2003 年情報論的学習理論ワークショップ
2003 Workshop on Information-Based Induction Sciences (IBIS2003)
Kyoto, Japan, Nov 11 - 12, 2003, pp. 113–118.

Multi-stage Unsupervised Learning for Multi-body Motion Segmentation

Yasuyuki Sugaya* Kenichi Kanatani*

Abstract: Many techniques have been proposed for separating 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 separation 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. This multi-stage learning enables us to not only separate simple motions that we frequently encounter with high precision but also preserve the high performance for considerably general 3-D motions. Doing simulations and real video experiments, we demonstrate that our method is superior to all existing methods.

1. Introduction

Separating 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 [11]. He also used the QR decomposition [5]. Inoue and Urahama [6] introduced fuzzy clustering. Kanatani [8, 9, 10] incorporated model selection using the geometric AIC [7]. Wu et al. [18] 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 (not necessarily along straight lines). This type of degeneracy frequently occurs in practical applications. Though strict degeneracy may be rare, the segmentation accuracy considerably deteriorates if the motions are nearly degenerate.

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, we have found through our experiments that many methods that exhibit high accuracy for complicated simulations perform very poorly for real video sequences.

In this paper, we introduce unsupervised learning [13] assuming degenerate motions followed by unsupervised learning assuming general 3-D motions. This multi-stage learning enables us to not only separate simple motions with high precision but also preserve the high performance for considerably general 3-D motions.

*Department of Information Technology, Okayama University, Okayama 700-8530, Japan, phone (086)251-8244/8173, e-mail: {sugaya,kanatani}@suri.it.okayama-u.ac.jp

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 $2M$-dimensional *trajectory vector*:

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

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 (including the background) 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

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

2.2 Affine camera model

We assume an *affine camera*, which generalizes orthographic, weak perspective, and paraperspective projections [12]: the 3-D point $\boldsymbol{r}_{\kappa\alpha}$ is projected onto the image position

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

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

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

where $\tilde{\boldsymbol{m}}_{0\kappa}$, $\tilde{\boldsymbol{m}}_{1\kappa}$, $\tilde{\boldsymbol{m}}_{2\kappa}$, and $\tilde{\boldsymbol{m}}_{3\kappa}$ are 2-dimensional 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

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

where $\boldsymbol{m}_0$, $\boldsymbol{m}_1$, $\boldsymbol{m}_2$, and $\boldsymbol{m}_3$ are the $2M$-dimensional 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-dimensional 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-dimensional 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-dimensional affine space* within that 4-dimensional subspace[1]. It follows that multiple moving objects can be segmented into individual motions by separating the trajectory vectors $\{\boldsymbol{p}_\alpha\}$ into distinct 3-dimensional 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. In fact, according to simulations, the affine space separation performs better than the subspace separation except in the case in which perspective effects are very strong in the presence of small noise [10]. For real video sequences, however, the affine space separation accuracy is sometimes lower than that of the subspace separation [14, 15], which is inconsistent with the simulation results. The cause of this inconsistency will be clarified in the subsequent analysis.

3. Unsupervised Learning

3.1 Principle

Segmentation by the subspace and affine space separation is not always correct. Here, we consider optimizing the segmentation *a posteriori* by optimally fitting a 3-dimensional affine space (or a 4-dimensional subspace) to each trajectory class, considering the data distributions over the fitted spaces. Then, each trajectory is fractionally classified to all classes according to the posterior probability of its belonging. This process is iterated until the classification converges.

This is the standard approach to unsupervised learning for pattern recognition. However, the existence of geometric constraints somewhat complicates the likelihood computation. For the affine space constraint, the actual procedure becomes as follows (the procedure for the subspaces constraint goes similarly).

[1]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.

3.2 Procedure

Let $n = 2M$. Suppose N n-dimensional trajectory vectors $\{\boldsymbol{p}_\alpha\}$ are initially classified into m classes. Define the weight $W_\alpha^{(k)}$ of the vector $\boldsymbol{p}_\alpha$ for the kth class by

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

Then, 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
$$w^{(k)} = \frac{1}{N} \sum_{\alpha=1}^{N} W_\alpha^{(k)}. \tag{7}$$
2. Compute the centroid $\boldsymbol{p}_C^{(k)}$ of the kth class:
$$\boldsymbol{p}_C^{(k)} = \frac{\sum_{\alpha=1}^N W_\alpha^{(k)} \boldsymbol{p}_\alpha}{\sum_{\alpha=1}^N W_\alpha^{(k)}}. \tag{8}$$
3. Compute the $n \times n$ moment matrix of the kth class:
$$\boldsymbol{M}^{(k)} = \frac{\sum_{\alpha=1}^N W_\alpha^{(k)} (\boldsymbol{p}_\alpha - \boldsymbol{p}_C^{(k)})(\boldsymbol{p}_\alpha - \boldsymbol{p}_C^{(k)})^\top}{\sum_{\alpha=1}^N W_\alpha^{(k)}}. \tag{9}$$
4. Let $\lambda_1 \geq \lambda_2 \geq \lambda_3$ be the three largest 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
$$\hat{\sigma}_k^2 = \max[\frac{\mathrm{tr}[\boldsymbol{P}_\perp^{(k)} \boldsymbol{M}^{(k)} \boldsymbol{P}_\perp^{(k)}]}{n-3}, \sigma^2], \tag{11}$$
where $\mathrm{tr}[\,\cdot\,]$ denotes the trace and σ is an estimate of the tracking accuracy[2].
7. Compute the $n \times n$ covariance matrix of the kth class 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]We found $\sigma = 0.5$ (pixels) a reasonable value [16].

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, the αth trajectory is classified into the kth class 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-dimensional 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 these distributions, we compute the probability $P(\alpha|k)$ of the trajectory vector $\boldsymbol{p}_\alpha$ conditioned to be in the kth class (Eq. (14)). Regarding the fraction $w^{(k)}$ as the a priori probability of the kth class (Eq. (7)), we compute the a posterior probability $W_\alpha^{(k)}$ by Bayes' theorem (Eq. (14)). Then, we reclassify all the trajectories according to $W_\alpha^{(k)}$, which are fractions in general. (i.e., one trajectory can belong to multiple classes with fractional weights). This procedure is iterated until all the weights $W_\alpha^{(k)}$ converge. Finally, we associate the αth trajectory with the kth class that maximizes $W_\alpha^{(k)}$.

This type of unsupervised learning[3] is widely used for pattern recognition, and the likelihood is known to increases monotonously in the course of iterations [13]. However, it is also well known that the iterations are very likely to be trapped at a local maximum. It is almost impossible to do correct segmentation by the above procedure alone unless we start from a very good initial value.

4. Degenerate Motion Model

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 $\boldsymbol{0}$; 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-dimensional affine space passing through $\boldsymbol{m}_0$ and spanned by $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ [14, 15].

[3]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 given by Bayes' theorem.

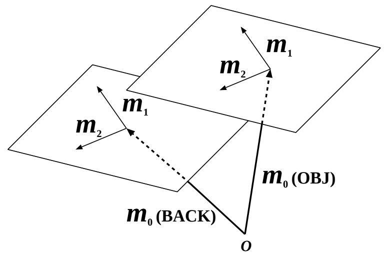

Figure 1: If the motions of the objects and the background are degenerate, their trajectory vectors belong to mutually parallel 2-dimensional affine spaces.

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-dimensional 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. Since the basis vectors $\boldsymbol{i}_\kappa$ and $\boldsymbol{j}_\kappa$ are common to the objects and the background, the vectors $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ in Eq. (5) are also common. Thus, the 2-dimensional affine spaces of all the motions are *parallel* (Fig. 1).

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

4.2 Learning for degenerate motions

Since most of the motions we encounter in practice are degenerate, we can expect that the segmentation accuracy increases by unsupervised learning assuming such degenerate motions. The actual procedure goes as follows:

First, we set the weight $W_\alpha^{(k)}$ of $\boldsymbol{p}_\alpha$ for the kth class by Eq. (6). Next, we iterate the following procedures A, B, and C in turn until all the weights $\{W_\alpha^{(k)}\}$ converge:

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

1. Compute $w^{(k)}$ by Eq. (7).
2. Compute the centroid $\boldsymbol{p}_C^{(k)}$ of the kth class 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

$$\boldsymbol{M} = \sum_{k=1}^{m} w^{(k)} \boldsymbol{M}^{(k)}. \tag{15}$$

2. Let $\lambda_1 \geq \lambda_2$ be the two largest eigenvalues of the matrix $\boldsymbol{M}$, and $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ the corresponding unit eigenvectors.

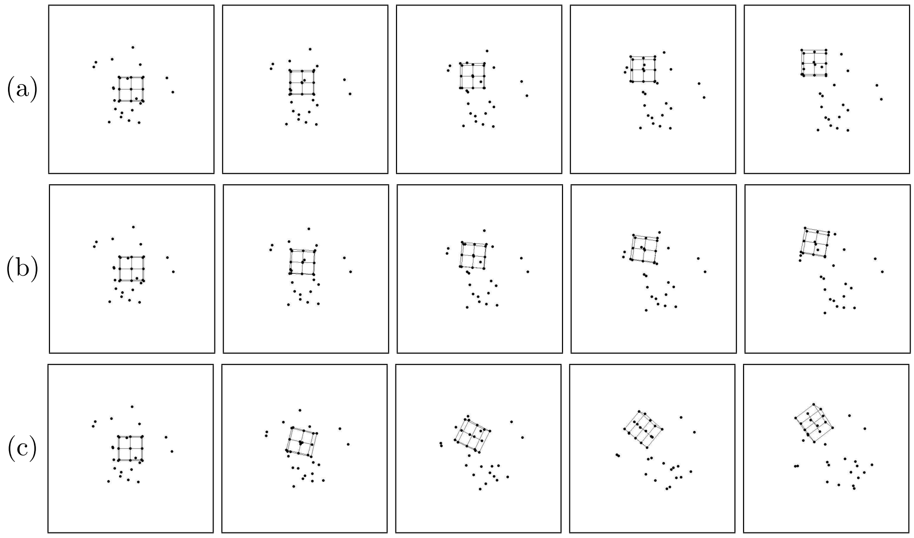

Figure 2: Simulated image sequences of 14 object points and 20 background points: (a) almost degenerate motion; (b) nearly degenerate motion; (c) general 3-D motion.

3. Compute the $n \times n$ projection matrices

$$\boldsymbol{P} = \sum_{i=1}^{2} \boldsymbol{u}_i \boldsymbol{u}_i^\top, \quad \boldsymbol{P}_\perp = \boldsymbol{I} - \boldsymbol{P}. \tag{16}$$

4. Estimate the noise variance in the direction orthogonal to all the affine spaces by

$$\hat{\sigma}^2 = \max[\frac{\mathrm{tr}[\boldsymbol{P}_\perp \boldsymbol{M} \boldsymbol{P}_\perp]}{n-2}, \sigma^2]. \tag{17}$$

5. Compute the $n \times n$ covariance matrix of the kth class by

$$\boldsymbol{V}^{(k)} = \boldsymbol{P}\boldsymbol{M}^{(k)}\boldsymbol{P} + \hat{\sigma}^2 \boldsymbol{P}_\perp. \tag{18}$$

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-dimensional 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.

After the iterations of A, B, and C have converged, the αth trajectory is classified to the kth class that maximizes $W_\alpha^{(k)}$, $k = 1, ..., m$.

4.3 Multi-stage learning

In order to start the above learning, we need a good initial value. Here, we use the affine space separation using 2-dimensional affine spaces, which effectively assumes planar motions with varying sizes. The resulting segmentation is then optimized by unsupervised learning assuming non-rotational motions.

The solution should be very accurate if the motions are truly degenerate. In reality, however, rotations may be involved to some extent. So, we relax the constraint and optimize the solution by unsupervised learning assuming general 3-D motions. In sum, our scheme consists of the following three stages:

1. Initial segmentation by the affine space separation using 2-dimensional affine spaces.
2. Unsupervised learning assuming degenerate motions.
3. Unsupervised learning assuming general 3-D motions.

This multi-stage learning enables us to not only separate degenerate motions that we frequently encounter with high precision but also preserve the high performance for general 3-D motions, as we now show.

5. Simulation Experiments

Fig. 2 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 given rotations by 2° per frame in opposite senses around different axes; they make 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).

We added independent Gaussian random noise of mean 0 and standard deviation σ to the coordinates of all the points and segmented them into two groups. Fig. 3 plots the average misclassification ratio over 500

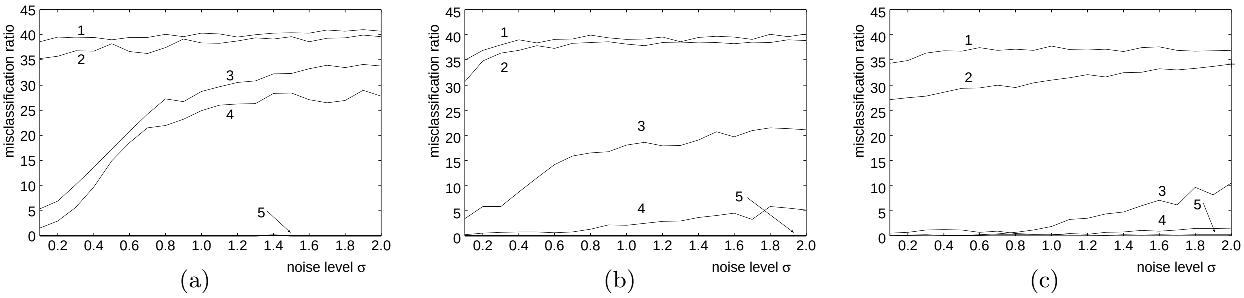

Figure 3: Misclassification ratio for the sequences (a), (b), and (c) in Fig. 2: 1) Costeira-Kanade; 2) Ichimura; 3) optimized subspace separation; 4) optimized affine space separation; 5) multi-stage learning.

trials using different noise for different σ. We compared 1) the Costeira-Kanade method [1], 2) Ichimura's method [4], 3) the subspace separation [8, 9] followed by unsupervised learning (we call this *optimized subspace separation* for short), 4) the affine space separation [10] followed by unsupervised learning (*optimized affine space separation* for short), and 5) our multi-stage learning.

For the almost degenerate motion in Fig. 2(a), the optimized subspace separation and the optimized affine space separation do not work very well. Also, the affine space separation is not superior to the subspace separation (Fig. 3(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. 2(b), the optimized subspace separation and the optimized affine space separation both work fairly well (Fig. 3(b)). However, our method still attains almost 100% accuracy.

For the general 3-D motion in Fig. 2(c), the optimized subspace separation and the optimized affine space separation exhibit relatively high performance (Fig. 3(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. This is the reason why the multi-stage learning achieves high performance even for considerably non-degenerate motions.

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. 2(c), but it does not compare with the optimized subspace or affine space separation, much less with our multi-stage learning method.

6. Real Video Examples

Fig. 4 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 [17]. The marks $\square$ indicate their positions. From the trajectories tracked throughout the sequence, we removed outlier trajectories using the method of Sugaya and Kanatani [16].

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-dimensional vectors [14]. We used Pentium 4 2.4B GHz for the CPU with 1 Gb main memory and Linux for the OS.

Table 2 lists the segmentation accuracies for different methods ("opt" stands for "optimized"). The accuracy is measured by (the number of correctly classified points)/(the total number of points) in percentage.

As we can see, the Costeira-Kanade method fails to produce meaningful segmentation. Ichimura's method is effective for sequences A and B but not so very 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, in contrast, the subspace separation is superior to the affine space separation, strongly 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 unsupervised learning very much depends on the quality of the initial segmentation. For all the three sequences, our multi-stage learning achieves 100% accuracy.

7. Concluding Remarks

In this paper, we have proposed a multi-stage unsupervised learning scheme first assuming degenerate motions and then assuming general 3-D motions. Doing simulations and real video experiments, we have 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, while existing methods mostly make use of the shape interaction matrix of Costeira and Kanade on the assumption that objects and backgrounds undergo general 3-D motions. As a result, they perform very poorly for simple motions such as in Fig. 4, while our method[4]

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

Figure 4: Three video sequences and successfully tracked feature points.

Table 1: The computation time for the multi-stage learning of the sequences in Fig. 4.

	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. 4.

	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**

has very high performance for degenerate motions, and the accuracy is preserved even for considerably non-degenerate motions due to the multi-stage learning.

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), the Support Center for Advanced Telecommunications Technology Research, 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. Computer Vision, 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. Vision, 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. Vision, Kerkyra, Greece, vol.1, pp.600–605, Sept. 1999.

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

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

[7] K. Kanatani, Geometric information criterion for model selection, Int. J. Comput. Vision, 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. Vision, Vancouver, Canada, vol.2, pp.301–306, 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. Vision, Copenhagen, Denmark, vol. 3,pp. 335–349, June 2002.

[11] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Sys. Man Cyber., vol.9, no.1, pp.62-66, Jan. 1979.

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

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

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

[15] 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.

[16] 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.

[17] C. Tomasi and T. Kanade, Detection and Tracking of Point Features, CMU Tech. Rep. CMU-CS-91-132, Apr. 1991: http://vision.stanford.edu/~birch/klt/

[18] 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.