VIDEO IMAGE SEQUENCE ANALYSIS: ESTIMATING MISSING DATA AND SEGMENTING MULTIPLE MOTIONS

KEINCHI KANATANI AND YASUYUKI SUGYA
Department of Computer Science
Okayama University, Okayama 700-8530 Japan

Abstract. We discuss two issues of video processing based on our recent results: missing data estimation and multiple motion segmentation. We first show that for a rigidly moving scene we can reliably extend interrupted feature point tracking by imposing a geometric constraint based on the affine camera modeling. For scenes of multiple motions, many techniques have been proposed for segmenting moving objects into individual motions. However, many methods perform very poorly for real video sequences. We resolve this mystery by analyzing the geometric structure of the degeneracy of the motion model, which leads to a new segmentation algorithm. We demonstrate its effectiveness, using real video images.

Key words: video processing, feature tracking, missing data estimation, outlier removal, motion segmentation, affine camera model.

16.1. Introduction

Video processing is one of the central topics for media technology today, and tracking feature points through the image sequence is a first step of many applications including 3-D reconstruction. Here, we discuss two issues in this respect based on our recent results (Sugaya and Kanatani, 2004a; Sugaya and Kanatani 2004b).

The first issue is *missing data*: Feature point tracking fails when the points go out of the field of view or behind other objects. Many techniques have been proposed to estimate the missing data (Brandt, 2002; Jacobs, 2001; Saito and Kamijima, 2003; Tomasi and Kanade, 1992), but most of them are based on tentative 3-D reconstruction from sampled frames, assuming that they are correct. Here, we describe a more reliable scheme which integrates extrapolation and outlier removal. The procedure is based on (Sugaya and Kanatani, 2004a).

The second issue is *multiple motion segmentation* for classifying feature

R. Klette et al. (eds.), Geometric Properties for Incomplete Data, 299–315.

point trajectories into independent motions. For this task, too, many techniques have been proposed (Chen and Suter, 2004; Costeira and Kanade, 1998; Gear, 1998; Ichimura, 1999; Ichimura, 2000; Inoue and Urahama, 2001; Kanatani, 2001; Kanatani, 2002a; Kanatani, 2002b; Park *et al.*, 2004; Vidal and Ma, 2004, Vidal and Hartley, 2004; Wu *et al.*, 2001). According to our experiments, however, many methods that exhibit high accuracy in simulations perform rather poorly for real video sequences. We show that this inconsistency is caused by the *degeneracy* of the motion model on which the segmentation is based. This finding leads to a new segmentation algorithm described in (Sugaya and Kanatani, 2004b). We demonstrate its effectiveness, using real video images.

This paper is organized as follows. Section 2 summarizes the geometric constraints. Section 3 describes our outlier removal procedure. Section 4 describes how we extend partial trajectories. In Section 5, we show real video examples of trajectory extension. In Section 6, we describe our principle of multiple-motion segmentation. In Section 7, we analyze the degeneracy of motion model. Section 8 describes our segmentation algorithm. In Section 9, we show real video examples. Section 10 concludes this paper.

16.2. Geometric Constraints

Our method is based on the geometric constraints described in (Chen and Suter, 2004; Debrunner and Ahuja, 1998; Huynh *et al.*, 2003; Irani, 2002; Kanatani, 2001; Kanatani, 2002a; Kanatani, 2002b; Kanatani and Sugaya, 2004; Sugaya and Kanatani, 2002a; Sugaya and Kanatani, 2002b; Sugaya and Kanatani, 2003; Sugaya and Kanatani, 2004a; Sugaya and Kanatani, 2004b). 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. We stack all the coordinates vertically and represent the entire trajectory by the following $2M$-D *trajectory vector*:

$$\boldsymbol{p}_\alpha = \begin{pmatrix} x_{1\alpha} & y_{1\alpha} & x_{2\alpha} & y_{2\alpha} & \cdots & x_{M\alpha} & y_{M\alpha} \end{pmatrix}^\top. \tag{1}$$

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 a reference, relative to which the scene is moving. Consider a 3-D coordinate system fixed to the scene, 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. Let $(a_\alpha, b_\alpha, c_\alpha)$ be the coordinates of the αth point with respect to this coordinate system. Its position with respect to the reference frame at time κ is

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

We assume an *affine camera*, which generalizes orthographic, weak perspective, and paraperspective projections (Kanatani and Sugaya, 2004; Poelman and Kanade, 19): 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-D vector determined by the position and orientation of the camera and its internal parameters at time κ. Substituting Equation (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 κ. From Equation (4), the trajectory vector $\boldsymbol{p}_\alpha$ in Equation (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.

Equation (5) implies that all the trajectories are constrained to be in the *4-D subspace* spanned by $\{\boldsymbol{m}_0, \boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3\}$. In addition, the coefficient of $\boldsymbol{m}_0$ in Equation (5) is identically 1 for all α. This means that the trajectories are in the *3-D affine space* within that 4-D subspace (Kanatani, 2002b).

16.3. Outlier Removal

Before extending partial trajectories, we must remove incorrectly tracked trajectories, or "outliers", from among observed complete trajectories. For this, we adopt the method described in (Sugaya and Kanatani, 2003), which also discusses problems about the approach in (Huynh and Heyden, 2001). Let $n = 2M$, where M is the number of frames, and let $\{\boldsymbol{p}_\alpha\}$, $\alpha = 1, ...,$ N, be the observed complete n-D trajectory vectors. The procedure is as follows (Sugaya and Kanatani, 2003):

1. Randomly choose four vectors $\boldsymbol{q}_1$, $\boldsymbol{q}_2$, $\boldsymbol{q}_3$, and $\boldsymbol{q}_4$ from among $\{\boldsymbol{p}_\alpha\}$.
2. Compute the $n \times n$ (second-order) moment matrix

$$\boldsymbol{M}_3 = \sum_{i=1}^{4} (\boldsymbol{q}_i - \boldsymbol{q}_C)(\boldsymbol{q}_i - \boldsymbol{q}_C)^\top, \tag{6}$$

where $\boldsymbol{q}_C$ is the centroid of $\{\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3, \boldsymbol{q}_4\}$.

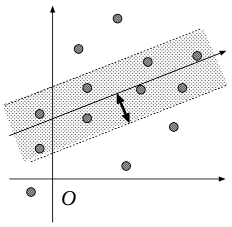


Figure 16.1. Removing outliers by fitting a 3-D affine space.

3. Let $\lambda_1 \geq \lambda_2 \geq \lambda_3$ be the three eigenvalues of the matrix $\boldsymbol{M}_3$, and $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ the orthonormal system of corresponding eigenvectors.
4. Compute the following $n \times n$ projection matrix ($\boldsymbol{I}$ denotes the $n \times n$ unit matrix):
$$\boldsymbol{P}_{n-3} = \boldsymbol{I} - \sum_{i=1}^{3} \boldsymbol{u}_i \boldsymbol{u}_i^\top . \tag{7}$$
5. Let S be the number of points $\boldsymbol{p}_\alpha$ that satisfy
$$\|\boldsymbol{P}_{n-3}(\boldsymbol{p}_\alpha - \boldsymbol{q}_C)\|^2 < (n-3)\sigma^2, \tag{8}$$
where σ is an estimate of the noise standard deviation.
6. Repeat the above procedure a sufficient number of times (we stopped if S did not increase for 200 consecutive iterations), and determine the projection matrix $\boldsymbol{P}_{n-3}$ that maximizes S.
7. Remove those $\boldsymbol{p}_\alpha$ that satisfy
$$\|\boldsymbol{P}_{n-3}(\boldsymbol{p}_\alpha - \boldsymbol{q}_C)\|^2 \geq \sigma^2 \chi^2_{n-3;99}, \tag{9}$$
where $\chi^2_{r;a}$ is the ath percentile of the χ^2 distribution with r degrees of freedom.

The term $\|\boldsymbol{P}_{n-3}(\boldsymbol{p}_\alpha - \boldsymbol{q}_C)\|^2$, called the *residual*, is the squared distance of point $\boldsymbol{p}_\alpha$ from the fitted 3-D affine space. We assume that the noise in the coordinates of the feature points is an independent Gaussian random variable of mean 0 and standard deviation σ. Then, the residual $\|\boldsymbol{P}_{n-3}(\boldsymbol{p}_\alpha - \boldsymbol{q}_C)\|^2$ divided by σ^2 should be subject to a χ^2 distribution with $n-3$ degrees of freedom with expectation $(n-3)\sigma^2$. The above procedure effectively fits a 3-D affine space that maximizes the number of the trajectories whose residuals are smaller than $(n-3)\sigma^2$. Then, we remove those trajectories which cannot be regarded as inliers with significance level 1% (Figure 16.1). We have confirmed that $\sigma = 0.5$ is a reasonable value (Sugaya and Kanatani, 2003).

16.4. Trajectory Extension

After removing outlier trajectories, we optimally fit a 3-D affine space to the resulting inlier trajectories. Let $\{\boldsymbol{p}_\alpha\}$, $\alpha = 1, ..., N$, be their trajectory vectors. We first compute their centroid and the (second-order) moment matrix

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

Let $\lambda_1 \geq \lambda_2 \geq \lambda_3$ be the largest three eigenvalues of the matrix $\boldsymbol{M}$, and $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ the orthonormal system of corresponding eigenvectors. The optimally fitted 3-D affine space is spanned by the three vectors of $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, and $\boldsymbol{u}_3$ starting from $\boldsymbol{p}_C$.

If the αth point can be tracked only over κ of the M frames, its trajectory vector $\boldsymbol{p}_\alpha$ has $n - k$ unknown components ($k = 2\kappa$). We partition the vector $\boldsymbol{p}_\alpha$ into the k-D part $\boldsymbol{p}_\alpha^{(0)}$ consisting of the k known components and the $(n-k)$-D part $\boldsymbol{p}_\alpha^{(1)}$ consisting of the remaining $n - k$ unknown components. Similarly, we partition the centroid $\boldsymbol{p}_C$ and the basis vectors $\{\boldsymbol{u}_1$, $\boldsymbol{u}_2$, $\boldsymbol{u}_3\}$ into the k-D parts $\boldsymbol{p}_C^{(0)}$ and $\{\boldsymbol{u}_1^{(0)}$, $\boldsymbol{u}_2^{(0)}$, $\boldsymbol{u}_3^{(0)}\}$ and the $(n-k)$-D parts $\boldsymbol{p}_C^{(1)}$ and $\{\boldsymbol{u}_1^{(1)}$, $\boldsymbol{u}_2^{(1)}$, $\boldsymbol{u}_3^{(1)}\}$ in accordance with the division of $\boldsymbol{p}_\alpha$.

We first test if each of the partial trajectories is sufficiently reliable. Let $\boldsymbol{p}_\alpha$ be a partial trajectory vector. If image noise does not exist, the deviation of $\boldsymbol{p}_\alpha$ from the centroid $\boldsymbol{p}_C$ should be expressed as a linear combination of $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, and $\boldsymbol{u}_3$. Hence, there should be constants c_1, c_2, and c_3 such that

$$\boldsymbol{p}_\alpha^{(0)} - \boldsymbol{p}_C^{(0)} = c_1\boldsymbol{u}_1^{(0)} + c_2\boldsymbol{u}_2^{(0)} + c_3\boldsymbol{u}^{(0)} \tag{11}$$

for the known part. In the presence of image noise, this equality does not hold. If we let $\boldsymbol{U}^{(0)}$ be the $k \times 3$ matrix consisting of $\boldsymbol{u}_1^{(0)}$, $\boldsymbol{u}_2^{(0)}$, and $\boldsymbol{u}_3^{(0)}$ as its columns, Equation (11) is replaced by

$$\boldsymbol{p}_\alpha^{(0)} - \boldsymbol{p}_C^{(0)} \approx \boldsymbol{U}^{(0)}\boldsymbol{c}, \tag{12}$$

where $\boldsymbol{c}$ is the 3-D vector consisting of c_1, c_2, and c_3. Assuming that $k \geq 3$, we estimate the vector $\boldsymbol{c}$ by least squares in the form

$$\hat{\boldsymbol{c}} = \boldsymbol{U}^{(0)-}(\boldsymbol{p}_\alpha^{(0)} - \boldsymbol{p}_C^{(0)}), \tag{13}$$

where $\boldsymbol{U}^{(0)-}$ is the generalized inverse of $\boldsymbol{U}^{(0)}$. It is computed by

$$\boldsymbol{U}^{(0)-} = (\boldsymbol{U}^{(0)\top}\boldsymbol{U}^{(0)})^{-1}\boldsymbol{U}^{(0)\top}. \tag{14}$$

The residual, i.e., the squared distance of point $\boldsymbol{p}_\alpha^{(0)}$ from the 3-D affine space spanned by $\{\boldsymbol{u}_1^{(0)}, \boldsymbol{u}_2^{(0)}, \boldsymbol{u}_3^{(0)}\}$ is $\|\boldsymbol{p}_\alpha^{(0)} - \boldsymbol{p}_C^{(0)} - \boldsymbol{U}^{(0)}\hat{\boldsymbol{c}}\|^2$. Under our noise model, the residual $\|\boldsymbol{p}_\alpha^{(0)} - \boldsymbol{p}_C^{(0)} - \boldsymbol{U}^{(0)}\hat{\boldsymbol{c}}\|^2$ divided by σ^2 should be subject to a χ^2 distribution with $k-3$ degrees of freedom. Hence, we regard those trajectories that satisfy

$$\|\boldsymbol{p}_\alpha^{(0)} - \boldsymbol{p}_C^{(0)} - \boldsymbol{U}^{(0)}\hat{\boldsymbol{c}}\|^2 \geq \sigma^2\chi^2_{k-3;99} \tag{15}$$

as outliers with significance level 1%.

The unknown part $\boldsymbol{p}_\alpha^{(1)}$ is estimated from the constraint implied by Equation (11), namely

$$\boldsymbol{p}_\alpha^{(1)} - \boldsymbol{p}_C^{(1)} = c_1\boldsymbol{u}_1^{(1)} + c_2\boldsymbol{u}_2^{(1)} + c_3\boldsymbol{u}^{(1)} = \boldsymbol{U}^{(1)}\boldsymbol{c}, \tag{16}$$

where $\boldsymbol{U}^{(1)}$ is the $(n-k)\times 3$ matrix consisting of $\boldsymbol{u}_1^{(1)}$, $\boldsymbol{u}_2^{(1)}$, and $\boldsymbol{u}_3^{(1)}$ as its columns. Substituting Equation (13) for $\boldsymbol{c}$, we obtain

$$\hat{\boldsymbol{p}}_\alpha^{(1)} = \boldsymbol{p}_C^{(1)} + \boldsymbol{U}^{(1)}\boldsymbol{U}^{(0)-}(\boldsymbol{p}_\alpha^{(0)} - \boldsymbol{p}_C^{(0)}). \tag{17}$$

Evidently, this is an optimal estimate in the presence of Gaussian noise. However, the underlying affine space is computed only from a small number of complete trajectories; no information contained in the partial trajectories is used, irrespective of how long they are. So, we also incorporate partial trajectories in the following manner.

Note that if three components of $\boldsymbol{p}_\alpha$ are specified, one can place it, in general, in any 3-D affine space by appropriately adjusting the remaining $n-3$ components. In view of this, we introduce the "weight" of the trajectory vector $\boldsymbol{p}_\alpha$ with k known components in the form

$$W_\alpha = \frac{k-3}{n-3}. \tag{18}$$

Let N be the number of all trajectories, complete or partial, inliers or outliers. The optimization goes as follows:

1. Set the weights W_α of those trajectories, complete or partial, that are so far judged to be outliers to 0. All other weights are set to the value in Equation (18).
2. Fit a 3-D affine space to all the trajectories. The procedure is the same as before except that Equations (10) are replaced by the *weighted* centroid and the *weighted* moment matrix

$$\boldsymbol{p}_C = \frac{\sum_{\alpha=1}^N W_\alpha \boldsymbol{p}_\alpha}{\sum_{\alpha=1}^N W_\alpha}, \qquad \boldsymbol{M} = \sum_{\alpha=1}^N W_\alpha(\boldsymbol{p}_\alpha - \boldsymbol{p}_C)(\boldsymbol{p}_\alpha - \boldsymbol{p}_C)^\top. \tag{19}$$

3. Test each trajectory if it is an outlier, using Equation (15).
4. Estimate the unknown parts of the inlier partial trajectories, using Equation (17).

These four steps are iterated until the fitted affine space converges. In the course of this optimization, trajectories once regarded as outliers may be judged to be inliers later, and vice versa. In the end, inlier partial trajectories are optimally extended with respect to the affine space that is optimally fitted to all the complete and partial inlier trajectories.

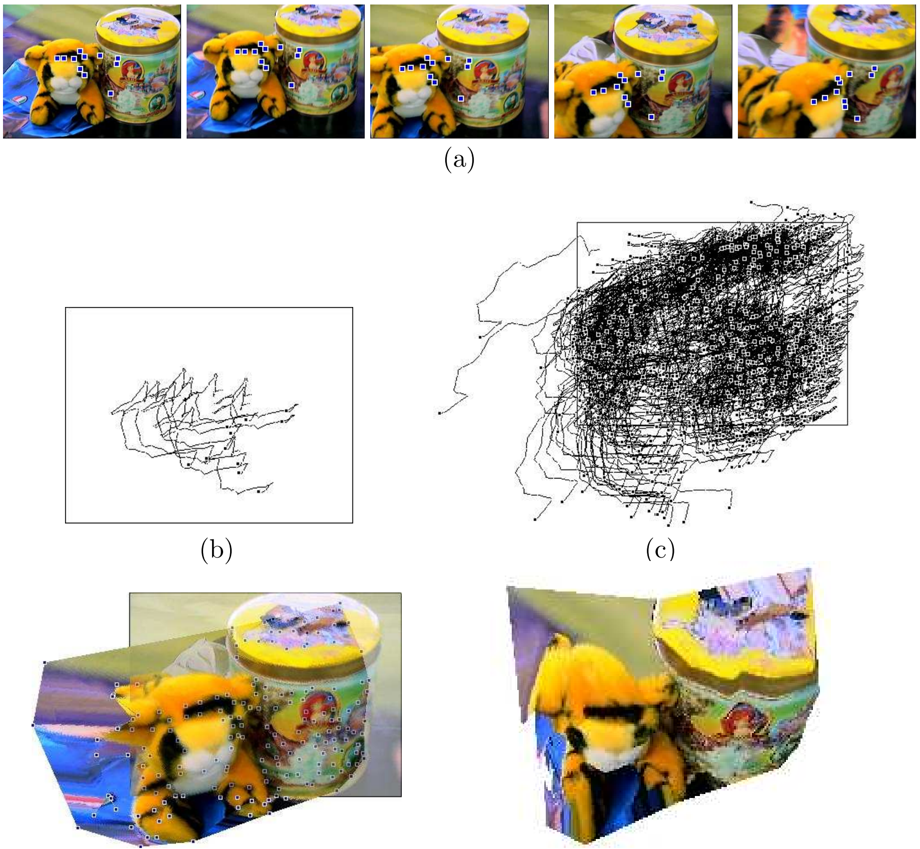
The iterations may not converge if the initial guess is very poor or a large proportion of the trajectories are incorrect. However, this did not happen in any of our experiments using real video sequences.

We need at least three complete trajectories for guessing the initial affine space. If no such trajectories are given, we may use the method of Jacobs (Jacobs, 2001), but it is much more practical to segment the sequence into overlapping blocks, extending partial trajectories over each block, and connecting the blocks.

16.5. Experiments

Figure 16.2(a) shows five decimated frames from a 50 frame sequence (320×240 pixels) of a static scene taken by a moving camera. We detected 200 feature points and tracked them using the Kanade-Lucas-Tomasi algorithm (Tomasi and Kanade, 1991). When tracking failed at some frame, we restarted the tracking after adding a new feature point in that frame. In the end, we obtained 29 complete trajectories, of which 11 are regarded as inliers by the procedure described in Section 3. The marks □ in Figure 16.2(a) indicate their positions; Figure 16.2(b) shows their trajectories.

Using the affine space they define, we extended the partial trajectories and optimized the affine space and the extended trajectories. The optimization converged after 11 iterations, resulting in the 560 inlier trajectories shown in Figure 16.2(c). The computation time for this optimization was 134 seconds. We used Pentium 4 2.4B GHz for the CPU with 1 GB main memory and Linux for the OS. Figure 16.2(d) is the extrapolated image of the 33th frame after missing feature positions are restored: using the 180 feature points visible in the first frame, we defined triangular patches, to which the texture in the first frame is mapped. We reconstructed the 3-D shape by factorization based on weak perspective projection (Kanatani and Sugaya, 2004) (Figure 16.2(e)); see (Sugaya and Kanatani, 2004a) for more experiment results.



(a)

(b)

(c)

(d)

(e)

Figure 16.2. (a) Five decimated frames from a 50 frame sequence and 11 points correctly tracked throughout the sequence. (b) The 11 complete inlier trajectories. (c) The 560 optimal extensions of the trajectories. (d) The extrapolated texture-mapped image of the 33th frame. (e) The reconstructed 3-D shape.

16.6. Multiple Motion Segmentation

So far, we have regarded the observed trajectories as points undergoing a single rigid motion. We now consider the case in which multiple motions exist.

Equation (5) states that the trajectory vectors of points that belong to one object are constrained to be in the 4-D subspace spanned by $\{\boldsymbol{m}_0, \boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3\}$. Hence, 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* (Kanatani, 2001; Kanatani, 2002a).

Equation (5) also states that the trajectory vectors of points that belong to one object are constrained to be in a 3-D affine space within that 4-D subspace. Hence, 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* (Kanatani, 2002b).

Theoretically, the segmentation accuracy should be higher if we use stronger constraints. For real video sequences, however, we have found that the affine space separation accuracy is often lower than that of the subspace separation (Sugaya and Kanatani, 2002a; Sugaya and Kanatani, 2002b). We will resolve this inconsistency in shortly.

As in the case of a single motion, we first need to remove outlier trajectories. If the trajectories were segmented into individual classes, we could apply the method of Section 3 to each motion separately. 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. We observed that all apparent outliers were removed by this method, although some inliers were also removed for safety (Sugaya and Kanatani, 2003).

16.7. 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 Equation (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 Equation (4) vanishes; the scale changes 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 Equation (5) belongs to the *2-D affine space* passing through $\boldsymbol{m}_0$ and spanned by $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ (Sugaya and Kanatani, 2002a; Sugaya and Kanatani, 2002b).

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 Equation (2) to be in the X and Y directions,

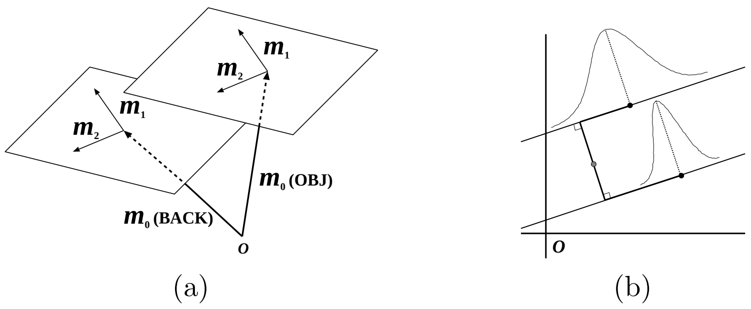


Figure 16.3. (a) If the motions of the objects and the background are degenerate, their trajectory vectors belong to mutually parallel 2-D planes. (b) The data distributions inside the individual 2-D planes are modeled by Gaussian distributions.

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 Equation (5) are also common. Hence, the 2-D affine spaces, or planes, of all the motions are *parallel* (Sugaya and Kanatani, 2004b) (Figure 16.3(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.

16.8. Degeneracy-tuned Learning

We now describe a learning procedure tuned to the parallel 2-D plane degeneracy (Sugaya and Kanatani, 2004b). First, we model the data distributions inside the individual 2-D planes by Gaussian distributions (Figure 16.3(b)). As before, we let $n = 2M$. Suppose N n-D 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

$$W_\alpha^{(k)} = \begin{cases} 1 & \text{if } \boldsymbol{p}_\alpha \text{ belongs to class } k \\ 0 & \text{otherwise} \end{cases}. \tag{20}$$

Then, we iterate the following procedures A, B, and C in turn until all the weights $\{W_\alpha^{(k)}\}$ converge (we stopped the iterations when the increments in $W_\alpha^{(k)}$ are all smaller than 10^{-10}).

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:

$$w^{(k)} = \frac{1}{N}\sum_{\alpha=1}^{N} W_\alpha^{(k)}, \qquad \boldsymbol{p}_C^{(k)} = \frac{\sum_{\alpha=1}^{N} W_\alpha^{(k)}\boldsymbol{p}_\alpha}{\sum_{\alpha=1}^{N} W_\alpha^{(k)}}. \tag{21}$$

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

$$\boldsymbol{M}^{(k)} = \frac{\sum_{\alpha=1}^{N} W_\alpha^{(k)}(\boldsymbol{p}_\alpha - \boldsymbol{p}_C^{(k)})(\boldsymbol{p}_\alpha - \boldsymbol{p}_C^{(k)})^\top}{\sum_{\alpha=1}^{N} W_\alpha^{(k)}}. \tag{22}$$

B. Do the following computation.

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

$$\boldsymbol{M} = \sum_{k=1}^{m} w^{(k)}\boldsymbol{M}^{(k)}. \tag{23}$$

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:

$$\boldsymbol{P} = \sum_{i=1}^{2} \boldsymbol{u}_i\boldsymbol{u}_i^\top, \qquad \boldsymbol{P}_\perp = \boldsymbol{I} - \boldsymbol{P}. \tag{24}$$

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

$$\hat{\sigma}^2 = \max[\frac{\mathrm{tr}[\boldsymbol{P}_\perp\boldsymbol{M}\boldsymbol{P}_\perp]}{n-2}, \sigma^2], \tag{25}$$

where $\mathrm{tr}[\,\cdot\,]$ denotes the trace and σ is an estimate of the tracking accuracy. As before we used the value $\sigma = 0.5$ (pixels).
5. Compute the $n \times n$ covariance matrix of the class k by

$$\boldsymbol{V}^{(k)} = \boldsymbol{P}\boldsymbol{M}^{(k)}\boldsymbol{P} + \hat{\sigma}^2\boldsymbol{P}_\perp. \tag{26}$$

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

$$P(\alpha|k) = \frac{e^{-(\boldsymbol{p}_\alpha - \boldsymbol{p}_C^{(k)}, \boldsymbol{V}^{(k)-1}(\boldsymbol{p}_\alpha - \boldsymbol{p}_C^{(k)}))/2}}{\sqrt{\det \boldsymbol{V}^{(k)}}}. \tag{27}$$

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

$$W_\alpha^{(k)} = \frac{w^{(k)} P(\alpha|k)}{\sum_{l=1}^{m} w^{(l)} P(\alpha|l)}. \tag{28}$$

After the iterations of A, B, and C have converged, the αth 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 Equations (21)) as the *a priori probability* of the class k, we compute the probability $P(\alpha|k)$ of the trajectory vector $\boldsymbol{p}_\alpha$ conditioned to be in the class k (Equation (27); common multipliers that will cancel out in Equation (28) are omitted). Then, we apply *Bayes' theorem* (Equation (28)) to compute the *a posterior probability* $W_\alpha^{(k)}$, according which all the trajectories are reclassified. Note that $W_\alpha^{(k)}$ is generally a fraction, so one trajectory belongs to multiple classes with fractional weights until the final classification is made.

This type of *unsupervised learning* (Schlesinger, 1968; Schlesinger and Hlaváč, 2002) (mathematically equivalent to the *EM algorithm* (Dempster *et al.*, 1977)) is widely used for clustering. However, 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.

16.9. 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 (Sugaya and Kanatani, 2004b). 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 method (Kanatani, 2002), which combines the shape interaction matrix of Costeira and Kanade (Costeira and Kanade, 1998), model selection by the geometric AIC (Kanatani, 1998), and robust estimation by LMedS (Rousseeuw and Leroy, 1987). Then, we optimize the resulting 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; see (Sugaya and Kanatani, 2004b) for the actual procedure.

Here, 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 (Costeira and Kanade, 1998; Gear, 1998; Inoue and Urahama, 2001), but none of them seems successful enough. This is because the number of motions is *not well-defined* (Kanatani, 2002a): 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 (Kanatani, 1998) and the geometric MDL (Kanatani, 2004) is used, the resulting number of motions depends on criteria (Kanatani, 2002a). In order to determine the number m of motions, one needs high-level processing using color, shape, and other information.

16.10. Real Video Experiments

Figure 16.4 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 (Tomasi and Kanade, 1991). The marks □ indicate their positions.

Table 16.1 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 (Sugaya and Kanatani, 2002a). We used Pentium 4 2.4GHz for the CPU with

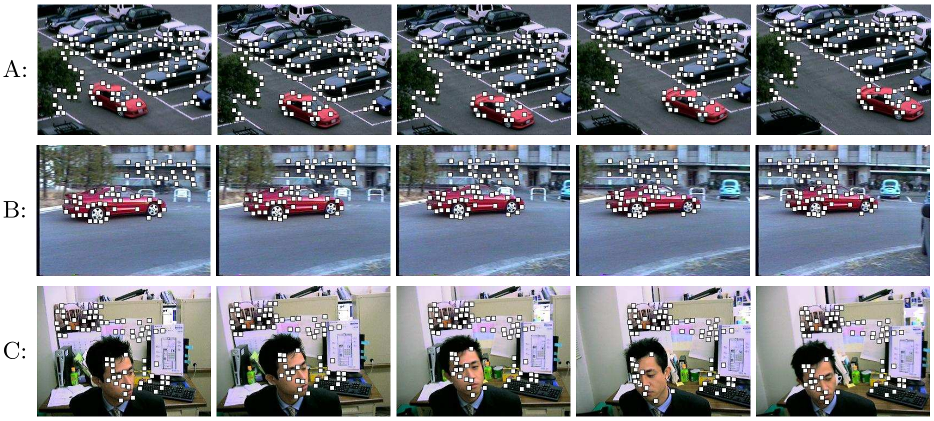


Figure 16.4. Three video sequences and successfully tracked feature points.

TABLE 16.1. The computation time for the multi-stage learning of the sequences in Figure 16.4.

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

1GB main memory and Linux for the OS. Table 16.2 lists the accuracies of different methods ("opt" stands for "optimized") measured by (the number of correctly classified points)/(the total number of points) in percentage.

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. For all the three sequences, our multi-stage learning achieves 100% accuracy.

16.11. Concluding Remarks

We discussed two issues of video processing, missing data estimation and multiple motion segmentation, based on our recent results (Sugaya and Kanatani, 2004a; Sugaya and Kanatani, 2004a; Sugaya and Kanatani, 2004b).

TABLE 16.2. Segmentation accuracy (%) for the sequences in Figure 16.4.

	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**

First, we described our method for extending interrupted feature point tracking (Sugaya and Kanatani, 2004a). We alternate optimal extension of the trajectories and optimal estimation of the affine space. To increase robustness, we test the reliability of the extended trajectories in every step and remove those judged to be outliers.

Next, we studied multiple motion segmentation. Our analysis of the geometric structure of the degeneracy of the motion model leads to a special type of degeneracy, which results in the multi-stage learning scheme described in (Sugaya and Kanatani, 2004b). We demonstrated its effectiveness, using real video images.

The source codes of the programs we used are available at:
`http://www.suri.it.okayama-u.ac.jp/e-program.html`.

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).

References

Brandt, S.: Closed-form solutions for affine reconstruction under missing data, In Proc. *Statistical Methods in Video Processing*. pages 109–114, 2002.

Chen, P. and D. Suter: Recovering the missing components in a large noisy low-rank matrix: Application to SFM. *IEEE Trans. Pattern Analysis Machine Intelligence*, **26**:1051–1063, 2004

Costeira, J. P. and T. Kanade: A multibody factorization method for independently moving objects, *Int. J. Computer Vision*, **29**:159–179, 1998.

Debrunner, C. and N. Ahuja: Segmentation and factorization-based motion and structure estimation for long image sequences. *IEEE Trans. Pattern Analysis Machine Intelligence*, **20**:206–211, 1998.

Dempster, A. P., N.M. Laird and D.B. Rubin: Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc.*, **B39**:1–38, 1977.

Gear, C. W.: Multibody grouping from motion images. *Int. J. Computer Vision*, **29**:133–150, 1998.

Huynh, D. Q. and A. Heyden: Outlier detection in video sequences under affine projection, In Proc. *IEEE Conf. Computer Vision Pattern Recognition*, volume 1, pages 695–701, 2001.

Huynh, D. Q., R. Hartley, and H. Heyden: Outlier correction in image sequences for the affine camera. In Proc. *Int. Conf. Computer Vision*, volume 1, pages 585-590, 2003.

Ichimura, N.: Motion segmentation based on factorization method and discriminant criterion. In Proc. *IEEE Int. Conf. Computer Vision*, volume 1, pages 600–605, 1999.

Ichimura, N.: Motion segmentation using feature selection and subspace method based on shape space. In Proc. *IEEE Int. Conf. Pattern Recognition*, volume 3, pages 858–864, 2000.

Inoue, K. and K. Urahama: Separation of multiple objects in motion images by clustering. In Proc. *IEEE Int. Conf. Computer Vision*, volume 1, pages 219–224, 2001.

Irani, M.: Multi-frame correspondence estimation using subspace constraints. *Int. J. Computer Vision*, **48**:173–194, 2002.

Jacobs, D. W.: Linear fitting with missing data for structure-from-motion. *Computer Vision Image Understand.*, **82**:57–81, 2001.

Kanatani, K.: Geometric information criterion for model selection. *Int. J. Computer Vision*, **26**:171–189, 1998.

Kanatani, K.: Motion segmentation by subspace separation and model selection. In Proc. *IEEE Int. Conf. Computer Vision*. volume 2, pages 301-306, 2001.

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

Kanatani, K.: Evaluation and selection of models for motion segmentation. In Proc. *European Conf. Computer Vision* (A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors), pages 335–349, LNCS 2352, Springer, Berlin, 2002.

Kanatani, K.: Uncertainty modeling and model selection for geometric inference. *IEEE Trans. Pattern Analysis Mach. Intelligence*, **26**:1307–1319, 2004.

Kanatani, K. and Y. Sugaya: Factorization without factorization: Complete recipe. *Memoirs of the Faculty of Engineering, Okayama University*, **38**:61–72, 2004.

Park, J., H. Zha and R. Kasturi: Spectral clustering for robust motion segmentation. In Proc. *European Conf. Computer Vision* (T. Pajdla and J. Matas, editors), pages 391–401, LNCS 3024, Springer, Berlin, 2004.

Poelman, C. J. and T. Kanade: A paraperspective factorization method for shape and motion recovery. *IEEE Trans. Pattern Analysis Mach. Intelligence*, **19**:206–218, 1997.

Rousseeuw, P. J. and A.M. Leroy: *Robust Regression and Outlier Detection*, J. Wiley and Sons, New York, 1987.

Saito, H. and S. Kamijima: Factorization method using interpolated feature tracking via projective geometry. In Proc. *British Machine Vision Conf.* (R. Harvey and A. Bangham, editors), volume 2, pages 449–458, 2003.

Schlesinger, M. I.: A connection between supervised and unsupervised learning in pattern recognition. *Kibernatika*, **2**:81–88, 1968.

Schlesinger, M. I., and V. Hlaváč: *Ten Lectures on Statistical and Structural Pattern Recognition*, Kluwer Academic Publishers, 2002.

Sugaya, Y. and K. Kanatani: Automatic camera model selection for multibody motion segmentation. In Proc. *Workshop on Science of Computer Vision*, pages 31–39, 2002.

Sugaya, Y. and K. Kanatani: Automatic camera model selection for multibody motion segmentation. *IAPR Workshop on Machine Vision Applications*, pages 412–415, 2002.

Sugaya, Y. and K. Kanatani: Outlier removal for motion tracking by subspace separation. *IEICE Trans. Inf. & Syst.*, **E86-D**:1095–1102, 2003.

Sugaya, Y. and K. Kanatani: Extending interrupted feature point tracking for 3-D affine reconstruction. *IEICE Trans. Inf. & Syst.*, **E87-D**:1031–1033, 2004.

Sugaya, Y. and K. Kanatani: Multi-stage optimization for multi-body motion segmentation. *IEICE Trans. Inf. & Syst.*, **E87-D**:1935–1942, 2004.

Tomasi, C. and T. Kanade: Detection and tracking of point features. Tech. Rep. CMU-CS-91-132, Pittsburgh, 1991.

Tomasi, C. and T. Kanade: Shape and motion from image streams under orthography—A factorization method. *Int. J. Computer Vision*, **9**:137–154, 1992.

Vidal R. and Y. Ma: A unified algebraic approach to 2-D and 3-D motion segmentation. In Proc. *European Conf. Computer Vision* (T. Pajdla and J. Matas, editors), pages 1–15, LNCS 3021, Springer, Berlin, 2004.

Vidal, R. and R. Hartley: Motion segmentation with missing data using PowerFactorization and GPCA. In Proc. *IEEE Conf. Computer Vision Pattern Recognition*, volume 2, pages 310-316, 2004.

Wu, Y., Z. Zhang, T.S. Huang and J.Y. Lin: Multibody grouping via orthogonal subspace decomposition, sequences under affine projection. In Proc. *IEEE Conf. Computer Vision Pattern Recognition*, volume 2, pages 695–701, 2001.