# Geometric Structure of Degeneracy for Multi-body Motion Segmentation

Yasuyuki Sugaya and Kenichi Kanatani

Depertment of Information Technology,
Okayama University, Okayama
700-8530 Japan
{kanatani, sugaya}@suri.it.okayama-u.ac.jp

**Abstract.** Many techniques have been proposed for segmenting feature point trajectories tracked through a video sequence into independent motions. It has been found, however, that methods that perform very well in simulations perform very poorly for real video sequences. This paper resolves this mystery by analyzing the geometric structure of the degeneracy of the motion model. This leads to a new segmentation algorithm: a multi-stage unsupervised learning scheme first using the degenerate motion model and then using the general 3-D motion model. We demonstrate by simulated and real video experiments that our method is superior to all existing methods in practical situations.

## 1 Introduction

Segmenting feature point trajectories tracked through a video sequence into independent motions is a 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 [13]. He also used the QR decomposition [5]. Inoue and Urahama [6] introduced fuzzy clustering. Kanatani [8, 9, 10] incorporated model selection by the geometric AIC [7] and robust estimation by LMedS [15]. Wu et al. [21] introduced orthogonal subspace decomposition.

According to our experiments, however, many methods that exhibit high accuracy in simulations perform very poorly for real video sequences. In this paper, we show that this inconsistency is caused by the *degeneracy* of the motion model on which the segmentation is based. The existence of such degeneracy was already pointed out by Costeira and Kanade [1]. Here, we report a new type of degeneracy, which we call *parallel 2-D plane degeneracy*, that, according to our experience, most frequently occurs in realistic scenes.

This discovery leads to a new segmentation algorithm, which we call *multi-stage unsupervised learning*: it operates first using our degeneracy model and then using the general motion model. We demonstrate that our method is superior to all existing methods in practical situations.

D. Comaniciu et al. (Eds.): SMVP 2004, LNCS 3247, pp. 13–25, 2004.

In Sec. 2, we describe the geometric constraints that underlie our method. In Sec. 3, we analyze the degeneracy of motion model. Sec. 4 describes our multi-stage learning scheme. In Sec. 5, we show synthetic and real video examples. Sec. 6 concludes this paper.

## 2 Geometric Constraints

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

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

For convenience, we identify the frame number $\kappa$ with "time" and refer to the $\kappa$th frame as "time $\kappa$".

We regard the $XYZ$ camera coordinate system as a reference, 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 $\kappa$. Let $(a_\alpha, b_\alpha, c_\alpha)$ be the coordinates of the $\alpha$th point that belong to that object. Its position with respect to the reference frame at time $\kappa$ is

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

We assume an affine camera, which generalizes orthographic, weak perspective, and paraperspective projections [12, 14]: 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 $2 \times 3$ matrix and a 2-D vector determined by the position and orientation of the camera and its internal parameters at time $\kappa$. 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-D vectors determined by the position and orientation of the camera and its internal parameters at time $\kappa$. 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 $2M$-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.

Eq. (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 $\alpha$. This means that the trajectories are in a *3-D affine space* within that 4-D 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-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. Indeed, it is reported that in simulation the affine space separation 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, we have found that the affine space separation accuracy is often lower than that of the subspace separation [18, 19]. To resolve this inconsistency is the first goal of this paper.

## 3 Structure of Degeneracy

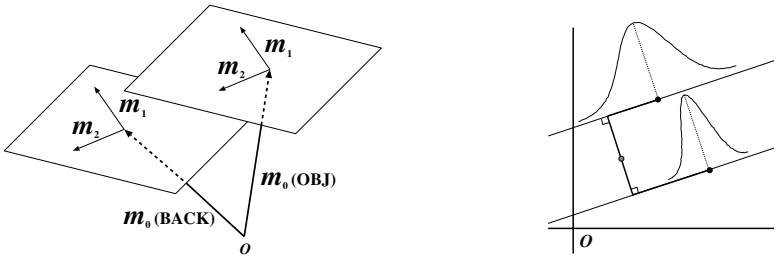
The motions we most frequently encounter are such that the objects and the background are translating and rotating 2-dimensionally in the image frame with varying sizes. For such a motion, we can choose the basis vector $\boldsymbol{k}_\kappa$ in Eq. (2) in the $Z$ direction (the camera optical axis 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 term $c_\alpha \tilde{\boldsymbol{m}}_{3\kappa}$ in Eq. (4) vanishes; 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 $\kappa$.

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 $\boldsymbol{m}_2$ [18, 19]. 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 all 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, or "planes", of all the motions are *parallel* (Fig. 1(a)).

Note that *parallel 2-D planes 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.

[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 [12].




(a) (b)

**Fig. 1.** (a) If the motions of the objects and the background are degenerate, their trajectory vectors belong to mutually parallel 2-D planes. (b) The data distributions inside the individual 2-D planes are modeled by Gaussian distributions

## 4 Degeneracy-Tuned Learning

We now define an unsupervised learning scheme [16] tuned to the parallel 2-D plane degeneracy. We assume that the noise in the coordinates of the feature points is an independent Gaussian random variable of mean 0 and a constant variance. We also model the data distributions inside the individual 2-D planes by Gaussian distributions (Fig.1(b)).

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

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

Then, we iterate the following procedures A, B, and C in turn until all the weights $\{W_\alpha^{(k)}\}$ converge[2].

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

1. Compute the fractional size $w^{(k)}$ and the centroid $\boldsymbol{p}_C^{(k)}$ of the class $k$:

$$w^{(k)} = \frac{1}{N}\sum_{\alpha=1}^{N} W_\alpha^{(k)}, \qquad \boldsymbol{p}_C^{(k)} = \frac{\sum_{\alpha=1}^{N} W_\alpha^{(k)} \boldsymbol{p}_\alpha}{\sum_{\alpha=1}^{N} W_\alpha^{(k)}}. \tag{7}$$

2. Compute the $n \times n$ (second-order) moment matrix $\boldsymbol{M}^{(k)}$:

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

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

B. Do the following computation.

1. Compute the *total* $n \times n$ moment matrix

$$\boldsymbol{M} = \sum_{k=1}^{m} w^{(k)} \boldsymbol{M}^{(k)}. \tag{9}$$

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 *common* $n \times n$ projection matrices ($\boldsymbol{I}$ denotes the $n \times n$ unit matrix):

$$\boldsymbol{P} = \sum_{i=1}^{2} \boldsymbol{u}_i \boldsymbol{u}_i^\top, \qquad \boldsymbol{P}_\perp = \boldsymbol{I} - \boldsymbol{P}. \tag{10}$$

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

$$\hat{\sigma}^2 = \max[\frac{\mathrm{tr}[\boldsymbol{P}_\perp \boldsymbol{M} \boldsymbol{P}_\perp]}{n-2}, \sigma^2], \tag{11}$$

where $\mathrm{tr}[\,\cdot\,]$ denotes the trace and $\sigma$ is an estimate of the tracking accuracy[3].
5. Compute the $n \times n$ covariance matrix of the class $k$ by

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

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

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

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, B, and C have converged, the $\alpha$th trajectory is classified into the class $k$ that maximizes $W_\alpha^{(k)}$, $k = 1, ..., N$.

In the above iterations, we fit 2-D planes of the same orientation to all classes by computing the common basis vectors $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ from all the data. We also estimate a common outside noise variance from all the data. Regarding the fraction $w^{(k)}$ (the first of Eqs. (7)) as the *a priori probability* of the class $k$, we compute the probability[4] $P(\alpha|k)$ of the trajectory vector $\boldsymbol{p}_\alpha$ conditioned to be in the class $k$ (Eq. (13)). Then, we apply *Bayes' theorem* (Eq. (14)) to compute the *a posterior probability* $W_\alpha^{(k)}$, according which all the trajectories are reclassified.

[3] The value $\sigma = 0.5$ (pixels) is suggested in [17] as a reasonable estimate.
[4] Multipliers independent of $\alpha$ and $k$ are omitted. They cancel out in Eq. (14).

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.

This type of learning[5] is widely used for clustering, and the likelihood is known to increase monotonously by iterations [16]. 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.

## 5 Multi-stage Learning

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. For this, we use the Kanatani's affine space separation [10], which combines the shape interaction matrix of Costeira and Kanade [1], model selection by the geometric AIC [7], and robust estimation by LMedS [15]. segmentation by using the parallel plane degeneracy model, as described in the preceding section.

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 by using the general 3-D motion model. 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, 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 using the parallel 2-D plane degeneracy model.
3. Unsupervised learning using the general 3-D motion model.

The last stage is similar to the second except that 3-D affine spaces are separately fitted to individual classes. The outside noise variance is also estimated separately for each class. The procedure goes as follows.

Initializing the weight $W_\alpha^{(k)}$ by Eq. (6), we iterate the following procedures A and B in turn until all $\{W_\alpha^{(k)}\}$ converge[6]:

[5] 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.

[6] The convergence condition is the same as in Sec. 4: see footnote 2.

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

1. Compute the fraction $w^{(k)}$ and the centroid $\boldsymbol{p}_C^{(k)}$ by Eqs. (7).
2. Compute the $n \times n$ moment matrix $\boldsymbol{M}^{(k)}$ by Eq. (8).
3. 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.
4. Compute the $n \times n$ projection matrices

$$\boldsymbol{P}^{(k)} = \sum_{i=1}^{3} \boldsymbol{u}_i^{(k)} \boldsymbol{u}_i^{(k)\top}, \qquad \boldsymbol{P}_\perp^{(k)} = \boldsymbol{I} - \boldsymbol{P}^{(k)}. \tag{15}$$

5. Estimate the noise variance in the direction orthogonal to the affine space of the class $k$ by

$$\hat{\sigma}_k^2 = \max[\frac{\mathrm{tr}[\boldsymbol{P}_\perp^{(k)} \boldsymbol{M}^{(k)} \boldsymbol{P}_\perp^{(k)}]}{n-3}, \sigma^2]. \tag{16}$$

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

$$\boldsymbol{V}^{(k)} = \boldsymbol{P}^{(k)} \boldsymbol{M}^{(k)} \boldsymbol{P}^{(k)} + \hat{\sigma}_k^2 \boldsymbol{P}_\perp^{(k)}. \tag{17}$$

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

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

## 6 Other Issues

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]: 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 *from motion information alone*, except using heuristic thresholds or ad-hoc criteria. If model selection such as the geometric AIC [7] and the geometric MDL [11] is used[7], the resulting number of motions depends on criteria as reported in [9]. In order to determine the number $m$ of motions, one needs high-level processing using color, shape, and other information.

The feature point trajectories tracked through video frames are not necessarily correct, so we need to remove outliers. If the trajectories were segmented

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

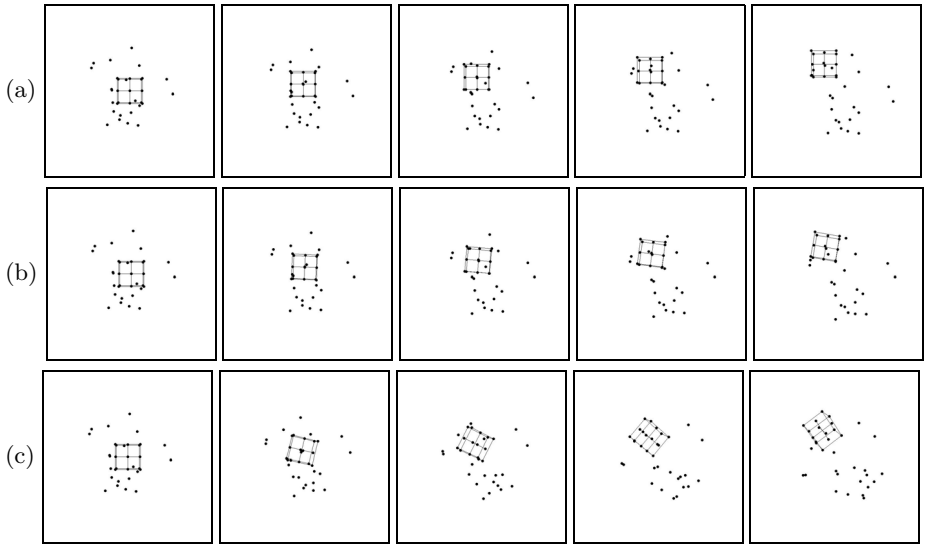



**Fig. 2.** Simulated image sequences of 14 object points and 20 background points: (a) almost degenerate motion; (b) nearly degenerate motion; (c) general 3-D motion

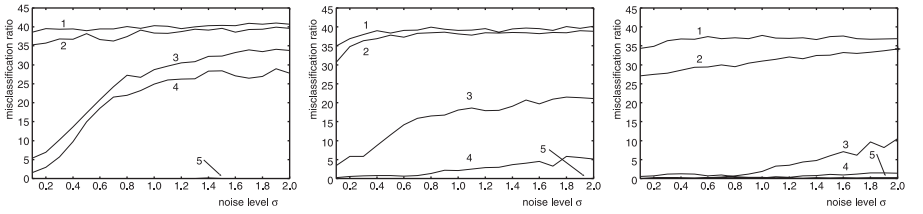
into individual classes, we could remove, for example, those that do not fit to the individual affine spaces. In the presence of outliers, however, we cannot do correct segmentation, and hence we do not know the affine spaces.

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[8], although some inliers were also removed for safety [17].

## 7 Simulation Experiments

Fig. 2 shows three sequences of five synthetic images (supposedly of $512 \times 512$ 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 $30^\circ$ angle of view, although the underlying theory is based on the affine camera model without perspective effects.

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




(a) (b) (c)

**Fig. 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

In all the these 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).

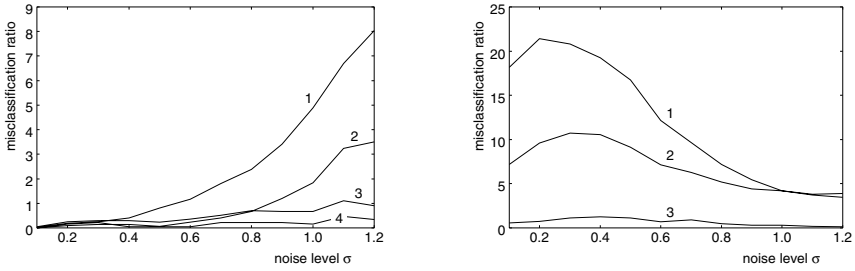
Adding independent Gaussian random noise of mean 0 and standard deviation $\sigma$ to the coordinates of all the points, we segmented them into two groups. Fig. 3 plots the average misclassification ratio over 500 trials using different noise. We compared 1) the Costeira-Kanade method [1], 2) Ichimura's method [4], 3) the subspace separation [8, 9] followed by unsupervised learning (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 and affine space separations do not work very well. Also, the latter is not superior to the former (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 and affine space separations 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 and affine space separations 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. 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




(a) (b)

**Fig. 4.** Comparison of misclassification ratios: (a) Effects of unsupervised learning: 1) subspace separation; 2) optimized subspace separation; 3) affine space separation; 4) optimized affine space separation. (b) Stage-wise effect of multi-stage learning: 1) affine space separation using 2-D affine spaces; 2) unsupervised learning using the parallel 2-D plane degeneracy model; 3) unsupervised learning using the general 3-D motion model

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

Fig. 4(a) shows the effects of learning for Fig. 2(c). We can see that the learning works effectively. As compared with them, however, our multi-stage learning is far better. Fig. 4(b) shows the stage-wise effects of our multi-stage learning for Fig. 2(c). For this general 3-D motion, the learning using the parallel 2-D plane degeneracy model does not perform so very well indeed, but the subsequent learning based on the general 3-D motion model successfully restores the accuracy up to almost 100%. The interesting fact is that the accuracy increases as the noise increases. This reflects the characteristics of the initial affine space separation, whose accuracy deteriorates if perspective projection is affinely approximated for accurate data [10].

## 8 Real Video Experiments

Fig. 5 shows five decimated frames from three video sequences A, B, and C ($320 \times 240$ pixels). For each sequence, we detected feature points in the initial frame and tracked them using the Kanade-Lucas-Tomasi algorithm [20]. The marks $\square$ indicate their positions.

Table 1(a) lists the number of frames, the number of inlier trajectories, and the computation time for our multi-stage learning. The computation time is reduced by compressing the trajectory data into 8-D vectors [18]. We used Pentium 4 2.4GHz for the CPU with 1GB main memory and Linux for the OS. Table 1(b) 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. Except for the Costeira-Kanade and Ichimura methods, the percentage is averaged over 50 trials, since the subspace and affine space separations inter-


**Fig. 5.** Three video sequences and successfully tracked feature points

**Table 1.** (a) The computation time for the multi-stage learning of the sequences in Fig. 5. (b) Segmentation accuracy (%) for the sequences in Fig. 5

(a)

| | A | B | C |
|---|---|---|---|
| # of frames | 30 | 17 | 100 |
| # of points | 136 | 63 | 73 |
| CPU time (sec) | 2.50 | 0.51 | 1.49 |

(b)

| | 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** |

nally use random sampling for robust estimation and hence the result is slightly different for each trial.

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 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 learning achieves 100% accuracy.

## 9 Conclusions

In this paper, we analyzed the geometric structure of the degeneracy of the motion model that underlies the subspace and affine space separation methods [8–10] and resolved the apparent inconsistency that the affine space separation accuracy is often lower than that of the subspace separation for real video sequences. Our conclusion is that this is due to the occurrence of a special type of degeneracy, which we call *parallel 2-D plane degeneracy*.

Exploiting this finding, we proposed a multi-stage learning scheme first using the parallel 2-D plane degeneracy model and then using the general 3-D motion model. Doing simulations and real video experiments, we demonstrated 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. 5, while our method[9] has very high performance without compromising the accuracy for considerably non-degenerate motions.

## References

1. J. P. Costeira and T. Kanade, A multibody factorization method for independently moving objects, *Int. J. Comput. Vision*, **29**-3 (1998-9), 159–179.
2. A. P. Dempster, N. M. Laird and D. B. Rubin, Maximum likelihood from incomplete data via the EM Algorithm, *J. Roy. Statist. Soc.*, B**39** (1977), 1–38.
3. C. W. Gear, Multibody grouping from motion images, *Int. J. Comput. Vision*, **29**-2 (1998-8/9), 133–150.
4. N. Ichimura, Motion segmentation based on factorization method and discriminant criterion, *Proc. 7th Int. Conf. Comput. Vision*, Vol. 1, Kerkyra, Greece, September 1999, pp. 600–605.
5. N. Ichimura, Motion segmentation using feature selection and subspace method based on shape space, *Proc. 15th Int. Conf. Pattern Recog.*, Vol. 3, Barcelona, Spain, September 2000, pp. 858–864.
6. K. Inoue and K. Urahama, Separation of multiple objects in motion images by clustering, *Proc. 8th Int. Conf. Comput. Vision*, Vol. 1, Vancouver, Canada, July 2001, pp. 219–224.
7. K. Kanatani, Geometric information criterion for model selection, *Int. J. Comput. Vision*, **26**-3 (1998-2/3), 171–189.
8. K. Kanatani, Motion segmentation by subspace separation and model selection, *Proc. 8th Int. Conf. Comput. Vision*, Vol. 2, Vancouver, Canada, July 2001, pp. 301–306.
9. K. Kanatani, Motion segmentation by subspace separation: Model selection and reliability evaluation, *Int. J. Image Graphics*, **2**-2 (2002-4), 179–197.
10. K. Kanatani, Evaluation and selection of models for motion segmentation, *Proc. 7th Euro. Conf. Comput. Vision*, Vol. 3, Copenhagen, Denmark, June 2002, pp. 335–349.

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

11. K. Kanatani, Uncertainty modeling and model selection for geometric inference, *IEEE Trans. Patt. Anal. Mach. Intell.*, **26**-10 (2004), to appear.
12. K. Kanatani and Y. Sugaya Factorization without factorization: Complete Recipe, *Memoirs of the Faculty of Engineering, Okayama University*, **38**-2 (2004), 61–72.
13. N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans. Sys. Man Cyber.*, **9**-1 (1979-1), 62–66.
14. C. J. Poelman and T. Kanade, A paraperspective factorization method for shape and motion recovery, *IEEE Trans. Pat. Anal. Mach. Intell.*, **19**-3 (1997-3), 206–218.
15. P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, Wiley, New York, 1987.
16. M. I. Schlesinger and V. Hlaváč, *Ten Lectures on Statistical and Structural Pattern Recognition*, Kluwer, Dordrecht, The Netherlands, 2002.
17. Y. Sugaya and K. Kanatani, Outlier removal for motion tracking by subspace separation, *IEICE Trans. Inf. Syst.*, **E86-D**-6 (2003-6), 1095–1102.
18. Y. Sugaya and K. Kanatani, Automatic camera model selection for multibody motion segmentation, *Proc. Workshop on Science of Computer Vision*, Okayama, Japan, Sepember. 2002, pp. 31–39.
19. Y. Sugaya and K. Kanatani, Automatic camera model selection for multibody motion segmentation, *Proc. IAPR Workshop on Machine Vision Applications*, Nara, Japan, December 2002, pp. 412–415.
20. 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/`
21. 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, Kauai, Hawaii, U.S.A., December 2001, pp. 695–701.