

Detecting Incorrect Feature Tracking by Affine Space Fitting

Chika Takada and Yasuyuki Sugaya

Department of Information and Computer Sciences,
Toyohashi University of Technology, Toyohashi, Aichi, 441-8580, Japan
{takada,sugaya}@iim.ics.tut.ac.jp

Abstract. 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 motion [18], and motion segmentation [9,10,11,14,17]. Many authors use the Kanade-Lucas-Tomasi algorithm [19]. However, the resulting trajectories are not always correct. In order to improve the tracking, Ichimura and Ikoma [6] and Ichimura [5] introduced nonlinear filtering. Hyunh and Heyden [4], 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 Kanatani [15] 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. Fig. 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 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 Kamijima [13] 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 Kanatani [16] extended partial trajectories by imposing the constraint that

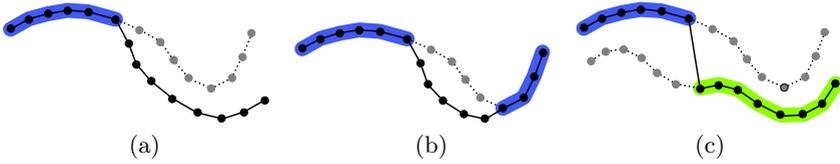


Fig. 1. Outlier trajectories which contain partially correct tracking. Solid lines are for trajectories. Dotted lines are for correct trajectories. (a) Tracking fails and strays after that. (b) Tracking returns to a correct path after failing. (c) Tracking fails and follows another path.

under the affine camera model all feature trajectories should be in an affine space. This is called the *affine space constraint*. 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.

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

2 Affine Space Constraint

We summarize the geometric constraints on which our method is based. The same constraints have already been used in [9,10,11,15]. 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 projections [12]) 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 (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 (4), the trajectory vector \mathbf{p}_α in (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

(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 Jacobs [7] is based.

In addition, the coefficient of \mathbf{m}_0 in (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 factorization [18] is based on this representation, and Brandt [1] 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 Kanatani [15] 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 constraint [15]. Our method is a direct consequence of the principle given in [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 times¹, 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 [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 $\{\mathbf{p}_\alpha\}$, $\alpha = 1, \dots, N$, be their trajectory vectors. We first compute their centroid

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

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

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

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 Incorrect Feature Tracking Detection

4.1 Partial Trajectories

If 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 partition² 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 Test

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 . 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, (12) is replaced by

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

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.

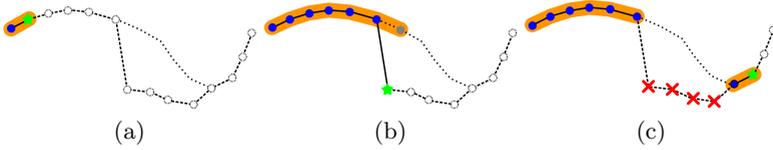


Fig. 2. Detection algorithm. Solid line is for a testing partial trajectory. Dashed line is for a really tracked trajectory. Dotted line is for an ideally correct trajectory. (a) Reliability test for the partial trajectory consisting of the image coordinates of the 1st and 2nd frames (• for inlier frame;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

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

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

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

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

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

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

as outliers with significance level 1%.

4.3 Detection of Incorrect Feature Tracking

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 (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 (16) (Fig. 2(a)). If the partial trajectory is judged to be reliable, we add 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 Sec. 3.1.
2. Fit a 3-dimensional affine space to the inlier trajectories. Compute the $n \times n$ moment matrix \mathbf{M} in (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 tracing 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 (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 (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 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 (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^3 .

³ In our experiment, we stopped if S did not increase for 5 consecutive iterations.

6 Experiments

We test our method using real video sequences. Fig. 3(a) shows three decimated frames from a 100 frame sequence (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 algorithm [19]. Among them, 121 feature points are completely tracked over the entire frames, and 6 are regarded as outliers. The symbol \square in Fig. 3 indicates inlier positions, and the symbol \times indicates outlier positions.

Fig. 3(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 partial trajectory, which is computed by the left hand side of (16). The dotted line indicates the threshold computed by the right hand side of (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 Gaussian [8]. 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 [15] as Sugaya and Kanatani [14] pointed out. Considering this, we set the value σ to be 0.5 and 0.3, respectively. We visually inspected all the outliers 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 Sec. 5. Fig. 3(e)–(g) show the result for the outlier trajectories in Fig. 3(b)–(d). In Fig. 3(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. 3(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. In Fig. 3(g), we also confirm that the same result are given in Fig. 3(d). Using the trajectories obtained in Fig. 3(e)–(g), we estimated the missing parts of the feature trajectories by the method of Sugaya and Kanatani[16]. As we can see in Fig. 3(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. 3(b)–(d), and 120 seconds for Fig. 3(e)–(g). We used Intel Core2Duo E6700 2.66 GHz for the CPU and Linux for the OS.

Fig. 4 shows the result of applying the proposed method to a structure from motion. Fig. 4(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 detected longest correct trajectories. Then, we extrapolated them by the method of Sugaya and Kanatani[16]. Fig. 4(b) shows the outlier trajectories. The solid lines indicate correct paths. The dotted lines indicate incorrect paths. The dashed lines indicate corrected paths. For details, we picked up one outlier trajectory in Fig. 4(c).

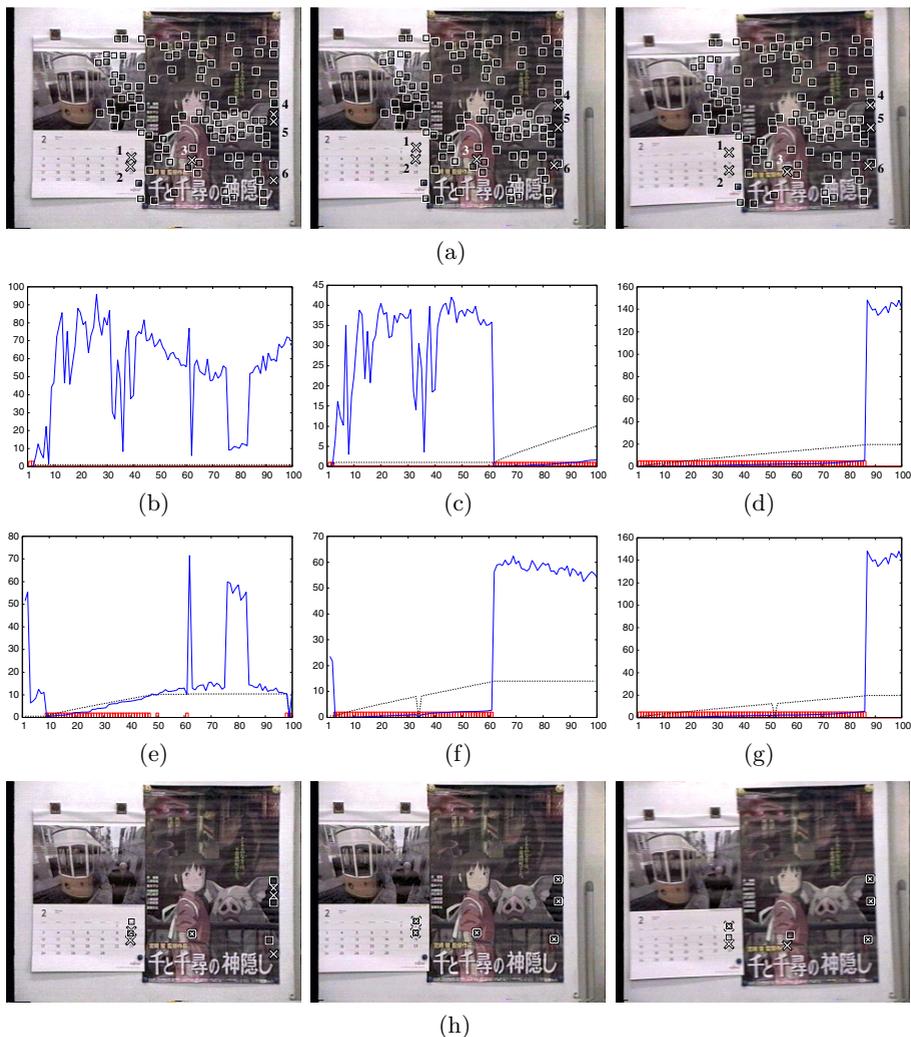
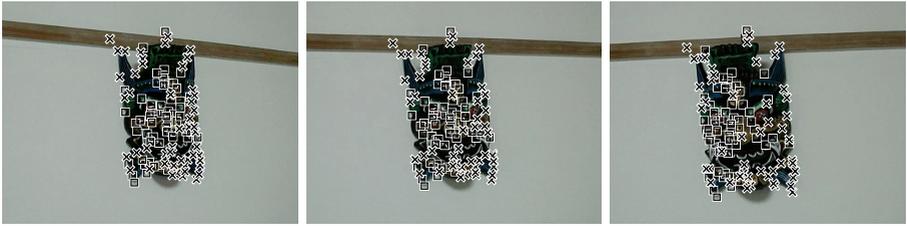
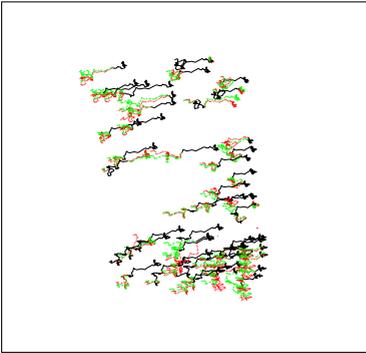


Fig. 3. (a) Three decimated frames of a 100 frame image sequence and 121 feature points successfully tracked (\square for inlier positions; \times for outlier positions). (b), (c), (d) The results of detecting incorrect tracking for the feature points starting from the 1st frame (the feature point ID 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).

We reconstructed the 3-D shape by factorization, assuming weak perspective projection. Fig. 4(d) shows the front and the side views of the texture-mapped 3-D shape reconstructed from the original 59 inlier trajectories. Fig. 4(e) shows



(a)



(b)



(c)



(d)



(e)

Fig. 4. (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) Outlier trajectories. Solid lines are for correct paths; dotted lines are for incorrect paths; dashed lines are for corrected paths. (c) One outlier trajectory. (d) The texture-mapped 3-D shape reconstructed from the original 59 inlier trajectories. (e) The texture-mapped 3-D shape reconstructed after adding 49 corrected trajectories.

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.

7 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. Using real video images, we have demonstrated that our proposed method can detect incorrect feature point tracking fairly well.

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. In: Proc. Statistical Methods in Video Processing Workshop, Copenhagen, Denmark, June 2002, pp. 109–114 (2002)
2. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM* 24(6), 381–395 (1981)
3. Hartley, R., Horseman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge (2000)
4. Huynh, D.Q., Heyden, A.: Outlier detection in video sequences under affine projection. In: Proc. IEEE Conf. Comput. Vision Pattern Recog., Kauai, HI, U.S.A., December 2001, vol. 2, pp. 695–701 (2001)
5. Ichimura, N.: Stochastic filtering for motion trajectory in image sequences using a Monte Carlo filter with estimation of hyper-parameters. In: Proc. 16th Int. Conf. Pattern Recog., Quebec City, Canada, August 2002, vol. 4, pp. 68–73 (2002)
6. Ichimura, N., Ikoma, N.: Filtering and smoothing for motion trajectory of feature point using non-gaussian state space model. *IEICE Trans. Inf. Syst.* E84-D(6), 755–759 (2001)
7. Jacobs, D.W.: Linear fitting with missing data for structure-from-motion. *Comput. Vision Image Understand.* 82(1), 57–81 (2001)
8. Kanatani, K.: *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier Science, Amsterdam (1996)
9. Kanatani, K.: Motion segmentation by subspace separation and model selection. In: Proc. 8th Int. Conf. Comput. Vision, Vancouver, Canada, vol. 2, pp. 301–306 (2001)
10. Kanatani, K.: Motion segmentation by subspace separation: Model selection and reliability evaluation. *Int. J. Image Graphics* 2(2), 179–197 (2002)
11. Kanatani, K.: Evaluation and selection of models for motion segmentation. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2352, pp. 335–349. Springer, Heidelberg (2002)
12. Poelman, C.J., Kanade, T.: A paraperspective factorization method for shape and motion recovery. *IEEE Trans. Patt. Anal. Mach. Intell.* 19(3), 206–218 (1997)

13. Saito, H., Kamijima, S.: Factorization method using interpolated feature tracking via projective geometry. In: Proc. 14th British Machine Vision Conf., Norwich, UK, vol. 2, pp. 449–458 (September 2003)
14. Sugaya, Y., Kanatani, K.: Automatic camera model selection for multibody motion segmentation. In: Proceedings of the IAPR Workshop on Machine Vision Applications (MVA 2002), Nara, Japan, 11-13 December, pp. 412–415 (2002)
15. Sugaya, Y., Kanatani, K.: Outlier removal for motion tracking by subspace separation. IEICE Trans. Inf. & Syst. E86-D(6), 1095–1102 (2003)
16. Sugaya, Y., Kanatani, K.: Extending interrupted feature point tracking for 3-D affine reconstruction. IEICE Transactions on Information and Systems E87-D(4), 1031–1038 (2004)
17. Sugaya, Y., Kanatani, K.: Multi-stage optimization for multi-body motion segmentation. IEICE Transactions on Information and Systems E87-D(7), 1935–1942 (2004)
18. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography—A factorization method. *Int. J. Comput. Vision* 9(2), 137–154 (1992)
19. Tomasi, C., Kanade, T.: Detection and Tracking of Point Features, CMU Tech. Rep. CMU-CS-91-132 (April 1991), <http://www.ces.clemson.edu/~stb/klf/>