coefficient of $\boldsymbol{m}_0$ in Eq. (5) is identically 1 for all α. This means that the trajectories are in a *3-D affine space* within that 4-D subspace[†]. It follows that multiple moving objects can be segmented into individual motions by separating the trajectory vectors $\{\boldsymbol{p}_\alpha\}$ into distinct 3-D affine spaces. This is the principle of the method of *affine space separation* [10].

Theoretically, the segmentation accuracy should be higher if we use stronger constraints. According to simulations, the affine space separation indeed performs better than the subspace separation except in the case in which perspective effects are very strong and the noise is small [10]. For real video sequences, however, the affine space separation accuracy is often lower than that of the subspace separation [15], [16]. The cause of this inconsistency will be clarified in the subsequent analysis.

2.4 Number of Motions

We assume that the number m of motions is specified by the user. For example, if a single object is moving in a static background, both moving relative to the camera, we have $m = 2$.

Many studies have been done for estimating the number of motions automatically [1], [3], [6], but none of them seems successful enough. This is because the number of motions is *not well-defined* [9], [11]: one moving object can also be viewed as multiple objects moving similarly, and there is no rational way to unify similarly moving objects into one except using heuristic thresholds or ad-hoc criteria. If we use model selection, for example, the resulting number of motions depends on criteria[‡] such as the geometric AIC and the geometric MDL [9], [11]. We conclude that the number m of motions should be input by the user and the unification process should be left to each application.



(a) (b)

Fig. 1 Segmentation criterion: (a) non-Bayesian type; (b) Bayesian type.

2.5 Outlier Removal

The feature point trajectories tracked through video frames are not necessarily correct, so we need to remove outliers. If the trajectories were segmented into individual classes, we could remove, for example, those that do not fit to the individual affine spaces. In the presence of outliers, however, we cannot do correct segmentation, and hence we do not know the affine spaces.

This difficulty can be resolved if we note that if the trajectory vectors $\{\boldsymbol{p}_\alpha\}$ belong to m d-D subspaces, they should be constrained to be in a dm-D subspace and if they belong to m d-D affine spaces, they should be in a $((d+1)m-1)$-D affine space. So, we robustly fit a dm-D subspace or a $((d+1)m-1)$-D affine space to $\{\boldsymbol{p}_\alpha\}$ by RANSAC and remove those that do not fit to it [17]. Thus, outliers can be removed *without knowing the segmentation results*.

Theoretically, the resulting trajectories may not necessarily be all correct. However, we observed that all apparent outliers were removed by this method[†††], although some inliers were also removed for safety [17].

3. Unsupervised Learning

3.1 Non-Bayesian Type

Segmentation by the subspace or affine space separation is not always correct. Here, we consider optimizing the segmentation *a posteriori* by fitting a 3-D affine space (or a 4-D subspace) to each class and reclassifying each trajectory to the closest affine space (or subspace) (Fig. 1(a)). This process is iterated until the classification converges.

If the noise in the coordinates of the feature points is an independent Gaussian random variable of mean 0

[†] Customarily, $\boldsymbol{m}_0$ is identified with the centroid of $\{\boldsymbol{p}_\alpha\}$, and Eq. (5) is written as $\begin{pmatrix} \boldsymbol{p}'_1 & \cdots & \boldsymbol{p}'_N \end{pmatrix} = \begin{pmatrix} \boldsymbol{m}_1 & \boldsymbol{m}_2 & \boldsymbol{m}_3 \end{pmatrix} \begin{pmatrix} a_1 & \cdots & a_N \\ b_1 & \cdots & b_N \\ c_1 & \cdots & c_N \end{pmatrix}$ or $\boldsymbol{W} = \boldsymbol{M}\boldsymbol{S}$, where $\boldsymbol{p}'_\alpha = \boldsymbol{p}_\alpha - \boldsymbol{m}_0$. However, our formulation is more convenient for the subsequent analysis.

[††] The program is available at:
http://www.suri.it.okayama-u.ac.jp/e-program.html

[†††] The program is available at:
http://www.suri.it.okayama-u.ac.jp/e-program.html

and a constant variance, this procedure can be viewed as unsupervised learning based on maximum likelihood estimation, since minimizing the distance of points from the fitted space is equivalent to maximizing their likelihood under our noise model.

3.2 Bayesian Type

We may also model the data distributions inside the fitted spaces (Fig. 1(b)). This is the standard approach to unsupervised learning for pattern recognition. However, the existence of geometric constraints complicates the likelihood computation. For the affine space constraint, the actual procedure becomes as follows (the procedure for the subspaces constraint goes similarly).

Let $n = 2M$. Suppose N n-D trajectory vectors $\{\boldsymbol{p}_\alpha\}$ are already segmented into m classes by some means. Initially, we define the weight $W_\alpha^{(k)}$ of the vector $\boldsymbol{p}_\alpha$ by

$$W_\alpha^{(k)} = \left\{ \begin{array}{ll} 1 & \text{if } \boldsymbol{p}_\alpha \text{ belongs to class } k \\ 0 & \text{otherwise} \end{array} \right. . \tag{6}$$

Then, we iterate the following procedures A and B in turn until all the weights $\{W_\alpha^{(k)}\}$ converge[†].

A. Do the following computation for each class $k = 1$, ..., m.

1. Compute the fractional size of the class k

$$w^{(k)} = \frac{1}{N}\sum_{\alpha=1}^{N} W_\alpha^{(k)}. \tag{7}$$

2. Compute the centroid $\boldsymbol{p}_C^{(k)}$ of the class k:

$$\boldsymbol{p}_C^{(k)} = \frac{\sum_{\alpha=1}^{N} W_\alpha^{(k)}\boldsymbol{p}_\alpha}{\sum_{\alpha=1}^{N} W_\alpha^{(k)}}. \tag{8}$$

3. Compute the $n \times n$ moment matrix of the class k:

$$\boldsymbol{M}^{(k)} = \frac{\sum_{\alpha=1}^{N} W_\alpha^{(k)}(\boldsymbol{p}_\alpha - \boldsymbol{p}_C^{(k)})(\boldsymbol{p}_\alpha - \boldsymbol{p}_C^{(k)})^\top}{\sum_{\alpha=1}^{N} W_\alpha^{(k)}}. \tag{9}$$

4. Let $\lambda_1 \geq \lambda_2 \geq \lambda_3$ be the largest three eigenvalues of the matrix $\boldsymbol{M}^{(k)}$, and $\boldsymbol{u}_1^{(k)}$, $\boldsymbol{u}_2^{(k)}$, and $\boldsymbol{u}_3^{(k)}$ the corresponding unit eigenvectors.
5. Compute the $n \times n$ projection matrices

$$\boldsymbol{P}^{(k)} = \sum_{i=1}^{3}\boldsymbol{u}_i^{(k)}\boldsymbol{u}_i^{(k)\top}, \quad \boldsymbol{P}_\perp^{(k)} = \boldsymbol{I} - \boldsymbol{P}^{(k)}, \tag{10}$$

where $\boldsymbol{I}$ denotes the $n \times n$ unit matrix.

6. Estimate the noise variance in the direction orthogonal to the kth affine space by

$$\hat{\sigma}_k^2 = \max\left[\frac{\mathrm{tr}[\boldsymbol{P}_\perp^{(k)}\boldsymbol{M}^{(k)}\boldsymbol{P}_\perp^{(k)}]}{n-3}, \sigma^2\right], \tag{11}$$

where $\mathrm{tr}[\,\cdot\,]$ denotes the trace and σ is an estimate of the tracking accuracy[‡].

7. Compute the $n \times n$ covariance matrix of the class k by

$$\boldsymbol{V}^{(k)} = \boldsymbol{P}^{(k)}\boldsymbol{M}^{(k)}\boldsymbol{P}^{(k)} + \hat{\sigma}_k^2\boldsymbol{P}_\perp^{(k)}. \tag{12}$$

B. Do the following computation for each trajectory vector $\boldsymbol{p}_\alpha$, $\alpha = 1$, ..., N.

1. Compute the conditional likelihood $P(\alpha|k)$, $k = 1$, ..., m, by

$$P(\alpha|k) = \frac{e^{-(\boldsymbol{p}_\alpha - \boldsymbol{p}_C^{(k)}, \boldsymbol{V}^{(k)-1}(\boldsymbol{p}_\alpha - \boldsymbol{p}_C^{(k)}))/2}}{\sqrt{\det \boldsymbol{V}^{(k)}}}. \tag{13}$$

2. Recompute the weights $W_\alpha^{(k)}$, $k = 1$, ..., m, by

$$W_\alpha^{(k)} = \frac{w^{(k)}P(\alpha|k)}{\sum_{l=1}^{m} w^{(l)}P(\alpha|l)}. \tag{14}$$

After the iterations of A and B have converged, $\boldsymbol{p}_\alpha$ is classified into the class k that maximizes $W_\alpha^{(k)}$, $k = 1$, ..., m.

3.3 Interpretation

In the above iterations, we fit a Gaussian distribution of mean $\boldsymbol{p}_C^{(k)}$ (Eq. (8)) and the rank 3 covariance matrix $\boldsymbol{P}^{(k)}\boldsymbol{M}^{(k)}\boldsymbol{P}^{(k)}$ (Eqs. (9), (10)) to the data distribution inside each 3-D affine space. For the outside deviations, we fit a Gaussian distribution of mean 0 and a constant variance $\hat{\sigma}_k^2$ (Eq. (11)).

Using this probabilistic interpretation, we compute the probability $P(\alpha|k)$ of the trajectory vector $\boldsymbol{p}_\alpha$ conditioned to be in the class k (Eq. (13)). Since the fraction $w^{(k)}$ (Eq. (7)) can be interpreted to be the *a priori probability* of the class k, we apply *Bayes' theorem* (Eq. (14)) to compute the *a posterior probability* $W_\alpha^{(k)}$, according which all the trajectories are reclassified. Note that $W_\alpha^{(k)}$ is generally a fraction, so one trajectory belongs to multiple classes with fractional weights until the final classification is made. If we consider only the outside deviations, the above procedure reduces to the non-Bayesian type.

This type of learning[†††] is widely used for pattern recognition, and the likelihood is known to increase monotonously by iterations [14]. It is also well known, however, that the iterations are very likely to be trapped at a local maximum. So, correct segmentation cannot be obtained by this type of iterations alone unless we start from a very good initial value.

4. Multi-Stage Learning

Now, we model degenerate motions and derive an associated learning procedure, from which we construct our multi-stage learning procedure.

[†] We stopped the iterations when the increments in $W_\alpha^{(k)}$ are all smaller than 10^{-10}.

[††] We found $\sigma = 0.5$ (pixels) a reasonable value [17].

[†††] This scheme is often referred to as the *EM algorithm* [2], because the mathematical structure is the same as estimating parameters from "incomplete data" by maximizing the logarithmic likelihood marginalized by the posterior of the missing data specified by Bayes' theorem.

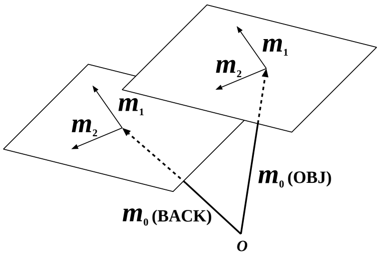


Fig. 2 If the motions of the objects and the background are degenerate, their trajectory vectors belong to mutually parallel 2-D affine spaces.

4.1 Degenerate Motions

The motions we most frequently encounter are such that the objects and the background are translating and rotating 2-dimensionally in the image frame with varying sizes.

For such a motion, we can choose the basis vector $\boldsymbol{k}_\kappa$ in Eq. (2) in the Z direction (the camera optical axis is identified with the Z-axis). Under the affine camera model, motions in the Z direction do not affect the projected image except for its size. Hence, the vector $\tilde{\boldsymbol{m}}_{3\kappa}$ in Eq. (4) can be taken to be $\mathbf{0}$; the scale changes of the projected image are absorbed by the scale changes of $\tilde{\boldsymbol{m}}_{1\kappa}$ and $\tilde{\boldsymbol{m}}_{2\kappa}$ over time κ. It follows that the trajectory vector $\boldsymbol{p}_\alpha$ in Eq. (5) belongs to the 2-D affine space passing through $\boldsymbol{m}_0$ and spanned by $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ [15], [16].

All existing segmentation methods based on the shape interaction matrix of Costeira and Kanade [1] assume that the trajectories of different motions belong to independent 3-D subspaces [8], [9]. Hence, degenerate motions cannot be correctly segmented.

If, in addition, the objects and the background do not rotate, we can fix the basis vectors $\boldsymbol{i}_\kappa$ and $\boldsymbol{j}_\kappa$ in Eq. (2) to be in the X and Y directions, respectively. Thus, the basis vectors $\boldsymbol{i}_\kappa$ and $\boldsymbol{j}_\kappa$ are common to the objects and the background, so the vectors $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ in Eq. (5) are also common. Hence, the 2-D affine spaces of all the motions are *parallel* (Fig. 2).

Note that *parallel 2-D affine spaces can be included in a 3-D affine space*. Since the affine space separation method attempts to segment the trajectories into different 3-D affine spaces, it does not work if the objects and the background undergo this type of degenerate motions. This explains why the accuracy of the affine space separation is not as high as expected for real video sequences.

4.2 Unsupervised Learning for Degenerate Motions

Since many motions we encounter in practice are degenerate, we can expect that the segmentation accuracy increases by learning based on such degenerate motions.

This is done as follows. Initializing the weight $W_\alpha^{(k)}$ by Eq. (6), we iterate the following procedures A, B, and C in turn until all $\{W_\alpha^{(k)}\}$ converge†:

A. Do the following computation for each class $k = 1, ..., m$.

1. Compute the fraction $w^{(k)}$ by Eq. (7).
2. Compute the centroid $\boldsymbol{p}_C^{(k)}$ of the class k by Eq. (8).
3. Compute the $n \times n$ moment matrix $\boldsymbol{M}^{(k)}$ by Eq. (9).

B. Do the following computation.

1. Compute the total $n \times n$ moment matrix
$$\boldsymbol{M} = \sum_{k=1}^{m} w^{(k)} \boldsymbol{M}^{(k)}. \qquad (15)$$
2. Let $\lambda_1 \geq \lambda_2$ be the largest two eigenvalues of the matrix $\boldsymbol{M}$, and $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ the corresponding unit eigenvectors.
3. Compute the $n \times n$ projection matrices
$$\boldsymbol{P} = \sum_{i=1}^{2} \boldsymbol{u}_i \boldsymbol{u}_i^\top, \qquad \boldsymbol{P}_\perp = \boldsymbol{I} - \boldsymbol{P}. \qquad (16)$$
4. Estimate the noise variance in the direction orthogonal to all the affine spaces by
$$\hat{\sigma}^2 = \max[\frac{\mathrm{tr}[\boldsymbol{P}_\perp \boldsymbol{M} \boldsymbol{P}_\perp]}{n-2}, \sigma^2]. \qquad (17)$$
5. Compute the $n \times n$ covariance matrix of the class k by
$$\boldsymbol{V}^{(k)} = \boldsymbol{P}\boldsymbol{M}^{(k)}\boldsymbol{P} + \hat{\sigma}^2 \boldsymbol{P}_\perp. \qquad (18)$$

C. Do the following computation for each trajectory vector $\boldsymbol{p}_\alpha$, $\alpha = 1, ..., N$.

1. Compute the conditional likelihood $P(\alpha|k)$, $k = 1, ..., m$, by Eq. (13).
2. Recompute the weights $\{W_\alpha^{(k)}\}$, $k = 1, ..., m$, by Eq. (14).

The computation is the same as in Sec. 3.2 except that 2-D affine spaces with the same orientation are fitted. The common basis vectors $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ and the common outside noise variance are estimated in the procedure B.

4.3 Multi-Stage Procedure

If we *know* that degeneracy exists, we can apply the above procedure for improving the segmentation. However, we do not know if degeneracy exists. If the trajectories were segmented into individual classes, we might detect degeneracy by checking the dimensions of the individual classes, but we cannot do correct segmentation unless we know whether or not degeneracy exists.

We resolve this difficulty by the following multi-stage learning. First, we use the affine space separation assuming 2-D affine spaces, which effectively assumes planar motions with varying sizes. Then, we optimize the resulting segmentation by using the degenerate model.

The resulting solution should be very accurate if such a degeneracy really exists. However, rotations may exist to some extent. So, we relax the constraint and optimize the solution again, assuming general 3-D

†See the footnote † in Section 3.2.

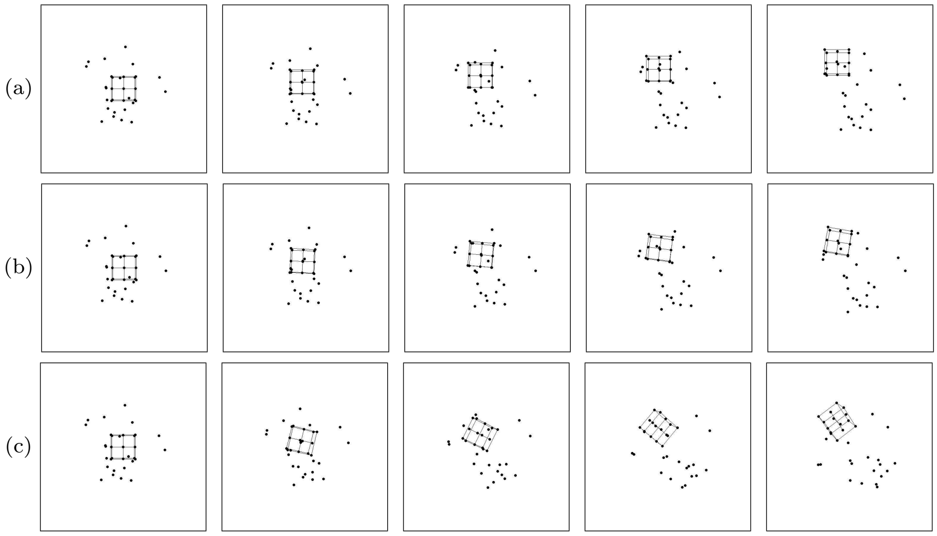
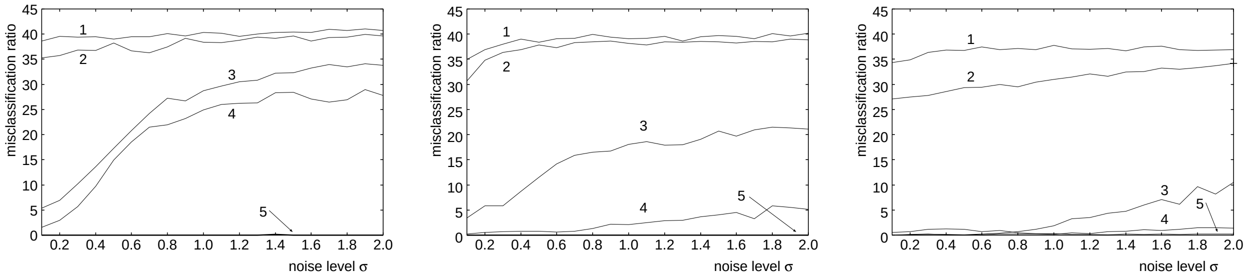


Fig. 3 Simulated image sequences of 14 object points and 20 background points: (a) almost degenerate motion; (b) nearly degenerate motion; (c) general 3-D motion.



(a) (b) (c)

Fig. 4 Misclassification ratio for the sequences (a), (b), and (c) in Fig. 3: 1) Costeira-Kanade; 2) Ichimura; 3) optimized subspace separation; 4) optimized affine space separation; 5) multi-stage learning.

motions. This is motivated by the fact that if the motions are really degenerate, the solution optimized by the degenerate model is *not affected* by the subsequent optimization using the general model, because the degenerate constraints also satisfy the general constraints.

In sum, our scheme consists of the following three stages:

1. Initial segmentation by the affine space separation using 2-D affine spaces.
2. Unsupervised learning of the Bayesian type based on degenerate motions.
3. Unsupervised learning of the Bayesian type based on general 3-D motions.

5. Experiments

5.1 Simulations

Fig. 3 shows three sequences of five synthetic images (supposedly of 512×512 pixels) of 14 object points and 20 background points; the object points are connected by line segments for the ease of visualization. To simulate real circumstances better, all the points are perspectively projected onto each frame with 30° angle of view, although the underlying theory is based on the affine camera model without perspective effects.

In all the three sequences, the object moves toward the viewer in one direction (10° from the image plane), while the background moves away from the viewer in another direction (10° from the image plane). In (a), the object and the background are simply translating in different directions. In (b) and (c), they are additionally rotating by 2° per frame in opposite senses around different axes making 10° from the optical axis in (b) and 60° in (b). Thus, all the three motions are not strictly degenerate (with perspective effects), but the motion is almost degenerate in (a), nearly degenerate in (b), and a general 3-D motion in (c).

Adding independent Gaussian random noise of mean 0 and standard deviation σ to the coordinates of all the points, we segmented them into two groups ($m = 2$). Fig. 4 plots the average misclassification ratio over 500 trials using different noise. We compared 1) the Costeira-Kanade method [1], 2) Ichimura's method [4], 3) the subspace separation [8], [9] followed by unsupervised learning of the Bayesian type (we call this *optimized subspace separation* for short), 4) the affine space separation [10] followed by unsupervised learning

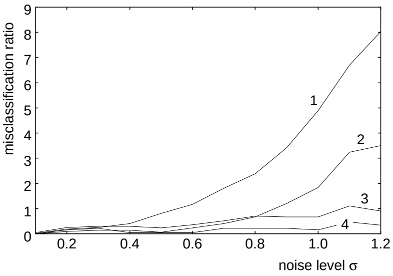


Fig. 5 Comparison of misclassification ratios: 1) subspace separation; 2) subspace separation followed by unsupervised learning of the non-Bayesian type; 3) subspace separation followed by unsupervised learning of the Bayesian type; 4) affine space separation; 5) affine space separation followed by unsupervised learning of the non-Bayesian type 6) affine space separation followed by unsupervised learning of the Bayesian type.

of the Bayesian type (*optimized affine space separation* for short), and 5) our multi-stage learning.

For the almost degenerate motion in Fig. 3(a), the optimized subspace and affine space separations do not work very well. Also, the latter is not superior to the former (Fig. 4(a)). Since our multi-stage learning is based on this type of degeneracy, it achieves 100% accuracy over all the noise range.

For the nearly degenerate motion in Fig. 3(b), the optimized subspace and affine space separations work fairly well (Fig. 4(b)). However, our method still attains almost 100% accuracy.

For the general 3-D motion in Fig. 3(c), the optimized subspace and affine space separations exhibit relatively high performance (Fig. 4(c)), but our method performs much better with nearly 100% accuracy again.

Although the same learning procedure is used in the end, the multi-stage learning performs better than the optimal affine space separation, because the former starts from a better initial value than the latter.

For all the motions, the Costeira-Kanade method performs very poorly. The accuracy is not 100% even in the absence of noise ($\sigma = 0$) because of the perspective effects. Ichimura's method is not effective, either. It works to some extent for the general 3-D motion in Fig. 3(c), but it does not compare with the optimized subspace or affine space separation, much less with our multi-stage optimization method.

5.2 Effects of Learning

Fig. 5 shows the effects of learning for Fig. 3(c). We see that both the non-Bayesian and the Bayesian types work effectively but that the latter is slightly better. However, our multi-stage learning is far better.

Fig. 6 shows the stage-wise effects of our multi-stage learning for Fig. 3(c). For this general 3-D motion, the learning based on degenerate motions does not perform so very well indeed, but the subsequent learning based on general 3-D motions successfully restores the accuracy up to almost 100%.

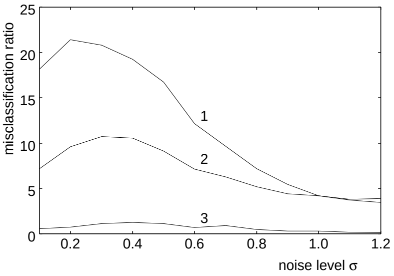


Fig. 6 Stage-wise misclassification ratios of multi-stage learning: 1) affine space separation using 2-D affine spaces; 2) unsupervised learning of the Bayesian type assuming degenerate motions; 3) unsupervised learning of the Bayesian type assuming general 3-D motions.

The interesting fact is that the accuracy increases as the noise increases. This is because the discrepancy between the assumed affine camera model and the actual perspective projection is more conspicuous when the noise is smaller [10].

5.3 Real Video Examples

Fig. 7 shows five decimated frames from three video sequences A, B, and C (320 × 240 pixels). For each sequence, we detected feature points in the initial frame and tracked them using the Kanade-Lucas-Tomasi algorithm [18]. The marks □ indicate their positions.

Table 1 lists the number of frames, the number of inlier trajectories, and the computation time for our multi-stage learning. We reduced the computation time by compressing the trajectory data into 8-D vectors [15]. We used Pentium 4 2.4GHz for the CPU with 1GB main memory and Linux for the OS.

Table 2 lists the accuracies of different methods ("opt" stands for "optimized") measured by (the number of correctly classified points)/(the total number of points) in percentage ($m = 2$). Except for the Costeira-Kanade and Ichimura methods, the percentage is averaged over 50 trials, since the subspace and affine space separations internally use random sampling for robust estimation and hence the result is slightly different for each trial.

As we see, the Costeira-Kanade method fails to produce meaningful segmentation. Ichimura's method is effective for sequences A and B but not so effective for sequence C. For sequence A, the affine space separation is superior to the subspace separation. For sequence B, the two methods have almost the same performance. For sequence C, the subspace separation is superior to the affine space separation, suggesting that the motion in sequence C is nearly degenerate.

The effect of learning is larger for sequence A than for sequences B and C, for which the accuracy is already high before the learning. Thus, the effect of learning very much depends on the quality of the initial segmentation. For all the three sequences, our multi-stage

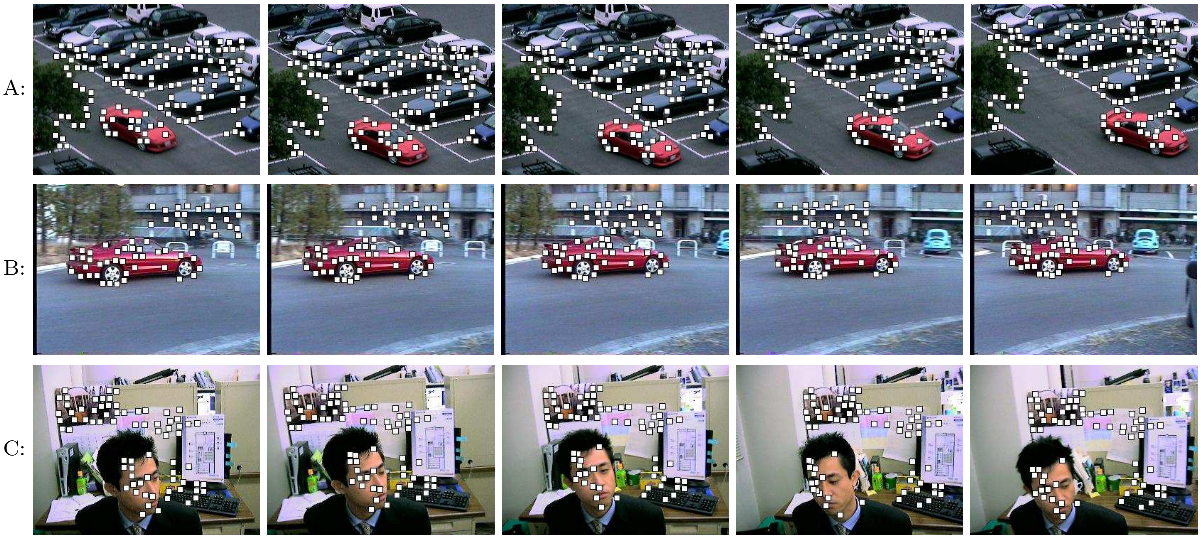


Fig. 7 Three video sequences and successfully tracked feature points.

Table 1 The computation time for the multi-stage learning of the sequences in Fig. 7.

	A	B	C
number of frames	30	17	100
number of points	136	63	73
computation time (sec)	2.50	0.51	1.49

Table 2 Segmentation accuracy (%) for the sequences in Fig. 7.

	A	B	C
Costeira-Kanade	60.3	71.3	58.8
Ichimura	92.6	80.1	68.3
subspace separation	59.3	99.5	98.9
affine space separation	81.8	99.7	67.5
opt. subspace separation	99.0	99.6	99.6
opt. affine space separation	99.0	99.8	69.3
multi-stage learning	**100.0**	**100.0**	**100.0**

learning achieves 100% accuracy.

6. Concluding Remarks

In this paper, we proposed a multi-stage learning scheme first assuming degenerate motions and then assuming general 3-D motions. Doing simulations and real video experiments, we confirmed that our method is superior to all existing methods in realistic circumstances.

The reason for this superiority is that our method is tuned to realistic circumstances, where the motions of objects and backgrounds are almost degenerate, whereas most existing methods implicitly assume that objects and backgrounds undergo general 3-D motions. As a result, they perform very poorly for simple motions such as in Fig. 7, while our method[†] has very high performance without compromising the accuracy for considerably non-degenerate motions.

Acknowledgments

This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under a Grant in Aid for Scientific Research C(2) (No. 15500113) and Kayamori Foundation of Informational Science Advancement.

References

[1] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," Int. J. Comput. Vis., vol.29, no.3, pp.159–179, Sept. 1998.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," J. Roy. Statist. Soc., ser.B, vol.39, pp.1–38, 1977.

[3] C. W. Gear, "Multibody grouping from motion images, Int. J. Comput. Vis., vol.29, no.2, pp.133–150, Aug./Sept. 1998.

[4] N. Ichimura, "Motion segmentation based on factorization method and discriminant criterion," Proc. 7th Int. Conf. Comput. Vis., vol.1, pp.600–605, Kerkyra, Greece, Sept. 1999.

[5] N. Ichimura, "Motion segmentation using feature selection and subspace method based on shape space," Proc. 15th Int. Conf. Pattern Recog., vol.3, pp.858–864, Barcelona, Spain, Sept. 2000.

[6] K. Inoue and K. Urahama, "Separation of multiple objects in motion images by clustering," Proc. 8th Int. Conf. Comput. Vis., vol.1, pp.219–224, Vancouver, Canada, July 2001.

[7] K. Kanatani, "Geometric information criterion for model selection," Int. J. Comput. Vis., vol.26, no.3, pp.171–189, Feb./March 1998.

[8] K. Kanatani, "Motion segmentation by subspace separation and model selection," Proc. 8th Int. Conf. Comput. Vis., vol.2, pp.301–306, Vancouver, Canada, July 2001.

[9] K. Kanatani, "Motion segmentation by subspace separation: Model selection and reliability evaluation," Int. J. Image Graphics, vol.2, no.2, pp.179–197, April 2002.

[10] K. Kanatani, "Evaluation and selection of models for motion segmentation," Proc. 7th Euro. Conf. Comput. Vis., vol. 3,pp. 335–349, Copenhagen, Denmark, June 2002.

[†] The program is available at:
http://www.suri.it.okayama-u.ac.jp/e-program.html

[11] K. Kanatani and C. Matsunaga, “Estimating the number of independent motions for multibody segmentation,” Proc. 5th Asian Conf. Comput. Vis., vol.1, pp.7–12, Melbourne, Australia, Jan. 2002.

[12] N. Otsu, “A threshold selection method from gray-level histograms,” IEEE Trans. Syst. Man Cybern., vol.9, no.1, pp.62–66, Jan. 1979.

[13] C. J. Poelman and T. Kanade, “A paraperspective factorization method for shape and motion recovery,” IEEE Trans. Pattern Anal. Mach. Intell., vol.19, no.3, pp.206–218, March 1997.

[14] M. I. Schlesinger and V. Hlaváč, Ten Lectures on Statistical and Structural Pattern Recognition, Kluwer, Dordrecht, The Netherlands, 2002.

[15] Y. Sugaya and K. Kanatani, “Automatic camera model selection for multibody motion segmentation,” Proc. Workshop on Science of Computer Vision, pp.31–39, Okayama, Japan, Sept. 2002.

[16] Y. Sugaya and K. Kanatani, “Automatic camera model selection for multibody motion segmentation,” IAPR Workshop on Machine Vision Applications, Nara, Japan, pp.412–415, Dec. 2002.

[17] Y. Sugaya and K. Kanatani, “Outlier removal for motion tracking by subspace separation,” IEICE Trans. Inf. & Syst., vol.E86-D, no.6, pp.1095–1102, June 2003.

[18] C. Tomasi and T. Kanade, “Detection and tracking of point features,” CMU Tech. Rep. CMU-CS-91-132, April 1991. http://vision.stanford.edu/~birch/klt/

[19] Y. Wu, Z. Zhang, T. S. Huang, and J. Y. Lin, “Multibody grouping via orthogonal subspace decomposition, sequences under affine projection,” Proc. IEEE Conf. Computer Vision Pattern Recog., vol.2, pp.695–701, Kauai, Hawaii, U.S.A., Dec. 2001.

Yasuyuki Sugaya received his M.S. and Ph.D. from the University of Tsukuba, Ibaraki, Japan, in 1998 and 2001, respectively. He is currently Assistant Professor of information technology at Okayama University, Okayama, Japan. His research interests include image processing and computer vision.

Kenichi Kanatani received his M.S. and Ph.D. in applied mathematics from the University of Tokyo in 1974 and 1979, respectively. After serving as Professor of computer science at Gunma University, Gunma, Japan, he is currently Professor of information technology at Okayama University, Okayama, Japan. He is the author of many books on computer vision. He is an IEEE Fellow.