Proc. Workshop on Science of Computer Vision, September 2002, Okayama Japan, pp. 31–39.

Automatic Camera Model Selection for Multibody Motion Segmentation

Yasuyuki Sugaya and Kenichi Kanatani

Department of Information Technology, Okayama University, Okayama 700-8530 Japan

{sugaya,kanatani}@suri.it.okayama-u.ac.jp

We study the problem of segmenting independently moving objects in a video sequence. Several algorithms exist for classifying the trajectories of the feature points into independent motions, but the performance depends on the validity of the underlying camera imaging model. In this paper, we present a scheme for automatically selecting the best model using the geometric AIC *before* the segmentation stage. Using real video sequences, we confirm that the segmentation accuracy indeed improves if the segmentation is based on the selected model. We also show that the trajectory data can be compressed into low-dimensional vectors using the selected model. This is very effective in reducing the computation time for a long video sequence.

1. Introduction

Segmenting individual objects from backgrounds is one of the most important tasks of video processing. For images taken by a stationary camera, many segmentation algorithms based on background subtraction and interframe subtraction have been proposed. For images taken by a moving camera, however, the segmentation is very difficult because the objects and the backgrounds are both moving in the image.

While most segmentation algorithms combine various heuristics based on miscellaneous cues such as optical flow, color, and texture, Costeira and Kanade [1] presented a segmentation algorithm based only on the image motion of feature points.

Since then, various modifications and extensions of their method have been proposed [3, 6, 10, 13, 15, 16]. Gear [3] used the reduced row echelon form and graph matching. Ichimura [6] applied the discrimination criterion of Otsu [20] and the QR decomposition for feature selection [7]. Inoue and Urahama [10] introduced fuzzy clustering. Incorporating model selection using the geometric AIC [12] and robust estimation using LMedS [22], Kanatani [13, 15, 16] derived segmentation algorithms called *subspace separation* and *affine space separation*. Maki and Wiles [18] and Maki and Hattori [19] used Kanatani's idea for analyzing the effect of illumination on moving objects. Wu, et al. [27] introduced orthogonal subspace decomposition.

To begin the segmentation, the number of independent motions needs to be estimated. This has usually been handled using empirical thresholds. Recently, Kanatani and Matsunaga [17] and Kanatani [15] proposed the use of model selection for this.

For tracking moving feature points, most authors use the Kanade-Lucas-Tomasi algorithm [24]. To improve the tracking accuracy, Huynh and Heyden [5] and Sugaya and Kanatani [23] showed that outlier trajectories can be removed by robust estimation using LMedS [22] and RANSAC [2]. Ichimura and Ikoma [8] and Ichimura [9] introduced nonlinear filtering.

In this paper, we propose a new method for improving the accuracy of Kanatani's subspace separation [13, 15] and affine space separation [16]. According to Kanatani [13, 16], the trajectories of feature points that belong to a rigid object are, under an affine camera model, constrained to be in a 4-dimensional subspace and at the same time in a 3-dimensional affine space in it. If the object is in a 2-dimensional rigid motion, the resulting trajectories are constrained to be in a 3-dimensional subspace or more strongly in a 2-dimensional affine space in it. Theoretically, the segmentation accuracy should be higher if we use stronger constraints. However, it has been pointed out that this is not necessarily true due to the modeling errors of the camera imaging geometry [16].

To cope with this, Kanatani [15, 16, 17] proposed *a posteriori* reliability evaluation using the geometric AIC [12] and the geometric MDL [14]. However, his procedure is based on the assumption that the segmentation is correctly done. In reality, if the final result is rejected as unreliable by Kanatani's method, one cannot tell whether the assumed model was wrong or the segmentation was not correctly done.

In this paper, we introduce model selection for choosing the best camera model and the associated space *before* doing segmentation. Using real video sequences, we demonstrate that the segmentation accuracy indeed improves if the segmentation is based on the selected model. We also show that we can compress the trajectory data into low-dimensional vectors by projecting them onto the subspace defined by the selected model. This is very effective in reducing the computation time for a long video sequence.

In Sec. 2, we summarize the subspace and affine space constraints that underlie our method. In Sec. 3, we discuss how the segmentation procedure is affected by the camera imaging model and motion patterns. In Sec. 4 and 5, we describe our procedure for selecting the best camera model *a priori* using the geometric AIC. In Sec. 6, we show how the trajectory data can be compressed into low-dimensional vectors. Sec. 7 summarizes our procedure. In Sec. 8, we demonstrate the effectiveness of our procedure by real video experiments. Sec. 9 is our conclusion.

2. Trajectory of Feature Points

We track N rigidly moving feature points over M frames and let $(x_{\kappa\alpha}, y_{\kappa\alpha})$ be the image coordinates of the αth point in the κth frame. We stack all the image coordinates vertically and represent the entire trajectory by the following *trajectory vector*:

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

Regarding the XYZ camera coordinate system as the world coordinate system, we fix a 3-D object coordinate system to the moving object. Let $\boldsymbol{t}_\kappa$ and $\{\boldsymbol{i}_\kappa, \boldsymbol{j}_\kappa, \boldsymbol{k}_\kappa\}$ be, respectively, its origin and 3-D orthonormal basis in the κth frame. If we let $(a_\alpha, b_\alpha, c_\alpha)$ be the 3-D object coordinates of the αth point, its 3-D position in the κth frame is

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

with respect to the world coordinate system.

If an affine camera model (e.g., orthographic, weak perspective, or paraperspective projection) is assumed, the 2-D position of $\boldsymbol{r}_\alpha$ in the image is given by

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

where $\boldsymbol{A}_\kappa$ and $\boldsymbol{b}_\kappa$ are, respectively, a 2×3 matrix and a 2-dimensional vector determined by the position and orientation of the camera and its internal parameters in the κth frame. From eq. (2), we can write eq. (3) as

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

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 in the κth frame. From eq. (4), the trajectory vector $\boldsymbol{p}_\alpha$ of eq. (1) can be written in the form

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

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.

3. Constraints on Image Motion

Eq. (5) implies that the trajectory vectors of the feature points that belong to the same 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 trajectory vectors $\{\boldsymbol{p}_\alpha\}$ into distinct 4-dimensional subspaces. This is the principle of the *subspace separation* [13, 15].

However, we can also see that the coefficient of $\boldsymbol{m}_0$ in eq. (5) is identically 1 for all α. This means that the trajectory vectors are also in the 3-dimensional affine space within that 4-dimensional subspace. 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 *affine space separation* [16].

Theoretically, the segmentation accuracy should be higher if a stronger constraint is used. However, eq. (5) was derived from an affine camera model, while the imaging geometry of real cameras is perspective projection. It can be shown [16] that the modeling errors for approximating the perspective projection by an affine camera are larger for the affine space constraint than for the subspace constraint. In general, the stronger the constraint, the more vulnerable to modeling errors. Conversely, the solution is more robust to modeling errors, if not very accurate, when weaker constraints are used.

According to Kanatani [16], the choice between the subspace separation and the affine space separation depends on the balance between the camera modeling errors and the image noise. The subspace separation performs well when the perspective effects are strong and the noise is small, while the affine space separation performs better for large noise with weak perspective effects. However, we do not know a priori which is the case for a given video sequence.

If the object motion is planar, i.e., if the object merely translates, rotates, and changes the scale within the 2-dimensional image, one of the three vectors $\boldsymbol{m}_1$, $\boldsymbol{m}_2$, and $\boldsymbol{m}_3$ can be set $\mathbf{0}$. Hence, $\boldsymbol{p}_\alpha$ is constrained to be in a 3-dimensional subspace. Since the coefficient of $\boldsymbol{m}_0$ is identically 1, $\boldsymbol{p}_\alpha$ is also in a 2-dimensional affine space within that 3-dimensional subspace. It follows that we can segment multiple planar motions into individual objects by separating the trajectory vectors $\{\boldsymbol{p}_\alpha\}$ into distinct 3-dimensional subspaces or distinct 2-dimensional affine spaces. However, we do not know a priori if the object motion is planar or which constraint should be used for a given video sequence.

4. A Priori Camera Models

For simplicity, let us hereafter call the constraint that specifies the camera imaging model and the type of motion the *camera model*. As we have observed, we can expect high accuracy if we know which camera model is suitable and accordingly use the corresponding algorithm. We may test all the models and the associated segmentation methods and evaluate the reliability of the results *a posteriori*, as Kanatani suggested [15, 16, 17]. However, this works only if the segmentation is done correctly; if the final result is rejected as unreliable, one cannot tell whether the assumed model was wrong or the segmentation was not correctly done.

To overcome this difficulty, we introduce camera models that should be valid *irrespective* of the segmentation results. If, for example, one object is moving relative to a stationary background while the camera is moving, two independent motions are observed in the image: the object motion and the background motion. Since the trajectory vectors for each motion is in a 4-dimensional subspace or a 3-dimensional affine space in it, the entire trajectory vectors $\{\boldsymbol{p}_\alpha\}$ should be in an 8-dimensional subspace $\mathcal{L}^8$ or a 7-dimensional affine space $\mathcal{A}^7$ in it[1].

If the object motion and the background motion are both planar, the trajectory vectors for each motion are in a 3-dimensional subspaces or a 2-dimensional affine spaces in it, so the entire trajectory vectors $\{\boldsymbol{p}_\alpha\}$ should be in a 6-dimensional subspace $\mathcal{L}^6$ or a 5-dimensional affine space $\mathcal{A}^5$ in it.

It follows that in the pre-segmentation stage we have $\mathcal{L}^8$, $\mathcal{A}^7$, $\mathcal{L}^6$, and $\mathcal{A}^5$ as candidate models *irrespective* of the segmentation results. They are related by the following inclusion relationships (the left-hand side of ← is a subspace of the right-hand side):

$$\begin{tikzcd}
& \mathcal{L}^6 \arrow[dl] & \\
\mathcal{L}^8 & & \mathcal{A}^5 \arrow[ul] \arrow[dl] \\
& \mathcal{A}^7 \arrow[ul] &
\end{tikzcd} \tag{6}$$

If the number of independent motions is m, the above $\mathcal{L}^8$, $\mathcal{A}^7$, $\mathcal{L}^6$, and $\mathcal{A}^5$ are replaced by $\mathcal{L}^{4m}$, $\mathcal{A}^{4m-1}$, $\mathcal{L}^{3m}$, and $\mathcal{A}^{3m-1}$, respectively.

5. Model Selection

A naive idea for model selection is to fit the candidate models to the observed data and choose the one for which the *residual*, i.e., the sum of the square distances of the data points to the fitted model, is the smallest. This does not work, however, because the model that has the largest degree of freedom, i.e., the largest number of parameters that can specify the model, always has the smallest residual. It follows that we must balance the increase in the residual against the decrease in the degree of freedom. For this purpose, we use the geometric AIC [11, 12] (see [25, 26] for other criteria).

Let $n = 2M$. For the N trajectory vectors $\{\boldsymbol{p}_\alpha\}$ in an n-dimensional space, define the $n \times n$ *moment matrix* by

$$\boldsymbol{M} = \sum_{\alpha=1}^{N} \boldsymbol{p}_\alpha \boldsymbol{p}_\alpha^\top. \tag{7}$$

Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ be its eigenvalues. If we optimally fit a d-dimensional subspace to $\{\boldsymbol{p}_\alpha\}$, the resulting residual $J_{\mathcal{L}^d}$ is given by

$$J_{\mathcal{L}^d} = \sum_{i=d+1}^{n} \lambda_i. \tag{8}$$

The geometric AIC has the following form [11, 12]:

$$\text{G-AIC}_{\mathcal{L}^d} = J_{\mathcal{L}^d} + 2d(N + n - d)\epsilon^2. \tag{9}$$

Here, ϵ, which we call the *noise level*, is the standard deviation of the noise in the coordinates of the feature points.

For fitting a d-dimensional affine space to $\{\boldsymbol{p}_\alpha\}$, the geometric AIC is computed as follows. Define the $n \times n$ moment matrix matrix by

$$\boldsymbol{M}' = \sum_{\alpha=1}^{N} (\boldsymbol{p}_\alpha - \boldsymbol{p}_C)(\boldsymbol{p}_\alpha - \boldsymbol{p}_C)^\top, \tag{10}$$

[1] The minimal subspace that includes an n_1-dimensional subspace and an n_2-dimensional subspace has dimension $n_1 + n_2$, while the minimal affine space that includes an m_1-dimensional affine space and an m_2-dimensional affine space has dimension $m_1 + m_2 + 1$.

where $\boldsymbol{p}_C$ is the centroid of $\{\boldsymbol{p}_\alpha\}$. Let $\lambda'_1 \geq \lambda'_2 \geq \cdots \geq \lambda'_n$ be the eigenvalues of the matrix $\boldsymbol{M}'$. The residual $J_{\mathcal{A}^d}$ of fitting a d-dimensional affine space to $\{\boldsymbol{p}_\alpha\}$ is given by

$$J_{\mathcal{A}^d} = \sum_{i=d+1}^{n} \lambda'_i. \tag{11}$$

The geometric AIC has the following form [11, 12]:

$$\text{G-AIC}_{\mathcal{A}^d} = J_{\mathcal{A}^d} + 2\Big(dN + (d+1)(n-d)\Big)\epsilon^2. \tag{12}$$

We compare the geometric AIC for each candidate model and choose the one that has the smallest geometric AIC.

6. Trajectory Data Compression

The segmentation procedure involves various vector and matrix computations. The trajectories over M frames are represented by $2M$-dimensional vectors. If, for example, we track 100 feature points over 100 frames, we have 100 200-dimensional vectors as input data. The computation costs increases as the number of frames increases.

However, all the trajectory vectors are constrained to be in a subspace of dimension d, which is determined by the number of independent motions, irrespective of the number M of frames. Usually, d is a very small number.

In the presence of noise, the trajectory vectors may not exactly be in that subspace, but the segmentation computation is done there. This observation suggests that we can represent the trajectories by d-dimensional vectors by projecting them onto that d-dimensional subspace and taking a new coordinate system in such a way that the first d coordinate axes span the d-dimensional subspace. If, for example, one object is moving relative to a stationary scene, all trajectories are represented by 8-dimensional vectors.

This coordinate change is justified, since the subspace separation procedure is based only on the subspace structure of the data, which is invariant to linear transformations of the entire space, resulting in the same segmentation.

The actual procedure goes as follows. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ be the eigenvalues of the matrix $\boldsymbol{M}$ given in eq. (7), and $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n\}$ the orthonormal system of the corresponding eigenvectors. All we need to do is replace the n-dimensional vectors $\boldsymbol{p}_\alpha$ by the d-dimensional vector

$$\tilde{\boldsymbol{p}}_\alpha = \begin{pmatrix} (\boldsymbol{p}_\alpha, \boldsymbol{u}_1) \\ (\boldsymbol{p}_\alpha, \boldsymbol{u}_2) \\ \vdots \\ (\boldsymbol{p}_\alpha, \boldsymbol{u}_d) \end{pmatrix}, \tag{13}$$

where $(\boldsymbol{a}, \boldsymbol{b})$ denotes the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

It seems that we can similarly convert the trajectory vectors $\{\boldsymbol{p}_\alpha\}$ into d-dimensional vectors when a d-dimensional affine space $\mathcal{A}^d$ is chosen as the camera model. Namely, we take a new coordinate system such that its origin is in $\mathcal{A}^d$ and the first d coordinate axes span $\mathcal{A}^d$. The affine space structure should be invariant to such a coordinate change.

However, the affine space separation described in [11, 16] also uses part of the subspace separation procedure as internal auxiliary routines. Since affine transformations destroy the subspace structure, affine coordinate changes are not allowed as long as we use Kanatani's affine space separation.

If a d-dimensional affine space $\mathcal{A}^d$ is chosen, we instead compress $\{\boldsymbol{p}_\alpha\}$ into $(d+1)$-dimensional vectors by projecting them onto the $(d+1)$-dimensional subspace $\mathcal{L}^{d+1}$ in which $\mathcal{A}^d$ is included, and the projecting them onto $\mathcal{A}^d$. The computation of the latter part goes as follows.

We calculate a $(d+1) \times (d+1)$ matrix $\boldsymbol{M}'$ in the same way as eq. (10) in the $(d+1)$-dimensional subspace $\mathcal{L}^{d+1}$. Let $\tilde{\lambda}'_1 \geq \tilde{\lambda}'_2 \geq \cdots \geq \tilde{\lambda}'_d$ be its eigenvalues, and $\{\tilde{\boldsymbol{u}}'_1, \tilde{\boldsymbol{u}}'_2, \ldots, \tilde{\boldsymbol{u}}'_d\}$ the orthonormal system of the corresponding eigenvectors. We compute the $(d+1) \times (d+1)$ projection matrix

$$\tilde{\boldsymbol{P}}'_d = \sum_{i=1}^{d} \tilde{\boldsymbol{u}}'_i \tilde{\boldsymbol{u}}'^{\top}_i, \tag{14}$$

and replace $(d+1)$-dimensional vectors $\tilde{\boldsymbol{p}}_\alpha$ by the following $(d+1)$-dimension vectors:

$$\hat{\boldsymbol{p}}'_\alpha = \tilde{\boldsymbol{p}}_C + \tilde{\boldsymbol{P}}'_d(\tilde{\boldsymbol{p}}_\alpha - \tilde{\boldsymbol{p}}_C). \tag{15}$$

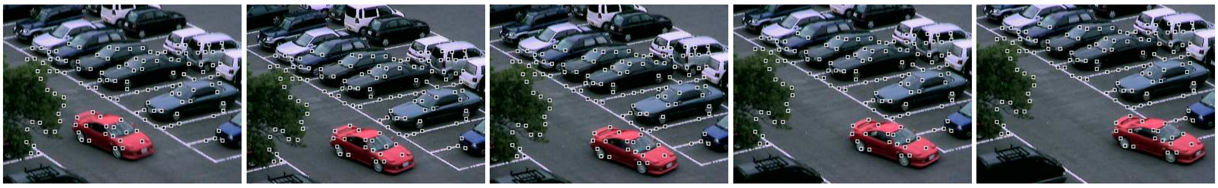
The subspace separation [13, 15] and the affine space separation [16] both internally use model selection by the geometric AIC, which involves the *codimension* of the constraint. If we use the compressed data as input, the codimension should be the difference of the dimension of the constraint not from the original dimension of the data but from their compressed dimension.

7. Summary of the Procedure

Our segmentation procedure is summarized as follows.

1. Detect feature points in the first frame using the Harris operator [4], and track them through the entire video stream using the Kanade-Lucas-Tomasi algorithm [24].
2. Estimate the number of independent motions using the method described in [17, 15][2].

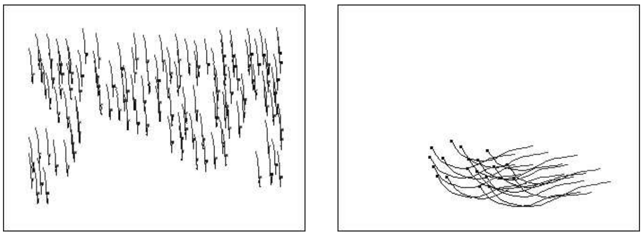
[2] The source program is publicly available at http://www.suri.it.okayama-u.ac.jp/e-program.html.



(a) Input sequence

Model	$\mathcal{L}^8$	$\mathcal{A}^7$	$\mathcal{L}^6$	$\mathcal{A}^5$
G-AIC	836.9	779.1	688.9	**631.1**

(b) Geometric AIC



(c) Background points

(d) Object points

Method	Costeira-Kanade	Ichimura	Shi-Malik	$\mathcal{L}^8$	$\mathcal{A}^7$	$\mathcal{L}^6$	$\mathcal{A}^5$
Correctness (%)	85.3	92.6	86.8	75.0	86.0	97.7	**100**

(e) Correctness of segmentation

Figure 1: (a) Input video sequence (1st, 8th, 15th, 22th, 30th frame) and successfully tracked 136 feature points. (b) The geometric AIC for each model. (c) The segmented trajectories of background points. (d) The segmented trajectories of object points. (e) The correctness of segmentation for different methods.

3. Remove outlier trajectories using the method described in [23][2].
4. Test if the trajectory vectors span a $4m$-dimensional subspace $\mathcal{L}^{4m}$, a $(4m-1)$-dimensional affine space $\mathcal{A}^{4m-1}$, a $3m$-dimensional subspace $\mathcal{L}^{3m}$, or a $(3m-1)$-dimensional affine space $\mathcal{L}^{3m-1}$, using the geometric AIC.
5. Select the model for which the geometric AIC is the smallest.
6. Compress the trajectories into low-dimensional vectors by projecting them onto the subspace defined by the selected model.
7. Do segmentation by subspace separation[2] [13, 15] or by affine space separation[2] [16] according to the selected model.

8. Real Video Experiments

We tested our proposed method using real video sequences. The image size is 320×240 pixels. In order to focus only on the segmentation performance by our proposed model selection, we assumed that the number of independent motions was two in the following examples.

Fig. 1(a) shows five frames decimated from a 30 frame sequence taken by a moving camera. We correctly tracked 136 points, which are indicated by the symbol □ in the images.

We fitted to them an 8-dimensional subspace $\mathcal{L}^8$, a 7-dimensional affine space $\mathcal{A}^7$, a 6-dimensional subspace $\mathcal{L}^6$, and a 5-dimensional affine space $\mathcal{A}^5$ and computed their geometric AICs. Fig. 1(b) shows their values. As we can see, the 5-dimensional affine space $\mathcal{A}^5$ was chosen as the best model.

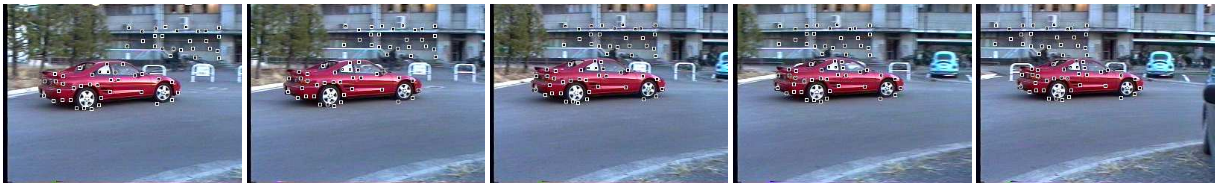
In order to compute the geometric AIC as given in eqs. (9) and (12), we need to know the noise level ϵ. Theoretically, it can be estimated from the residual of the most general model $\mathcal{L}^8$ if the noise in each frame is independent and Gaussian [11]. In reality, however, strong correlations exist over consecutive frames, so that some points are tracked unambiguously throughout the sequence, while others fluctuate from frame to frame [23]. Considering this, we empirically set ϵ to 0.5 pixels[3]. We have confirmed that changing this value over $0.1 \sim 1.0$ does not affect the selected model in this and the subsequent experiments.

The video sequence of Fig. 1(a) was taken from a distance, and the object (a car) and the background are moving almost rigidly in the image. Hence, the selection of $\mathcal{A}^5$ seems reasonable.

Figs. 1(c) and (d) show the trajectories of the object points and the background points segmented by the affine space separation based on the selected model $\mathcal{A}^5$.

Fig. 1(e) compares the correctness of segmentation measured by (the number of correctly classified points)/(the total number of points) in percentage

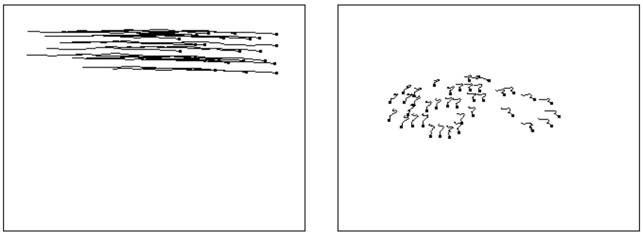
[3] We also used this value for the outlier removal procedure [23].



(a) Input sequence

Model	$\mathcal{L}^8$	$\mathcal{A}^7$	$\mathcal{L}^6$	$\mathcal{A}^5$
G-AIC	434.6	413.6	398.7	**379.8**

(b) Geometric AIC



(c) Background points

(d) Object points

Method	Costeira-Kanade	Ichimura	Shi-Malik	$\mathcal{L}^8$	$\mathcal{A}^7$	$\mathcal{L}^6$	$\mathcal{A}^5$
Correctness (%)	57.1	57.1	57.1	92.0	61.9	61.9	**100**

(e) Correctness of segmentation

Figure 2: (a) Input video sequence (1st, 5th, 9th, 13th, 17th frame) and successfully tracked 63 feature points. (b) The geometric AIC for each model. (c) The segmented trajectories of background points. (d) The segmented trajectories of object points. (e) The correctness of segmentation for different methods.

for different methods. The correctness of individual matches was judged by visual inspection.

In the table, "Costeira-Kanade" means the method of Costeira and Kanade [1], which progressively interchanges the rows and columns of the (*shape*) *interaction matrix* (Appendix A) to make it approximately block-diagonal in such a way that the off-diagonal elements have small absolute values. "Ichimura" means the method of Ichimura [6], who applied the *Otsu discrimination criterion* [20] to each row of the interaction matrix and segmented the elements according to the row with the highest discrimination measure. "Shi-Malik" indicates the result obtained by partitioning the graph defined by the interaction matrix (the feature points as vertices and the absolute values of its elements as the weights of the corresponding edges) in such a way that the *normalized cut* [21] is minimized (Appendix B). The fuzzy clustering of Inoue and Urahama [10] is also based on a similar idea. The symbols $\mathcal{L}^8$, $\mathcal{A}^7$, $\mathcal{L}^6$, and $\mathcal{A}^5$ indicate the subspace separation and affine space separation using the corresponding models. As expected, the affine space separation using the selected model $\mathcal{A}^5$ alone achieved 100% correct segmentation.

Fig. 2(a) shows another video sequence, through which 63 points are tracked over 17 frames. The results are arranged in the same way as Figs. 2(b)–(e). Again, $\mathcal{A}^5$ was chosen as the best model, and the affine space separation using this model alone achieved 100% correct segmentation. This sequence was also taken from a distance, and the object and the background are moving almost rigidly in the image, so the choice of $\mathcal{A}^5$ seems reasonable.

Fig. 3(a) shows a different sequence, through which 73 points are tracked over 100 frames. This sequence was taken near the moving object (a person) by a moving camera, so the perspective effects are relatively strong. As expected, the 8-dimensional subspace $\mathcal{L}^8$ was chosen as the best model, and the subspace separation using it gave the best result.

Table 1 shows the computation time and its reduction ratio brought about by our dimension compression for the above three examples. Here, we converted the trajectories into 8-dimensional vectors, irrespective of the selected model. The reduction ratio is measured by (the computation time for compressed data)/(the computation time for the original data) in percentage.

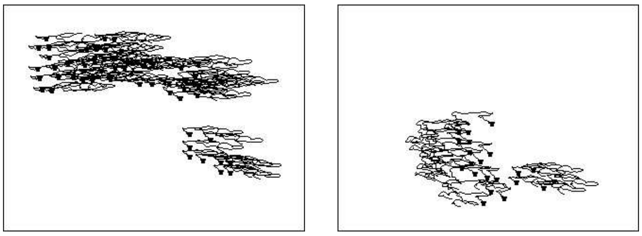
The sequence in Fig. 1 is only 30 frames long, but the number of feature points is very large. In this case, the reduction ratio is only 94.7%. The sequence in Fig. 2 is also short, and the number of feature points is very small. Again, the reduction ratio is merely 71.4%. In contrast, the sequence in Fig. 3 is very long with relatively a small number of feature points. In this case, the computation time is dramatically reduced to 15.2%.

Thus, the reduction of computation time is par-

(a) Input images

Model	$\mathcal{L}^8$	$\mathcal{A}^7$	$\mathcal{L}^6$	$\mathcal{A}^5$
G-AIC	**2117.9**	2281.5	3158.5	3340.1

(b) Geometric AIC



(c) Background points

(d) Object points

Method	Costeira-Kanade	Ichimura	Shi-Malik	$\mathcal{L}^8$	$\mathcal{A}^7$	$\mathcal{L}^6$	$\mathcal{A}^5$
Correctness (%)	76.7	58.9	76.7	**93.1**	60.2	57.5	89.0

(e) correctness

Figure 3: (a) Input video sequence (1st, 25th, 50th, 75th, 100th frame) and successfully tracked 73 feature points. (b) The geometric AIC for each model. (c) The segmented trajectories of background points. (d) The segmented trajectories of object points. (e) The correctness of segmentation for different methods.

Table 1: The computation time and its reduction ratio.

	Fig. 1	Fig. 2	Fig. 3
Number of frames	30	17	100
Number of points	136	63	73
Computation time (sec)	373	5	12
Reduction ratio (%)	94.7	71.4	15.2

ticularly significant for a long sequence. This is because the compressed dimension d of the data depends only on the camera model, irrespective of the number M of the frames. As a result, the computation time is approximately a function of the number N of feature points alone. From Table 1, we can guess that the computation time is approximately $O(N^5)$ or $O(N^6)$, through rigorous analysis is very difficult.

9. Concluding Remarks

We have proposed a technique for automatically selecting the best model by using the geometric AIC in an attempt to improve the segmentation accuracy of the subspace separation [13] and the affine space separation [16] *before* doing segmentation. Using real video sequences, we demonstrated that the separation accuracy indeed improves if the segmentation is based on the selected model.

We also confirmed that we can compress the trajectories into low-dimensional vectors, irrespective of the frame number, by projecting them onto the subspace defined by the selected model. This is very effective in reducing the computation time for long video sequence.

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. 13680432), 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*, **29**-3, 159–179, Sept. 1998.

[2] M. A. Fischler and R. C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Comm. ACM*, **24**-6, 381–395, 1981.

[3] C. W. Gear, Multibody grouping from motion images, *Int. J. Comput. Vision*, **29**-2, 133–150, Aug./Sept. 1998.

[4] C. Harris and M. Stephens, A combined corner and edge detector, *Proc. 4th Alvey Vision Conf.*, Manchester, U.K., pp. 147–151, Aug. 1988.

[5] D. Q. Huynh and A. Heyden, Outlier detection in video sequences under affine projection, *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, Kauai, Hawaii, U.S.A., Vol. 1, pp. 695–701, Dec. 2001.

[6] 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.

[7] 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.

[8] N. Ichimura and N. Ikoma, Filtering and smoothing for motion trajectory of feature point using non-gaussian state space model, *IEICE Trans. Inf. Syst.*, **E84-D**-6, 755–759, June 2001.

[9] N. Ichimura, Stochastic filtering for motion trajectory in image sequences using a Monte Carlo filter with estimation of hyper-parameters, *Proc. 16th Int. Conf. Pattern Recog.*, Quebec City, Canada, Vol. 4, pp. 68–73, Aug. 2002.

[10] 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.

[11] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier Science, Amsterdam, the Netherlands, 1996.

[12] K. Kanatani, Geometric information criterion for model selection, *Int. J. Comput. Vision*, **26**-3, 171–189, 1998.

[13] 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.

[14] K. Kanatani, Model selection for geometric inference, *Proc. 5th Asian Conf. Comput. Vision*, Melbourne, Australia, Vol. 1, pp. xxi–xxxii, Jan. 2002.

[15] K. Kanatani, Motion segmentation by subspace separation: Model selection and reliability evaluation, *Int. J. Image Graphics*, **2**-2, 179–197, April 2002.

[16] K. Kanatani, Evaluation and selection of models for motion segmentation, *Proc. 7th Euro. Conf. Comput. Vision*, Copenhagen, Denmark, June 2002, pp. 335–349.

[17] K. Kanatani and C. Matsunaga, Estimating the number independent motions for multibody segmentation, *Proc. 5th Asian Conf. Comput. Vision*, Melbourne, Australia, Vol. 1, pp. 7-12, Jan. 2002.

[18] A. Maki and C. Wiles, Geotensity constraint for 3D surface reconstruction under multiple light sources, *Proc. 6th Euro. Conf. Comput. Vision*, Dublin, Ireland, Vol. 1, pp. 725–741, June/July 2000.

[19] A. Maki and K. Hattori, Illumination subspace for multibody motion segmentation, *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, Kauai, Hawaii, U.S.A., Vol. 2, pp. 11–17, Dec. 2001.

[20] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans. Sys. Man Cyber.*, **9**-1, 62-66, 1979.

[21] J. Shi and J. Malik, Normalized cuts and image segmentation, *IEEE Trans. Patt. Anal. Machine Intell.*, **22**-8, 888–905, Aug. 2000.

[22] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, Wiley, New York, 1987.

[23] Y. Sugaya and K. Kanatani, Outlier removal for motion tracking by subspace separation, *Proc. 8th Symposium on Sensing via Image Information*, Yokohama, Japan, pp. 603–608, 2002.

[24] 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/`.

[25] P. H. S. Torr, An assignment of information criteria for motion model selection, *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Puerto Rico, pp. 47–53, June 1997.

[26] P. H. Torr and A. Zisserman, Robust detection of degenerate configurations while estimating the fundamental matrix, *Comput. Vision Image Understand.*, **71**-3, 312–333, 1998.

[27] 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.*, Kauai, Hawaii, U.S.A., Vol. 2, pp. 695–701, Dec. 2001.

Appendix A: Interaction Matrix

Consider a set of N points $\{\boldsymbol{p}_\alpha\} \in \mathcal{R}^n$. Suppose each point belongs to one of the m r-dimensional subspaces $\mathcal{L}_i^r$ of $\mathcal{R}^n$, $i = 1, ..., m$, in such a way that each $\mathcal{L}_i^r$ contains more than r points. Let $d = rm$.

Define the $N \times N$ *metric matrix* $\boldsymbol{G} = (G_{\alpha\beta})$ by

$$G_{\alpha\beta} = (\boldsymbol{p}_\alpha, \boldsymbol{p}_\beta). \tag{16}$$

Let $\lambda_1 \geq \cdots \geq \lambda_N$ be its eigenvalues, and $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}$ the orthonormal system of the corresponding eigenvectors. Define the $N \times N$ (*shape*) *interaction matrix* $\boldsymbol{Q}$ by

$$\boldsymbol{Q} = \sum_{i=1}^{d} \boldsymbol{v}_i \boldsymbol{v}_i^\top. \tag{17}$$

Theorem 1 *The $(\alpha\beta)$ element of $\boldsymbol{Q}$ is zero if the αth and βth points belong to different subspaces:*

$$Q_{\alpha\beta} = 0, \qquad \boldsymbol{p}_\alpha \in \mathcal{L}_i^r, \ \ \boldsymbol{p}_\beta \in \mathcal{L}_j^r, \ \ ]i \neq j \tag{18}$$

This theorem, which is the essence of the Costeira-Kanade algorithm [1], is proved as follows. Since N ($> n$) vectors $\{\boldsymbol{p}_\alpha\}$ are linearly dependent, there exist infinitely many sets of numbers $\{c_\alpha\}$, not all zero, such that $\sum_{\alpha=1}^N c_\alpha \boldsymbol{p}_\alpha = \boldsymbol{0}$, but if the points $\{\boldsymbol{p}_\alpha\}$ belong to two separate subspaces $\mathcal{L}_1$ and $\mathcal{L}_2$ such that $\mathcal{L}_1 \oplus \mathcal{L}_2 = \mathcal{R}^n$ ($\oplus$ denotes direct sum), the set of such "annihilating coefficients" $\{c_\alpha\}$ ("null space" to be precise) is generated by those for which $\sum_{\boldsymbol{p}_\alpha \in \mathcal{L}_1} c_\alpha \boldsymbol{p}_\alpha = \boldsymbol{0}$ and those for which $\sum_{\boldsymbol{p}_\alpha \in \mathcal{L}_2} c_\alpha \boldsymbol{p}_\alpha = \boldsymbol{0}$ (A formal proof is given in [13]). This theorem also plays an important role in Kanatani's subspace separation [13, 15] and affine space separation [16].

The eigenvalues and eigenvectors of the metric matrix $\boldsymbol{G}$ can also be obtained by computing the eigenvalues and eigenvectors of the $N \times N$ *moment matrix*

$$\boldsymbol{M} = \sum_{\alpha=1}^{N} \boldsymbol{p}_\alpha \boldsymbol{p}_\alpha^\top. \tag{19}$$

Let $\lambda_1 \geq \cdots \geq \lambda_N$ be its eigenvalues, and $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}$ the orthonormal system of corresponding eigenvectors. The matrices $\boldsymbol{G}$ and $\boldsymbol{M}$ are both positive semi-definite symmetric and of the same rank, sharing the same nonzero eigenvalues. Their eigenvectors for the nonzero eigenvalues are converted to each other in the form

$$\boldsymbol{v}_i = \frac{1}{\sqrt{\lambda_i}} \begin{pmatrix} (\boldsymbol{p}_1, \boldsymbol{u}_i) \\ \vdots \\ (\boldsymbol{p}_N, \boldsymbol{u}_i) \end{pmatrix}, \quad \boldsymbol{u}_i = \frac{1}{\sqrt{\lambda_i}} \sum_{\alpha=1}^{N} v_{i\alpha} \boldsymbol{p}_\alpha, \tag{20}$$

where $v_{i\alpha}$ is the αth component of $\boldsymbol{v}_i$.

A third way[4] is to do the *singular value decomposition* (*SVD*) of the $n \times N$ *observation matrix*

$$\boldsymbol{W} = \begin{pmatrix} \boldsymbol{p}_1 & \cdots & \boldsymbol{p}_N \end{pmatrix}, \tag{21}$$

into the form

$$\boldsymbol{W} = \boldsymbol{U}_{n\times n} \mathrm{diag}(\sigma_1, \sigma_2, ..., \sigma_n) \boldsymbol{V}_{N\times n}^\top, \tag{22}$$

where $\mathrm{diag}(\sigma_1, \sigma_2, ..., \sigma_n)$ denotes the diagonal matrix with the *singular values* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ as its diagonal elements in that order. It is easy to see that $\sigma_1^2, \ldots, \sigma_n^2$ coincide with $\lambda_1, \ldots \lambda_n$ and that $\boldsymbol{V}_{N\times n}$ and $\boldsymbol{U}_{n\times n}$ are, respectively, the $N \times n$ matrix consisting of $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ as its columns and the $n \times n$ matrix consisting of $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ as its columns.

We can choose from among the above three methods the most efficient one[5], which generally depends on the relative magnitude of N and n.

Appendix B: Normalized Cut Minimization

Consider the problem of partitioning a weighted undirected graph with N vertices. Regarding the weight of each edge as the similarity between the two vertices connected by that edge, we want to partition the vertices into two groups A and B in such a way that the similarities between the vertices within each group are large while the similarities between the vertices that belong to different groups are small.

Let $W_{\alpha\beta}$ be the weight of the edge that connects vertices α and β, and d_α $(= \sum_{\beta=1}^{N} W_{\alpha\beta})$ the *degree* of the vertex α, i.e., the sum of the weights of the edges starting from it.

Let x_α be the group indicator of vertex α, taking the value 1 if it belongs to group A and the value 0 if it belongs to group B. Shi and Malik [21] proposed to partition the graph in such a way that the following *normalized cuts* is minimized:

$$\mathrm{Ncut} = \frac{\displaystyle\sum_{x_\alpha=1, x_\beta=-1} W_{\alpha\beta}}{\displaystyle\sum_{x_\kappa=1} d_\kappa} + \frac{\displaystyle\sum_{x_\alpha=-1, x_\beta=1} W_{\alpha\beta}}{\displaystyle\sum_{x_\kappa=-1} d_\kappa}. \tag{23}$$

It appears that in order to reduce the similarity between the two groups one only needs to minimize the *cut*, i.e., the sum of the weights of the edges that connect the two groups. However, this would often result in an unbalanced partitioning such as a single vertex with a small degree forming one group. Eq. (23) is obtained by normalizing the cut by the sum of the similarities within each group, so that the similarities within each group become large while the similarities between the two groups become small.

Shi and Malik [21] showed that the normalized cut can be minimized by the following procedure:

1. Define the $N \times N$ diagonal matrix
$$\boldsymbol{D} = \mathrm{diag}(d_1, ..., d_N). \tag{24}$$
2. Let $\boldsymbol{W}$ be the $N \times N$ matrix whose $(\alpha\beta)$ element is $W_{\alpha\beta}$.
3. Compute the N-dimensional generalized eigenvector $\boldsymbol{y} = (y_1, \ldots, y_N)^\top$ of the generalized eigenvalue problem
$$(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{y} = \lambda \boldsymbol{D}\boldsymbol{y}, \tag{25}$$
for the second smallest generalized eigenvalue.
4. Let $y_{\max}$ and $y_{\min}$ be, respectively, the maximum and the minimum of $y_1, \cdots, y_N$. Divide the interval $[y_{\min}, y_{\max}]$ into an appropriate number of subintervals of equal width. For each dividing point $y*$, let $x_\alpha = 1$ if $y_\alpha > y_*$ and $x_\alpha = -1$ if $y_\alpha \leq y_*$, $\alpha = 1$, ..., N, and compute the normalized cut (23). Do this for all the dividing points and find the value y_* for which the normalized cut is minimized.
5. Return the N-dimensional binary vector $\boldsymbol{x} = (x_1, \ldots, x_N)^\top$ given by that y_*.

Step 3 of the above procedure can be computed as follows: Let

$$\boldsymbol{D}^{-1/2} = \mathrm{diag}(\frac{1}{\sqrt{d_1}}, ..., \frac{1}{\sqrt{d_N}}), \tag{26}$$

and compute the N-dimensional eigenvector $\boldsymbol{z}$ of the $N \times N$ symmetric matrix

$$\boldsymbol{A} = \boldsymbol{D}^{-1/2}(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{D}^{-1/2}, \tag{27}$$

for the second smallest eigenvalue. The vector $\boldsymbol{y}$ is given by multiplying $\boldsymbol{z}$ by the $N \times N$ matrix

$$\boldsymbol{D}^{1/2} = \mathrm{diag}(\sqrt{d_1}, ..., \sqrt{d_N}). \tag{28}$$

Namely, return

$$\boldsymbol{y} = \boldsymbol{D}^{1/2}\boldsymbol{z}. \tag{29}$$

[4] This is the original form described by Costeira and Kanade [1], but their proof is rather difficult to understand. Theoretically, it is more consistent to start from the metric matrix $\boldsymbol{G}$ and regard the SVD as a computational tool.

[5] In theory, the use of SVD should be the most efficient if it is properly implemented.