

Research Paper

Incorrect Feature Tracking Detection by Affine Space Fitting

CHIKA TAKADA^{†1} and YASUYUKI SUGAYA^{†1}

We present a new method for detecting incorrect feature point tracking. In this paper, we detect incorrect feature point tracking by imposing the constraint that under the affine camera model feature trajectories should be in an affine space in the parameter space. Introducing a statistical model of image noise, we test detected partial trajectories are sufficiently reliable. Then we detect incorrect partial trajectories. Using real video images, we demonstrate that our proposed method can detect incorrect feature point tracking fairly well.

1. Introduction

Extracting feature points from a video sequence and tracking them is the first step of many computer vision applications including structure from motion¹⁸⁾, and motion segmentation^{9)–11),14),17)}. Many authors use the Kanade-Lucas-Tomasi algorithm¹⁹⁾. However, the resulting trajectories are not always correct. In order to improve the tracking, Ichimura and Ikoma⁶⁾ and Ichimura⁵⁾ introduced nonlinear filtering. Hyunh and Heyden⁴⁾, motivated by 3-D reconstruction applications, showed that outlier trajectories in an image sequence of a static scene taken by a moving camera can be removed by fitting a 4-dimensional subspace to them by LMedS. Sugaya and Kanatani¹⁵⁾ fitted a 4-dimensional subspace to the observed trajectories by RANSAC^{2),3)} and removed outliers using a χ^2 criterion by observing the error behavior of actual video tracking.

Usually, we simply discard detected outliers. However, outlier trajectories may partially contain correctly tracked data. **Figure 1** shows three examples of such trajectories. In Fig. 1 (a), the tracking fails and strays after that. In Fig. 1 (b), the tracking returns to a correct path after failing. In Fig. 1 (c), the tracking fails and follows another path.

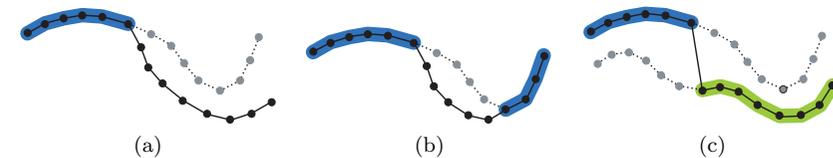


Fig. 1 Outlier trajectories which contain partially correct tracking. Solid lines are for trajectories. Dotted lines are for correct trajectories. (a) Tracking fails and strays after that. (b) Tracking returns to a correct path after failing. (c) Tracking fails and follows another path.

and start tracking another path. If we detect incorrect paths from such outlier trajectories, we can estimate their correct paths from correct partial trajectories, and re-use such corrected trajectories as inliers.

Many techniques have proposed in the past for interpolating missing parts of tracking data. Saito and Kamijima¹³⁾ projectively reconstructed tentative 3-D positions of the missing points by sampling two frames in which they are visible and then reprojected them onto the frames in which they are invisible. Sugaya and Kanatani¹⁶⁾ extended partial trajectories by imposing the constraint that under the affine camera model all feature trajectories should be in an affine space. This is called the *affine space constraint*¹¹⁾. This method consists of iterations for optimally extending the trajectories and for optimally estimating the affine space.

In this paper, we present a new method for detecting incorrect parts in outlier trajectories by imposing the affine space constraint. We first detect outlier trajectories from among complete trajectories. Next, we evaluate the reliability of partial trajectories of outlier trajectories by considering the error behavior of video tracking and regard unreliable parts as incorrect tracking data.

Section 2 summarizes the affine space constraint. Section 3 describes the outlier removal procedure. Section 4 describes how to detect incorrect feature tracking. In Section 5, we describe a method for detecting the longest correct partial trajectories by RANSAC. In Section 6, we show simulation experiment, and in Section 7, we show real video examples and demonstrate that our method works fairly well. Section 8 is our conclusion.

^{†1} Toyohashi University of Technology

2. Affine Space Constraint

We summarize the geometric constraints on which our method is based. The same constraints have already been used in Refs. 9)–11), 15). We reiterate them here, because they play a fundamental role in our method.

2.1 Trajectory of Feature Points

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

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

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

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

2.2 Affine Camera Model

If an affine camera model (generalizing orthographic, weak perspective, and paraperspective projections¹²⁾) is assumed, the image position of $\mathbf{r}_{\kappa\alpha}$ is

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

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

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

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

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

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

2.3 Affine Space Constraint

Equation (5) implies that all the trajectories are constrained to be in the 4-dimensional subspace spanned by $\{\mathbf{m}_0, \mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3\}$ in \mathcal{R}^{2M} . This is called the *subspace constraint*^{9),10)}, on which the method of Jacobs⁷⁾ is based.

In addition, the coefficient of \mathbf{m}_0 in Eq. (5) is identically 1 for all α . This means that the trajectories are in the 3-dimensional affine space within that 4-dimensional subspace. This is called the *affine space constraint*¹¹⁾.

If all the feature points are tracked to the final frame, we can translate the coordinate system so that its origin is at the centroid of the trajectory vectors $\{\mathbf{p}_\alpha\}$. Then, the trajectory vectors are constrained to be in a 3-dimensional subspace in \mathcal{R}^{2M} . The Tomasi-Kanade factorization¹⁸⁾ is based on this representation, and Brandt¹⁾ tried to find this representation by iterations. In this paper, we directly use the affine space constraint without searching for the centroid.

3. Outlier Removal

In order to locate incorrect tracking data in feature trajectories, we first detect incorrect trajectories, or “outliers”, from among completely tracked trajectories.

Sugaya and Kanatani¹⁵⁾ fitted a 4-dimensional subspace to the observed trajectories by RANSAC^{2),3)} and detected outliers using a χ^2 criterion by observing the error behavior of actual video tracking. They also modified their method specifically for the affine space constraint¹⁵⁾. Our method is a direct consequence of the principle given in Ref. 15), but we describe it here, because it plays a crucial role for our method we introduce later.

3.1 Procedure

Let $n = 2M$, where M is the number of frames, and let $\{\mathbf{p}_\alpha\}$, $\alpha = 1, \dots, N$, be the observed complete trajectory vectors. Our outlier detection procedure goes as follows:

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

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

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

- (3) Let $\lambda_1 \geq \lambda_2 \geq \lambda_3$ be the three eigenvalues of the matrix \mathbf{M}_3 , and $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ the orthonormal system of corresponding eigenvectors.
- (4) Compute the $n \times n$ projection matrix

$$\mathbf{P}_{n-3} = \mathbf{I} - \sum_{i=1}^3 \mathbf{u}_i \mathbf{u}_i^\top. \quad (7)$$

- (5) Let S be the number of points \mathbf{p}_α that satisfy

$$\|\mathbf{P}_{n-3}(\mathbf{p}_\alpha - \mathbf{q}_C)\|^2 < (n-3)\sigma^2, \quad (8)$$

where σ is an estimate of the noise standard deviation.

- (6) Repeat the above procedure a sufficient number of times^{*1}, and determine the projection matrix \mathbf{P}_{n-3} that maximizes S .
- (7) Detect those \mathbf{p}_α that satisfy

$$\|\mathbf{P}_{n-3}(\mathbf{p}_\alpha - \mathbf{q}_C)\|^2 \geq \sigma^2 \chi_{n-3;99}^2, \quad (9)$$

where $\chi_{r;a}^2$ is the a th percentile of the χ^2 distribution with r degrees of freedom.

The term $\|\mathbf{P}_{n-3}(\mathbf{p}_\alpha - \mathbf{q}_C)\|^2$, which we call the *residual*, is the squared distance of point \mathbf{p}_α from the fitted 3-dimensional affine space. We regard the uncertainty feature tracking as “noise”. If the noise in the coordinates of the feature points is an independent Gaussian random variable of mean 0 and standard deviation σ , the residual $\|\mathbf{P}_{n-3}(\mathbf{p}_\alpha - \mathbf{q}_C)\|^2$ divided by σ^2 should be subject to a χ^2 distribution with $n-3$ degrees of freedom. Hence, its expectation is $(n-3)\sigma^2$. The above procedure effectively fits a 3-dimensional affine space that maximizes the number of the trajectories whose residuals are smaller than $(n-3)\sigma^2$. After fitting such an affine space, we detect those trajectories which cannot be regarded as inliers with significance level 1%. In Ref. 15), the value $\sigma = 0.5$ is recommended for KLT tracking.

*1 In our experiment, we stopped if S did not increase for 200 consecutive iterations.

3.2 Final Affine Space Fitting

After removing outlier trajectories, we optimally fit a 3-dimensional affine space to the resulting inlier trajectories. Let $\{\mathbf{p}_\alpha\}$, $\alpha = 1, \dots, \tilde{N}$, be their trajectory vectors. We first compute their centroid

$$\mathbf{p}_C = \frac{1}{\tilde{N}} \sum_{\alpha=1}^{\tilde{N}} \mathbf{p}_\alpha. \quad (10)$$

Then, we compute the $n \times n$ moment matrix

$$\mathbf{M} = \sum_{\alpha=1}^{\tilde{N}} (\mathbf{p}_\alpha - \mathbf{p}_C)(\mathbf{p}_\alpha - \mathbf{p}_C)^\top. \quad (11)$$

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

4. Reliability Test

4.1 Partial Trajectories

We assume that the α th feature point is correctly tracked only over κ of the M frames. Its trajectory vector \mathbf{p}_α has $n-k$ incorrect components (we put $n = 2M$ as before and put $k = 2\kappa$). We partition the vector \mathbf{p}_α into the k -dimensional part $\mathbf{p}_\alpha^{(0)}$ consisting of the k correct components and the $(n-k)$ -dimensional part $\mathbf{p}_\alpha^{(1)}$ consisting of the remaining $n-k$ incorrect components. Similarly, we partition^{*2} the centroid \mathbf{p}_C and the basis vectors $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ into the k -dimensional parts $\mathbf{p}_C^{(0)}$ and $\{\mathbf{u}_1^{(0)}, \mathbf{u}_2^{(0)}, \mathbf{u}_3^{(0)}\}$ and the $(n-k)$ -dimensional parts $\mathbf{p}_C^{(1)}$ and $\{\mathbf{u}_1^{(1)}, \mathbf{u}_2^{(1)}, \mathbf{u}_3^{(1)}\}$ in accordance with the division of \mathbf{p}_α .

4.2 Reliability of Partial Trajectory

We test if each of the partial trajectories is sufficiently reliable. Let \mathbf{p}_α be a partial trajectory vector. If noise does not exist, the deviation of \mathbf{p}_α from the centroid \mathbf{p}_C should be expressed as a linear combination of \mathbf{u}_1 , \mathbf{u}_2 , and \mathbf{u}_3 .

*2 This is merely for the convenience of description. In real computation, we treat all data as n -dimensional vectors after multiplying them by an appropriate diagonal matrix consisting of 1s and 0s.

Hence, there should be constants c_1 , c_2 , and c_3 such that

$$\mathbf{p}_\alpha^{(0)} - \mathbf{p}_C^{(0)} = c_1 \mathbf{u}_1^{(0)} + c_2 \mathbf{u}_2^{(0)} + c_3 \mathbf{u}_3^{(0)} \quad (12)$$

for the correct part. In the presence of noise, this equality does not hold. If we let $\mathbf{U}^{(0)}$ be the $k \times 3$ matrix consisting of $\mathbf{u}_1^{(0)}$, $\mathbf{u}_2^{(0)}$, and $\mathbf{u}_3^{(0)}$ as its columns, Eq. (12) is replaced by

$$\mathbf{p}_\alpha^{(0)} - \mathbf{p}_C^{(0)} \approx \mathbf{U}^{(0)} \mathbf{c}, \quad (13)$$

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

$$\hat{\mathbf{c}} = \mathbf{U}^{(0)-} (\mathbf{p}_\alpha^{(0)} - \mathbf{p}_C^{(0)}), \quad (14)$$

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

$$\mathbf{U}^{(0)-} = (\mathbf{U}^{(0)\top} \mathbf{U}^{(0)})^{-1} \mathbf{U}^{(0)\top}. \quad (15)$$

The residual, i.e., the squared distance of point $\mathbf{p}_\alpha^{(0)}$ from the 3-dimensional affine space spanned by $\{\mathbf{u}_1^{(0)}, \mathbf{u}_2^{(0)}, \mathbf{u}_3^{(0)}\}$ is $\|\mathbf{p}_\alpha^{(0)} - \mathbf{p}_C^{(0)} - \mathbf{U}^{(0)} \hat{\mathbf{c}}\|^2$. If the noise in the coordinates of the feature points is an independent Gaussian random variable of mean 0 and standard deviation σ , the residual $\|\mathbf{p}_\alpha^{(0)} - \mathbf{p}_C^{(0)} - \mathbf{U}^{(0)} \hat{\mathbf{c}}\|^2$ 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

$$\|\mathbf{p}_\alpha^{(0)} - \mathbf{p}_C^{(0)} - \mathbf{U}^{(0)} \hat{\mathbf{c}}\|^2 \geq \sigma^2 \chi_{k-3;99}^2 \quad (16)$$

as outliers with significance level 1%.

5. Incorrect Feature Tracking Detection

5.1 Basic Algorithm

If a partial trajectory contains incorrectly tracked data, its residual from the fitted affine space becomes large. So, we can detect such partial trajectories by the reliability test of Eq. (16). Given a complete trajectory, we first choose the feature point in the 1st frame as the base point and generate a partial trajectory by adding the feature point in the 2nd frame. Then, we test its reliability by Eq. (16) (Fig. 2(a)). If the partial trajectory is judged to be reliable, we add

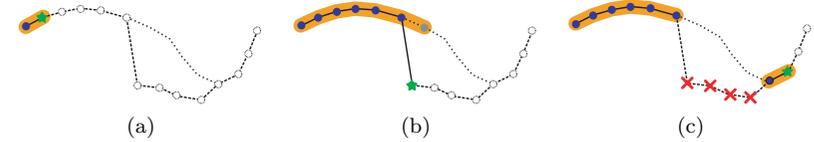


Fig. 2 Detection algorithm. Solid line is for a testing partial trajectory. Dashed line is for a really tracked trajectory. Dotted line is for an ideally correct trajectory. (a) Reliability test for the partial trajectory consisting of the image coordinates of the 1st and 2nd frames (\bullet for inlier frame; \star for testing frame). (b), (c) Reliability test for a partial trajectory consisting of correctly tracked feature positions (\times for outlier frame).

the feature point in the 3rd frame and do the reliability test again. We repeat this until the partial trajectory is judged to be unreliable (Fig. 2(b)). If the partial trajectory is judged to be unreliable, we remove the added point. For the partial trajectory judged to be reliable, we add the next point and test its reliability. Repeating this for all the frames, we can detect incorrectly tracked data (Fig. 2(c)).

The above procedure for detecting incorrect feature tracking is summarized as follows:

Affine space fitting:

- (1) Detect outliers for all the feature point trajectories \mathbf{p}_α , $\alpha = 1, \dots, N$ using the procedure described in Section 3.1.
- (2) Fit a 3-dimensional affine space to the inlier trajectories. Compute the $n \times n$ moment matrix \mathbf{M} in Eq. (11). Let $\lambda_1 \geq \lambda_2 \geq \lambda_3$ be the three largest eigenvalues of the matrix \mathbf{M} , and $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ the orthonormal system of corresponding eigenvectors.

Incorrect feature tracking detection:

For each outlier trajectory \mathbf{p}_α , we do the following:

- (1) Consider the partial trajectory $\mathbf{p}_\alpha^{(0)}$ consisting of the point in the 1st frame. Let $k = 2$, $\kappa = 2$.
- (2) Add the point in the κ -th frame, and let $k \leftarrow k + 2$.
- (3) Test the resulting partial trajectory $\mathbf{p}_\alpha^{(0)}$ if it is reliable, using Eq. (16). If it is not judged to be reliable, remove the point in the κ -th frame, and let

$$k \leftarrow k - 2.$$

- (4) Let $\kappa \leftarrow \kappa + 1$, and go back to Step 2. Repeat this until all the frames are tested.

The right hand of Eq. (16) is defined if the dimension of the partial trajectory is larger than four, which means we need at least two frames for testing the reliability. Therefore, we can theoretically detect an incorrectly tracked data using the partial trajectory consisting of points in only the 1st and the target frame. However, the residual may be very small. By accumulating the detected reliable tracking points, however, we expect that our proposed method effectively detects incorrectly tracked data.

5.2 RANSAC Approach

Feature point tracking often fails in the first several frames and begins to track another point. Sometimes, the correctly tracked point may be longer than the first part. If we extend a very short partial trajectory, the accuracy is generally low.

In order to cope with this problem, we detect the longest partial trajectory consisting of correctly tracked points by RANSAC^{2),3)}. Instead of starting from the first frame, we randomly select a base point and detect a correctly tracked part. We select the longest partial trajectory as follows:

- (1) Randomly select one frame, and consider the partial trajectory $\mathbf{p}_\alpha^{(0)}$ consisting of the point in the selected frame. Let $k = 2$, $\kappa = 1$, $S = 0$
- (2) Add the point in the κ -th frame, and let $k \leftarrow k + 2$. If κ is the initially selected frame, go back to Step 2 after updating κ as $\kappa \leftarrow \kappa + 1$.
- (3) Test the partial trajectory $\mathbf{p}_\alpha^{(0)}$ if it is reliable, using Eq. (16). If it is not judged to be reliable, remove the point in the κ -th frame, and let $k \leftarrow k - 2$.
- (4) Let $\kappa \leftarrow \kappa + 1$, and go back to Step 2. Repeat this until all the frames are tested.
- (5) If $k > S$, then $S \leftarrow k$. Repeat the above procedure and determine the partial trajectory $\mathbf{p}_\alpha^{(0)}$ that maximizes S ^{*1}.

As we described, the RANSAC approach can detect the longest partial tra-

jectories consisting of correctly tracked points. This is an advantage against the method which selects the first point as the base point. However, we sometimes need to track the feature points extracted from the first frame. For example, if we reconstruct a 3-D shape by the factorization method, we may manually choose feature points which lay on the object corners and contours in the first frame. In this case, the method which selects the first point as the base point has an advantage against the RANSAC approach. Users can select the two approaches and use together according to applications.

6. Simulations

6.1 Simulations for Perspective Effects

We assume an affine camera model, however, tracked feature points obtained from real video sequences do not necessarily satisfy this assumption enough. In order to confirm how our method was effective for such tracking data with a perspective effect, we did experiment using simulated data.

We generated 91 3-D points on a quarter cylinder, and generated four types of 30 frames tracking data by projected them onto an image plane assuming four camera models with different perspective effects. **Figure 3** (a)–(d) are the feature points in the 15th frame computed from Eq. (17),

$$x = \frac{X}{1 + \frac{Z \tan \theta}{L}}, \quad y = \frac{Y}{1 + \frac{Z \tan \theta}{L}}, \quad (17)$$

and we set $\theta = 0^\circ, 10^\circ, 20^\circ, \text{ and } 30^\circ$, respectively. X, Y , and Z are the 3-D position of the simulated data, and x and y are its projected 2-D position. L is a constant for controlling perspective effects^{*2}.

For four partial feature trajectories, which consist of the tracked data from the 10th frame to the 30th frame, shown by the symbol \times in Fig. 3 (a)–(d), we moved its positions by 5, 10, 25, and 50 pixels from their original positions, respectively, then added random Gaussian noise of mean 0 and standard deviation 1 (pixel). We detected outliers from these tracking data by using the method described in Section 3, then we obtained 4, 18, 37, and 31 outliers including four true outliers

*1 In our experiment, we stopped if S did not increase for 5 consecutive iterations.

*2 In our experiment, we set $L = 400$, it is about 10 times of the depth range.

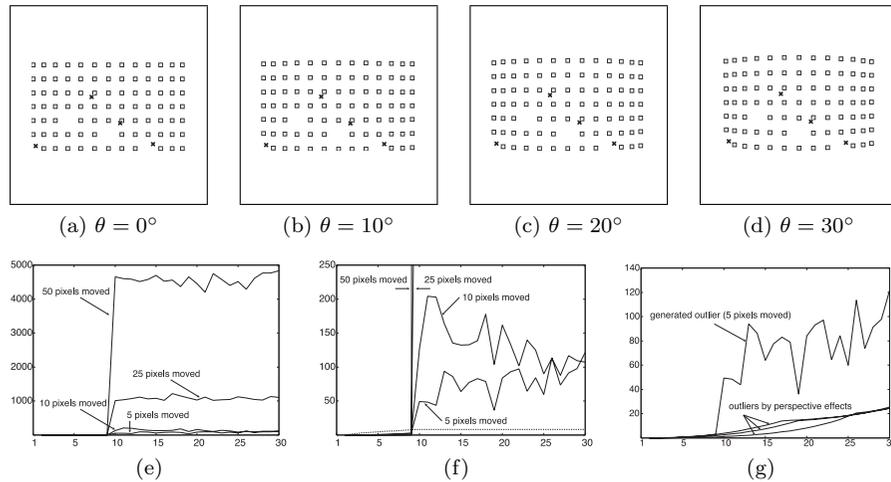


Fig. 3 (a)–(d) feature positions in the 15th frame projected by Eq. (17), $\theta = 0^\circ, 10^\circ, 20^\circ$, and 30° (\times for outliers). (e) The results of detecting incorrect tracking for the data set of $\theta = 30^\circ$. (f) expansion of (e). (g) The residuals for other outliers.

for the data sets of $\theta = 0^\circ, 10^\circ, 20^\circ$, and 30° , respectively. The reason why some inliers are detected as outliers is the tracking data are not strictly satisfied an affine camera model by a perspective effect.

For detected outliers, we applied our proposed method and confirmed our method successfully worked for all data sets. Figure 3(e) shows the results of incorrect tracking detection for four outliers of the data set of $\theta = 30^\circ$. The horizontal and vertical axes show the frame number and the residual for the fitted affine space, respectively. We can see that the residuals become large from the 10th frame for all outlier trajectories. Figure 3(f) shows the expansion of Fig. 3(e). The dotted line indicates the threshold computed by the right hand of Eq. (16). All the residuals become larger than the threshold from the 10th frame. We also checked the other data sets and confirmed that comparable results are obtained. From this, our method is effective for tracking data which do not strictly satisfied an affine camera model. In Fig. 3(g), on the other hands, we show three residuals which are regarded as outliers by perspective effects, for contrast one residual for the simulated outlier is plotted. As we can see, the

Table 1 Data set configurations.

	number of frames	correctly tracked frame
data1	5	1–3
data2	10	1–6
data3	15	1–8
data4	30	1–16

residuals gradually increase, and become larger than the threshold. However, the residuals are generally small, so if we set the threshold more large then we may avoid to detect inliers as outliers.

6.2 Stability of RANSAC Approach

We also experimented the stability of the RANSAC based method against number of frames. We generated a simulation data by Eq.(17) with $\theta = 0$, and then manually generated four miss tracking data set which were shown in **Table 1**. We decimated each number of frames from the 30 frames sequence, and moved its positions by 5 pixels from their original positions, then added random Gaussian noise of mean 0 and standard deviation 1 (pixel).

All data set include nearly 50% outliers. For each data set, we detected incorrect tracking frames by the RANSAC based method. We did 100 trials for each data set, and visually confirmed that the RANSAC based method output the correct results for all data set of all trials. From this result, the RANSAC based method is stable for the number of frames.

7. Real Video Experiments

We test our method using real video sequences. **Figure 4**(a) show three decimated frames from a 100 frame sequences (320×240 pixels) of a static poster scene taken by a moving camera. We detected 200 feature points and tracked them using the Kanade-Lucas-Tomasi algorithm¹⁹⁾. Among them, 121 feature points are completely tracked over the entire frames, and 6 are regarded as outliers. The symbol \square in Fig. 4 indicates inlier positions, and the symbol \times indicates outlier positions.

Figure 4(b)–(d) show the results of incorrect tracking detection for three outliers. The horizontal and vertical axes show the frame number and the residual for the fitted affine space, respectively. The solid line indicates the residual of the

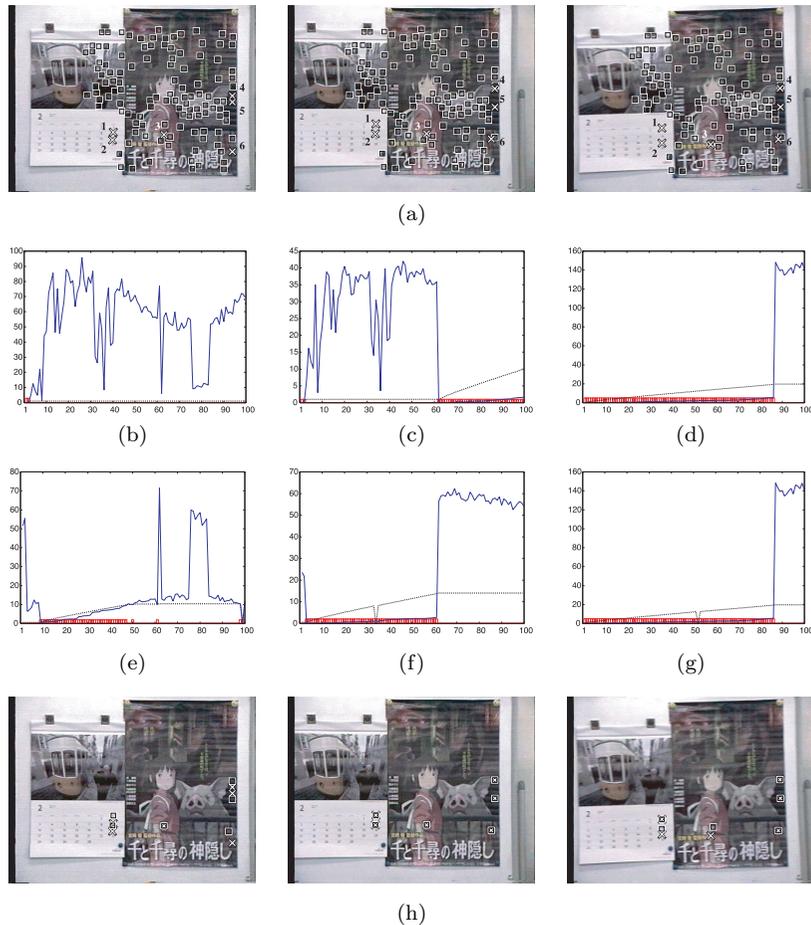


Fig. 4 (a) Three decimated frames of a 100 frame image sequence and 121 feature points successfully tracked (\square for inlier positions; \times for outlier positions). (b), (c), (d) The results of detecting incorrect tracking for the feature points starting from the 1st frame (the feature point ID 1, 2, and 3). Solid line is for the residual of the fitted affine space; dotted line is for the threshold; the box marks are for detecting frames for the correctly tracked feature points. (e), (f), (g) The results of detecting the longest tracked feature points (the feature point ID 1, 2, and 3). (h) Estimation of the missing positions for the resulting longest partial trajectories (\square for estimated positions; \times for originally tracked positions).

partial trajectory, which is computed by the left hand of Eq. (16). The dotted line indicates the threshold computed by the right hand of Eq. (16). The box marks indicate that the feature points in its frame are correctly tracked. In order to remove outliers and detect incorrect tracking, we need to know the standard deviation of noise ϵ . Theoretically, it can be estimated if the noise in each frame is independent and Gaussian⁸⁾. 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 frame¹⁵⁾ as Sugaya and Kanatani¹⁴⁾ pointed out. Considering this, we set the value σ for removing outliers and detecting incorrect tracking to be 0.5 and 0.3, respectively (including the simulation experiment). We visually inspected all the outlier trajectories frame by frame to see if they are really correct and confirmed that our method worked correctly.

We also detected partial trajectories consisting of the points correctly tracked through the longest frame sequence by the method described in Section 5. Figure 4 (e)–(g) show the result for the outlier trajectories in Fig. 4 (b)–(d). In Fig. 4 (e), we visually inspected the result and noticed that some correctly tracked feature points were not detected. However, we confirmed that all the detected feature points were correctly tracked. From the result in Fig. 4 (f), we can see that another feature point, not the point extracted from the 1st frame, are correctly tracked from the 3rd frame to the 61st frame. Figure 4 (g), we also confirm that the same result are given in Fig. 4 (d). Using the trajectories obtained in Fig. 4 (e)–(g), we estimated the missing parts of the feature trajectories by the method of Sugaya and Kanatani¹⁶⁾. As we can see in Fig. 4 (h), the correct positions are obtained. We also computed the execution time for detecting incorrect tracking. It took about 20 seconds for obtaining each of the results in Fig. 4 (b)–(d), and 120 seconds for Fig. 4 (e)–(g). We used Intel Core2Duo E6700 2.66 GHz for the CPU and Linux for the OS.

Figure 5 shows the result of applying the proposed method to a structure from motion. Figure 5 (a) shows three decimated frames from a 150 frame sequence (640×480 pixels). We detected 200 feature points and tracked them. Among them, 108 were completely tracked through the sequence. From them, 49 trajectories were regarded as outliers. From these outlier trajectories, we de-

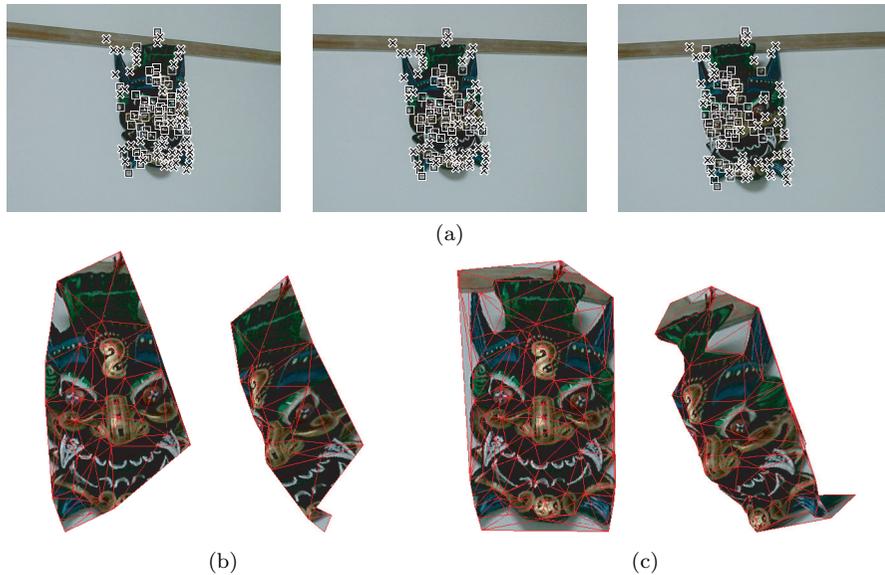


Fig. 5 (a) Three decimated frames of a 150 frame age sequence and 108 feature points successfully tracked (\square for inlier positions; \times for outlier positions). (b) The texture-mapped 3-D shape reconstructed from the original 59 inlier trajectories. (c) The texture-mapped 3-D shape reconstructed after adding 49 corrected trajectories.

tected longest correct trajectories. Then, we extrapolated them by the method of Sugaya and Kanatani¹⁶⁾.

We reconstructed the 3-D shape by factorization, assuming weak perspective projection. Figure 5 (b) shows the front and the side views of the texture-mapped 3-D shape reconstructed from the original 59 inlier trajectories. Figure 5 (c) shows the front and the side views of the texture-mapped 3-D shape reconstructed after adding 49 corrected trajectories. From these results, we can see that the detailed structure is reconstructed by detecting and correcting incorrect tracking data.

8. Concluding Remarks

We have presented a new method for detecting incorrect tracking data in a feature point tracking. We have detected incorrect parts by imposing the constraint

that under the affine camera model feature trajectories should be in an affine space in the parameter space. Introducing a statistical model of image noise, we have tested if a partial trajectory is sufficiently reliable. Then we have detected incorrect partial trajectories.

From the simulation, we confirmed that our method works well for tracking data which do not strictly satisfied an affine camera model. Using real video images, we have demonstrated that our proposed method can detect incorrect feature point tracking fairly well. We also confirmed the effectiveness of our method using a shape from motion application example.

Acknowledgments This work was partially supported in part by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under the Grant in Aid for Young Scientists (B) (No.18700181), 2008. The authors thank Kenichi Kanatani of Okayama University, Japan for helpful comments.

References

- 1) Brandt, S.: Closed-form solutions for affine reconstruction under missing data, *Proc. Statistical Methods in Video Processing Workshop*, pp.109–114, Copenhagen, Denmark (June 2002).
- 2) Fischler, M.A. and Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Comm. ACM*, Vol.24, No.6, pp.381–395 (June 1981).
- 3) Hartley, R. and Zisserman, A.: *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, U.K. (2000).
- 4) Huynh, D.Q. and Heyden, A.: Outlier detection in video sequences under affine projection, *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, Vol.2, pp.695–701, Kauai, HI, U.S.A. (Dec. 2001).
- 5) Ichimura, N.: Stochastic filtering for motion trajectory in image sequences using a Monte Carlo filter with estimation of hyper-parameters, *Proc. 16th Int. Conf. Pattern Recog.*, Vol.4, pp.68–73, Quebec City, Canada (Aug. 2002).
- 6) Ichimura, N. and Ikoma, N.: Filtering and smoothing for motion trajectory of feature point using non-gaussian state space model, *IEICE Trans. Inf. Syst.*, Vol.E84-D, No.6, pp.755–759 (2001).
- 7) Jacobs, D.W.: Linear fitting with missing data for structure-from-motion, *Comput. Vision Image Understand*, Vol.82, No.1, pp.57–81 (Apr. 2001).
- 8) Kanatani, K.: *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier Science, Amsterdam, the Netherlands (1996).
- 9) Kanatani, K.: Motion segmentation by subspace separation and model selection,

Proc. 8th Int. Conf. Comput. Vision, Vancouver, Canada, Vol.2, pp.301–306 (July 2001).

- 10) Kanatani, K.: Motion segmentation by subspace separation: Model selection and reliability evaluation, *Int. J. Image Graphics*, Vol.2, No.2, pp.179–197 (Apr. 2002).
- 11) Kanatani, K.: Evaluation and selection of models for motion segmentation, *Proc. 7th Euro. Conf. Comput. Vision*, Copenhagen, Denmark, pp.335–349 (June 2002).
- 12) Poelman, C.J. and Kanade, T.: A paraperspective factorization method for shape and motion recovery, *IEEE Trans. Patt. Anal. Mach. Intell.*, Vol.19, No.3, pp.206–218 (March 1997).
- 13) Saito, H. and Kamijima, S.: Factorization method using interpolated feature tracking via projective geometry, *Proc. 14th British Machine Vision Conf.*, Vol.2, pp.449–458, Norwich, UK (Sept. 2003).
- 14) Sugaya, Y. and Kanatani, K.: Automatic camera model selection for multi-body motion segmentation, *Proc. IAPR Workshop on Machine Vision Applications (MVA 2002)*, pp.412–415, Nara, Japan (Dec. 2002).
- 15) Sugaya, Y. and Kanatani, K.: Outlier removal for motion tracking by subspace separation, *IEICE Trans. Inf. Syst.*, Vol.E86-D, No.6, pp.1095–1102 (June 2003).
- 16) Sugaya, Y. and Kanatani, K.: Extending interrupted feature point tracking for 3-D affine reconstruction, *IEICE Trans. Inf. Syst.*, Vol.E87-D, No.4, pp.1031–1038 (2004).
- 17) Sugaya, Y. and Kanatani, K.: Multi-stage optimization for multi-body motion segmentation, *IEICE Trans. Inf. Syst.*, Vol.E87-D, No.7, pp.1935–1942 (2004).
- 18) Tomasi, C. and Kanade, T.: Shape and motion from image streams under orthography — A factorization method, *Int. J. Comput. Vision*, Vol.9, No.2, pp.137–154 (Nov. 1992).
- 19) Tomasi, C. and Kanade, T.: Detection and Tracking of Point Features, CMU Tech. Rep. CMU-CS-91-132 (Apr. 1991). <http://www.ces.clemson.edu/~stb/klt/>

(Received October 3, 2008)

(Accepted March 15, 2009)

(Released September 24, 2009)

(Communicated by Akihiro Sugimoto)



Chika Takada received B.S. and M.S. degrees in computer science in 2007 and 2009, respectively, from Toyohashi University of Technology, where she studied computer vision.



Yasuyuki Sugaya was born in 1972. He received his M.S. and Ph.D. in computer science from University of Tsukuba, Ibaraki, Japan, in 1998 and 2001, respectively. After serving as Assistant Professor of computer science at Okayama University, Okayama, Japan, he is currently Lecturer of computer science at Toyohashi University of Technology, Aichi, Japan. His research interests include image processing and computer vision. He received the IEICE best paper awards in 2005.