Memoirs of the Faculty of Engineering, Okayama University, Vol. 37, No. 1, pp. 37–44, November 2002

Outlier Removal for Motion Tracking by Subspace Separation

Yasuyuki SUGAYA* and Kenichi KANATANI
Department of Information Technology, Okayama University
Okayama 700-8530 Japan

(Received September 27, 2002)

Many feature tracking algorithms have been proposed for motion segmentation, but the resulting trajectories are not necessarily correct. In this paper, we propose a technique for removing outliers based on the knowledge that correct trajectories are constrained to be in a subspace of their domain. We first fit the subspace to the detected trajectories robustly using RANSAC and then remove those that have large residuals. Using real video sequences, we demonstrate that our method is effective even if multiple objects are moving in the scene. We also confirm that the separation accuracy is indeed improved by our method.

1. Introduction

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

While most existing methods for multi-body segmentation combine such information as optical flow, color, and texture along with miscellaneous heuristics, Costeira and Kanade [1] presented a segmentation algorithm based only on the image motion of feature points. Since then, various modifications and extensions of their method have been proposed.

Gear [3] used the reduced row echelon form and graph matching. Ichimura [5] applied the discrimination criterion of Otsu [18]. He also used the QR decomposition for feature selection [6]. Inoue and Urahama [9] introduced fuzzy clustering. Kanatani [13, 14] introduced model selection and robust estimation based on a new geometric interpretation of the Costeira-Kanade algorithm. Maki and Wiles [17] and Maki and Hattori [16] used Kanatani's method for analyzing the effect of illumination on moving objects. Wu, et al. [22] introduced orthogonal subspace decomposition.

*E-mail sugaya@suri.it.okayama-u.ac.jp

In all these methods, two issues need to be resolved. One is the estimation of the number of independent motions. Many authors set an appropriate threshold for this, but it has been reported that estimating the number of motions is often more difficult than the segmentation itself [3]. To cope with this problem, Kanatani and Matsunaga [15] proposed the use of model selection criteria.

The other issue is the feature tracking. Most authors use the Kanade-Lucas-Tomasi algorithm [20] for this, but the resulting trajectories are not always correct. In order to improve the tracking results, Ichimura and Ikoma [8] and Ichimura [7] introduced nonlinear filtering. Huynh 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 robustly fitting a 4-dimensional subspace to them.

In this paper, we extend the method of Huynh and Heyden [4] to multiple moving objects. Adopting Kanatani's geometric interpretation of the segmentation problem [13, 14], we robustly fit an appropriate subspace to the detected trajectories using RANSAC and remove those that have large residuals.

Section 2 summarizes the subspace constraint used by Kanatani [13]. Sections 3 and 4 describe our outlier removal procedure. In Sec. 5, we show real video sequence examples and demonstrate that our method is effective even if multiple objects are moving in the scene. We also confirm that the separation accuracy

is indeed improved by our method. Section 6 gives our conclusion.

2. Subspace Constraint

We track N rigidly moving feature points over M frames and let $(x_{\kappa\alpha}, y_{\kappa\alpha})$ be the image coordinates of the αth point in the κth frame. If we stack all the image coordinates vertically into a $2M$-dimensional vector in the form

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

the trajectory of a moving point can be represented as a single point in a $2M$-dimensional space.

We take an XYZ camera coordinate system with the Z-axis in the direction of the optical axis. We fix an object coordinate system to the moving object and let $\boldsymbol{t}_\kappa$ and $\{\boldsymbol{i}_\kappa, \boldsymbol{j}_\kappa, \boldsymbol{k}_\kappa\}$ be, respectively, its origin and orthonormal basis in the κth frame. If we let $(a_\alpha, b_\alpha, c_\alpha)$ be the object coordinates of the αth point, its position in the κth frame is

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

with respect to the camera coordinate system.

Assuming an affine camera model (e.g., orthographic, weak perspective, or paraperspective projection [10]), we have

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

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

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

where $\tilde{\boldsymbol{m}}_{0\kappa}, \tilde{\boldsymbol{m}}_{1\kappa}, \tilde{\boldsymbol{m}}_{2\kappa}$, and $\tilde{\boldsymbol{m}}_{3\kappa}$ are 2-dimensional vectors determined by the position and the orientation of the camera and its internal parameters in the κth frame. If the vectors $\tilde{\boldsymbol{m}}_{0\kappa}, \tilde{\boldsymbol{m}}_{1\kappa}, \tilde{\boldsymbol{m}}_{2\kappa}$, and $\tilde{\boldsymbol{m}}_{3\kappa}$ are stacked over the M frames vertically into $2M$-dimensional vectors $\boldsymbol{m}_0$, $\boldsymbol{m}_1$, $\boldsymbol{m}_2$ and $\boldsymbol{m}_3$, respectively, the vector $\boldsymbol{p}_\alpha$ has the form

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

This implies that the trajectories of points that belong to the same object are constrained to be in the 4-dimensional subspace spanned by $\{\boldsymbol{m}_0, \boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3\}$ in $\mathcal{R}^{2M}$.

3. Subspace Separation

It follows from the above observation that multiple moving objects can be segmented by separating the corresponding points in $\mathcal{R}^{2M}$ into distinct 4-dimensional subspaces. This is the principle underlying the *subspace separation* of Kanatani [13], who constructed a robust segmentation algorithm by combining the Costeira-Kanade algorithm [1] with model selection using the geometric AIC [12] and robust estimation using LMedS [19]. Applying his method to real and synthetic images, he demonstrated that the performance was indeed superior to other existing methods.

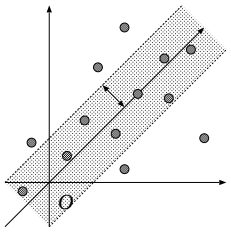


Figure 1: Removing outliers by fitting a subspace.

However, his method, as well as other similar methods, works only when all the trajectories are correct. In real video processing, detected trajectories are not all correct. Motivated by 3-D reconstruction applications, Huynh and Heyden [4] proposed a procedure for removing outlier trajectories from an image sequence of a static scene taken by a moving camera. They robustly fitted a 4-dimensional subspace to the trajectories by random sampling and removed those that have large residuals.

In this paper, we extend their method to multiple objects by noting that if m objects are moving independently in a scene, the points $\{\boldsymbol{p}_\alpha\}$ that represent their trajectories should belong to a $4m$-dimensional subspace. So, we fit a $4m$-dimensional subspace to the detected trajectories using RANSAC [2, 11] and remove those that have large residuals (Fig. 1).

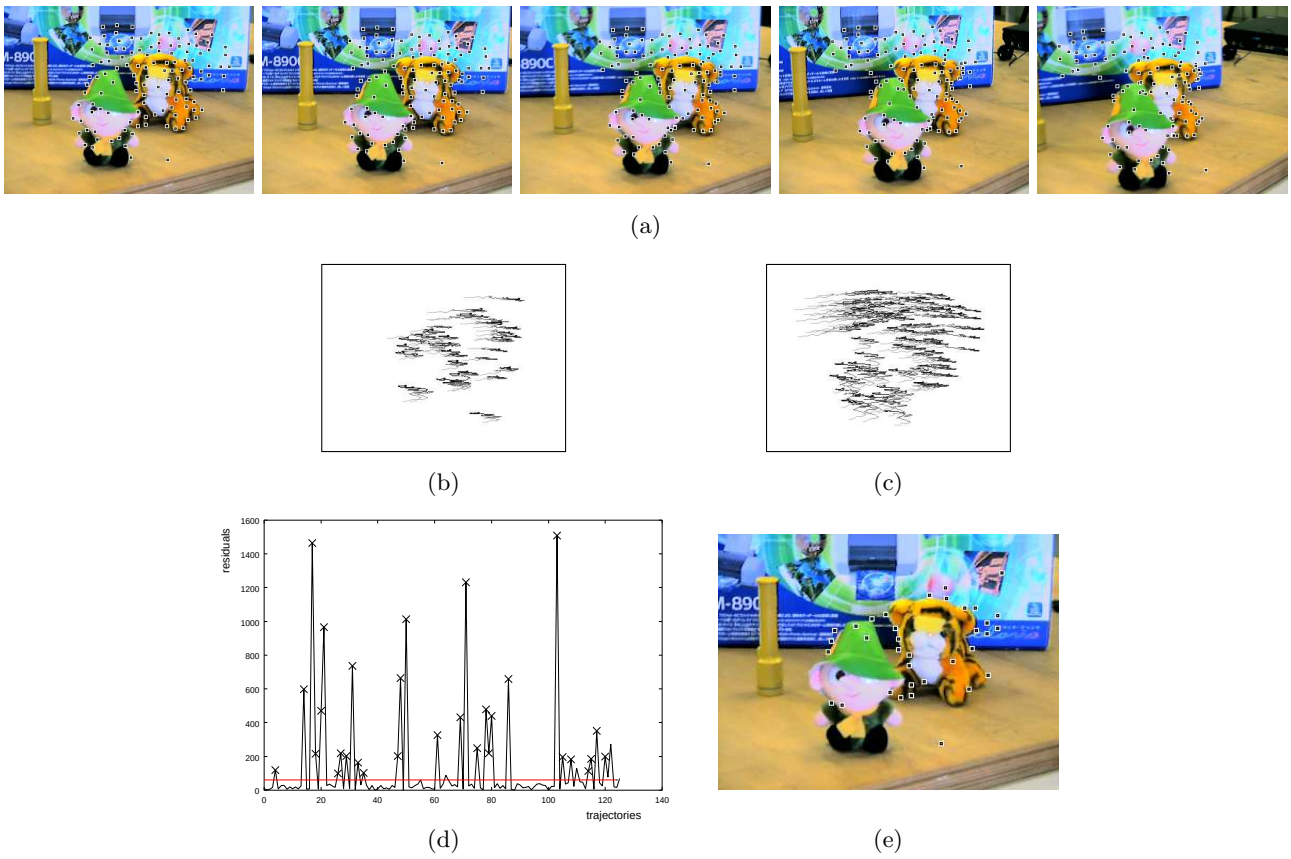
4. Outlier Removal Procedure

We assume that we know the maximum number m of independently moving objects in the scene. Assuming too large a number m is likely to deteriorate the performance of our algorithm, but we do not go into the details of estimating it precisely, since this is a very difficult task with a lot of subtleties involved [15]. In the following, we are mainly concerned with the case for $m = 1$ or 2, which occurs in most practical applications (though theoretically m can be any number).

Since we assume a $4m$-dimensional subspace to the detected trajectories, we assume that more than $4m$ feature points are tracked throughout the sequence. Let $n = 2M$ and $d = 4m$. Our procedure is as follows:

1. Randomly choose d vectors $\boldsymbol{q}_1$, $\boldsymbol{q}_2$, ..., $\boldsymbol{q}_d$ from $\{\boldsymbol{p}_\alpha\}$, $\alpha = 1, \ldots, N$.
2. Define an $n \times n$ matrix

$$\boldsymbol{M}_d = \sum_{i=1}^{d} \boldsymbol{q}_i \boldsymbol{q}_i^\top. \quad (6)$$



(a) (b) (c) (d) (e)

Figure 2: (a) Five decimated frames of an image sequence of a static scene with 126 feature points successfully tracked. (b) The trajectories of detected outliers. (c) The trajectories of detected inliers. (d) The residuals of the trajectories. (e) The locations of the outliers.

3. Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$ be the d eigenvalues of matrix $\boldsymbol{M}_d$, and $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_d\}$ the orthonormal set of the corresponding eigenvectors.

4. Define an $n \times n$ projection matrix

$$\boldsymbol{P}_{n-d} = \boldsymbol{I} - \sum_{i=1}^{d} \boldsymbol{u}_i \boldsymbol{u}_i^\top. \tag{7}$$

5. Let S be the number of points $\boldsymbol{p}_\alpha$ that satisfy

$$\|\boldsymbol{P}_{n-d}\boldsymbol{p}_\alpha\|^2 < (n-d)\sigma^2. \tag{8}$$

Here, $\|\boldsymbol{P}_{n-d}\boldsymbol{p}_\alpha\|^2$, which we call the *residual*, is the squared distance of point $\boldsymbol{p}_\alpha$ from the fitted d-dimensional subspace in $\mathcal{R}^n$, and σ measures the uncertainty of locating feature positions in images.

6. Repeat the above procedure a sufficient number of times[1], and determine the projection matrix $\boldsymbol{P}_{n-d}$ that maximizes S.

7. Remove those $\boldsymbol{p}_\alpha$ that satisfy

$$\|\boldsymbol{P}_{n-d}\boldsymbol{p}_\alpha\|^2 \geq \sigma^2 \chi^2_{n-d;99}, \tag{9}$$

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

[1] In our experiment, we stopped if S did not increase 200 times consecutively.

If the noise in the coordinates of the feature points is an independent random Gaussian variable of mean 0 and standard deviation σ and if the fitted subspace is correct, the residual $\|\boldsymbol{P}_{n-d}\boldsymbol{p}_\alpha\|^2$ divided by σ^2 should be subject to a χ^2 distribution with $n-d$ degrees of freedom, hence its expectation is $(n-d)\sigma^2$, provided $\boldsymbol{p}_\alpha$ is an inlier. The above procedure effectively fits a d-dimensional subspace that maximizes the number of the points whose residuals are smaller than $(n-d)\sigma^2$. After determining the subspace, we remove those points which cannot be regarded as inliers with significance level 1%.

5. Experiments

We tested our method using real video sequences of static scenes and multiple moving objects. We generated and tracked feature points through the entire video stream using the Kanade-Lucas-Tomasi algorithm [20].

Figure 2(a) shows five frames decimated from a 100 frame sequence (320 × 240 pixels) of a static scene taken by a moving camera. We tracked 126 points as indicated by the symbol □ in the images. Setting σ

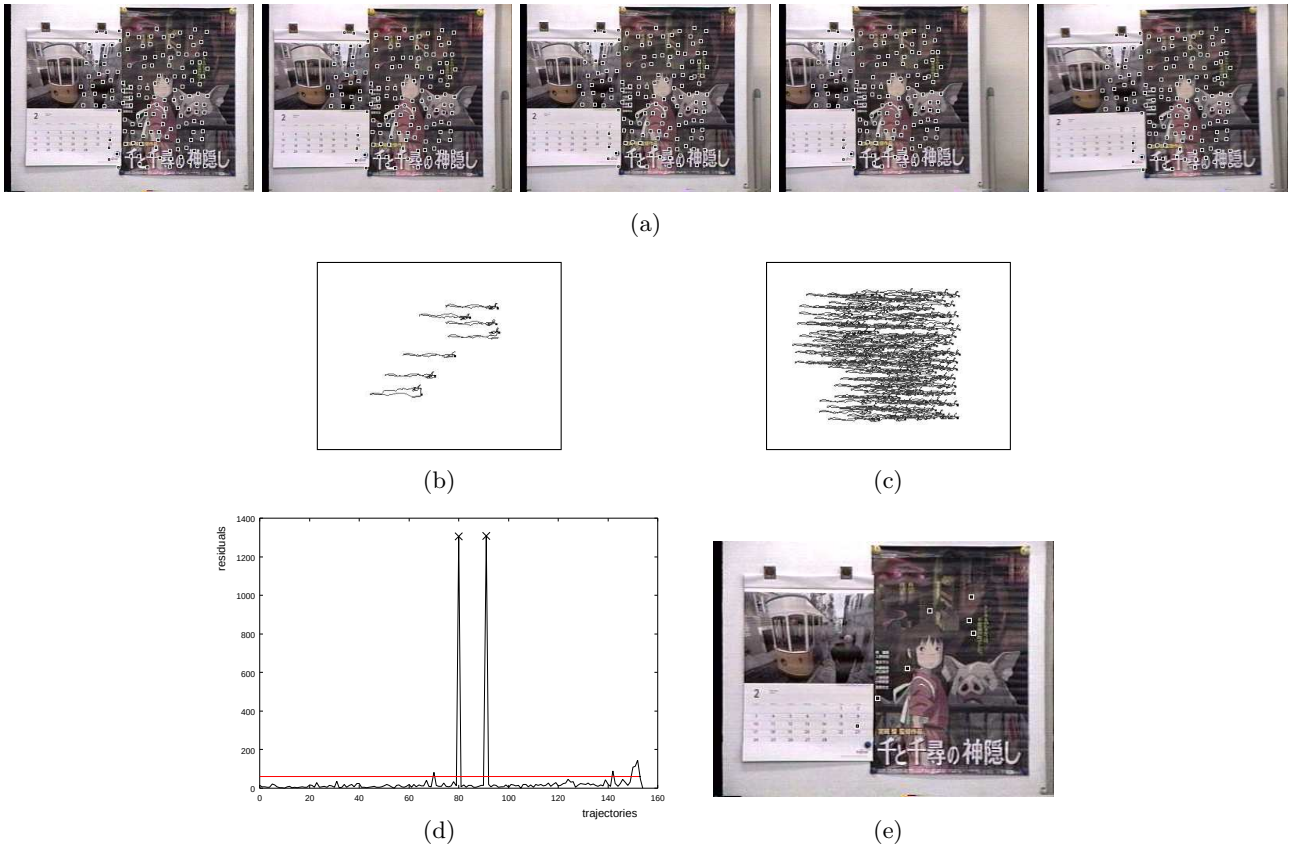


Figure 3: (a) Five decimated frames of an image sequence of a static scene with 155 feature points successfully tracked. (b) The trajectories of detected outliers. (c) The trajectories of detected inliers. (d) The residuals of the trajectories. (e) The locations of the outliers.

= 0.5 (pixels), we removed outlier trajectories by our method. Figures 2(b) and (c) show the trajectories judged to be outliers and inliers, respectively.

Figure 2(d) plots the residuals of the 126 trajectories; they are marked on the horizontal axis in numerical order. The horizontal line in the graph indicates the threshold determined by Eq. (9). Figure 2(e) shows the locations of the detected outliers in the first frame. We see that many of them are on the occluding contours.

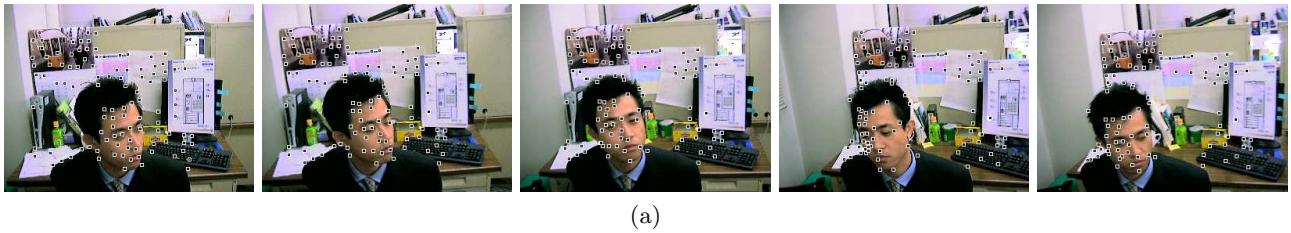
Closely inspecting all the images frame by frame, we checked if the trajectories judged to be outliers were really incorrect. In Fig. 2(d), those trajectories that are indeed false are indicated by the symbol $\times$. As can be seen, their residuals are large enough to be rejected by our procedure. However, some apparently correct trajectories are also rejected as outliers. A close examination revealed that they were caused by points that were fluctuating around their supposed positions by a few pixels throughout the sequence. In practice, removing them, correct as they may be, is a reasonable choice, since inclusion of such unreliable trajectories would lower the reliability of the subsequent segmentation.

Figure 3(a) shows another sequence of a static scene. The results are arranged in the same way in Figs. 3(b)–(e), and similar observations hold. In this case, however, the number of outliers is relatively small, probably because the scene is a planar surface without occluding contours.

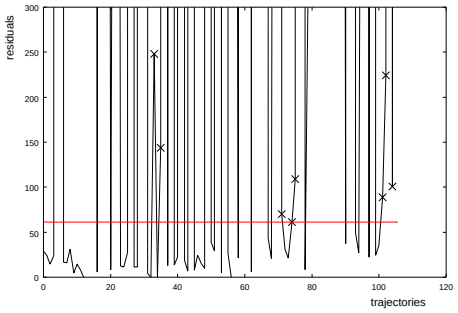
In the sequence shown in Fig. 4(a), an object (a human body) is moving independently of the background, which is also moving in the images. Figure 4(b) shows the residuals of the 107 feature points successfully tracked. This time, the rejected trajectories are all incorrect, while the remaining ones are all correct. Figure 4(c) shows the locations of the detected outliers in the first frame. As can be seen, many of them are on the occluding contours of the moving object.

In order to see the effect of outliers on segmentation, we applied the subspace separation algorithm[2] of Kanatani [13] to this image sequence with and without removing outliers: Figure 4(d) shows the segmentation result without removing outliers; Figure 4(e) shows the result after removing outliers. The symbol $\square$ indicates points classified to the background; the symbols $\times$ indicates points classified to the moving object.

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

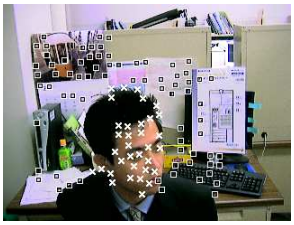


(a)

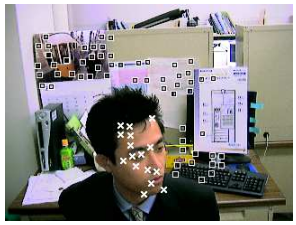


(b)

(c)



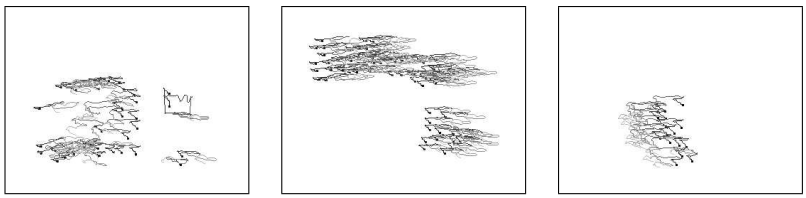
(d)



(e)

	correctness	details
with outliers	94.4%	6/40, 0/67
without outliers	100.0%	0/16, 0/57

(f)



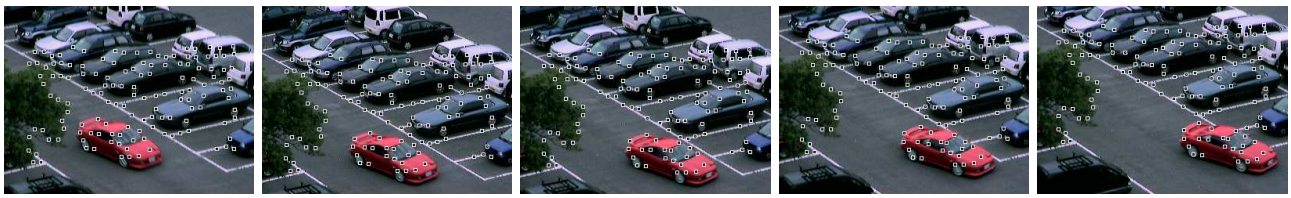
(g) (h) (i)

Figure 4: (a) Five decimated frames of an image sequence of a static scene and a moving object with 107 feature points successfully tracked. (b) The residuals of the trajectories. (c) The locations of the outliers. (d) The segmentation with outliers (× for object points; □ for background points). (e) The segmentation without outliers (× for object points; □ for background points). (f) The correctness of segmentation and the classification details. (g) The trajectories of detected outliers. (h) The trajectories of detected background points. (i) The trajectories of detected object points.

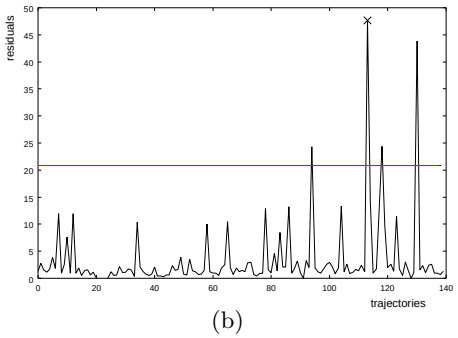
In Fig. 4(d), all inliers are correctly classified; misclassifications occur only for outliers, most of which are on the occluding boundaries of the moving object. In Fig. 4(e), in contrast, all points are correctly classified. The second column of the table in Fig. 4(f) lists the correctness of the segmentation: (the number of misclassified points)/(the total number of points) in percentage for Figs. 4(d) and (e), respectively. The third column lists (the number of misclassifications)/(the number of points classified to the object) and (the number of misclassifications)/(the number of points classified to the background). Figures 4(g), (h), and (i) show, respectively, the trajectories of the detected outliers, the inliers classified to the background, and the inliers classified to the moving object.

Figure 5 shows another example similarly arranged. In this example, the existence of outliers adversely affects the segmentation of inliers, as shown in Figs. 5(d) and (e). In fact, the accuracy of segmentation is improved by removing outliers beforehand.

From Fig. 5(b), we see that some correct trajectories are rejected as outliers. We also see that correct trajectories consists of those with very small residuals and those with relatively large residuals. This clear distinction implies that the detected feature points are divided into two types, unambiguous and ambiguous. An unambiguous point is correctly tracked thoughout the sequence, while an ambiguous point is always ambiguous in the course of the tracking. This

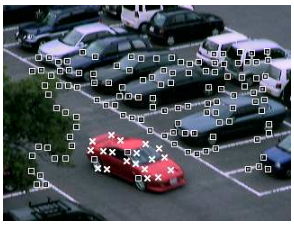


(a)

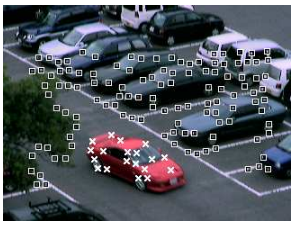


(b)

(c)



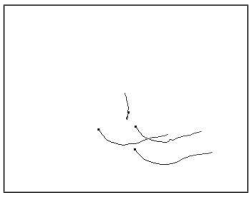
(d)



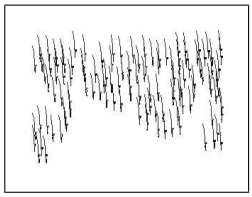
(e)

	correctness	details
with outliers	97.9%	3/22, 0/118
without outliers	100.0%	0/19, 0/117

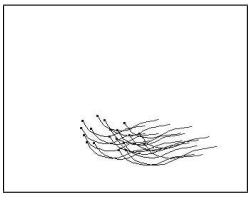
(f)



(g)



(h)



(i)

Figure 5: (a) Five decimated frames of an image sequence of a static scene and a moving object with 140 feature points successfully tracked. (b) The residuals of the trajectories. (c) The locations of the outliers. (d) The segmentation with outliers ($\times$ for object points; $\square$ for background points). (e) The segmentation without outliers ($\times$ for object points; $\square$ for background points). (f) The correctness of segmentation and the classification details. (g) The trajectories of detected outliers. (h) The trajectories of detected background points. (i) The trajectories of detected object points.

phenomenon can be observed more or less in all the previous examples but is particularly strong for this sequence. This is probably because the scene is very far away and the range of the gray levels is relatively narrow.

This is the very reason why we did not set the threshold automatically. If the noise in the coordinates of the feature points were Gaussian and independent for each point and each frame, we could use LMedS [19], estimating the noise level σ from the median of the residuals as described in [19]. In reality, however, this is difficult due to the existence of strong temporal correlations, so we empirically set σ (0.5 pixels in our experiment) and the siginificance level (1% in our experiment) of the rejection decision.

The computation time for the outlier removal procedure was 33.17 sec., 36.83 sec, 32.57 sec, and 1.95 sec for the examples of Figs. 2, 3, 4, and 5, respectively. We used Pentium IV 1.8GHz for the CPU and Linux for the OS.

6. Concluding Remarks

In this paper, we have proposed a technique for removing outliers from the trajectories of feature points detected over a video sequence. Our algorithm robustly fits a subspace to the trajectories and removes those that have large residuals.

Using real video sequences, we demonstrated that our method was effective even if multiple objects are moving in the scene. We also confirmed that the sepa-

ration accuracy was indeed improved by our method.

Our method is based on an affine camera model. Also, feature points must be tracked throughout the sequence. These limit the use of our method to a relatively short sequence of images. For a long sequence, however, we can divide it into overlapping short segments and apply our method to them separately. We can safely assume an affine camera model if the depth of the scene does not vary much in each segment. How to cope with strong perspective effects and how to automatically set the involved parameters are left for future research.

Our approach is based on the *geometric* constraint that the image motion of feature points should be interpreted to be projections of rigid motions in the scene. In contrast, the use of nonlinear filtering proposed by Ichimura [7] and Ichimura and Ikoma [8] is based on the *stochastic* constraint that the image motion of feature points should be "smooth" with a strong temporal coherence. Since these two approaches are complementary in nature, it is expected that the segmentation accuracy will be further increased by combining them.

Acknowledgments: This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under a Grant in Aid for Scientific Research C(2) (No. 13680432), the Support Center for Advanced Telecommunications Technology Research, and Kayamori Foundation of Informational Science Advancement.

References

[1] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," Int. J. Computer Vision, vol.29, no.3, pp.159–179, Sept. 1998.

[2] M. A. Fischer and R. C. Bolles, "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] C. W. Gear, "Multibody grouping from motion images," Int. J. Comput. Vision, vol.29, no.2, pp.133–150, Aug./Sept. 1998.

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

[5] N. Ichimura, "Motion segmentation based on factorization method and discriminant criterion," Proc. 7th Int. Conf. Comput. Vision, vol.1, pp.600–605, Kerkyra, Greece, Sept. 1999.

[6] N. Ichimura, "Motion segmentation using feature selection and subspace method based on shape space," Proc. 15th Int. Conf. Pattern Recog., vol.3, pp.858–864, Barcelona, Spain, Sept. 2000.

[7] N. Ichimura, "Stochastic filtering for motion trajectory in image sequences using a Monte Carlo filter with estimation of hyper-parameters," Proc. 16th Int. Conf. Pattern Recog., Quebec City, Canada, Aug. 2002, to appear.

[8] N. Ichimura and N. Ikoma, "Filtering and smoothing for motion trajectory of feature point using non-gaussian state space model," IEICE Trans. Inf. Syst., vol.E84-D, no.6, pp.755–759, 2001.

[9] K. Inoue and K. Urahama, "Separation of multiple objects in motion images by clustering," Proc. 8th Int. Conf. Comput. Vision, vol.1, pp.219–224, Vancouver, Canada, July 2001.

[10] C. J. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," IEEE Trans. Pat. Anal. Mach. Intell., vol.19, no.3, pp.206–218, 1997.

[11] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, Cambridge, U.K., 2000.

[12] K. Kanatani, Statistical Optimization for Geometric Computation: Theory and Practice, Elsevier, Amsterdam, 1996.

[13] K. Kanatani, "Motion segmentation by subspace separation and model selection," Proc. 8th Int. Conf. Comput. Vision, vol.2, pp.301–306, Vancouver, Canada, July 2001.

[14] K. Kanatani, "Evaluation and selection of models for motion segmentation," Proc. 7th Euro. Conf. Comput. Vision, Copenhagen, Denmark, June 2002.

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

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

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

[18] N. Otsu, "A threshold selection method from gray-level histograms," IEEE Trans. Sys. Man Cyber., vol.9, no.1, pp.62-66, 1979.

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

[20] C. Tomasi and T. Kanade, Detection and Tracking of Point Features, CMU Tech. Rep. CMU-CS-91-132, April 1991;
`http://vision.stanford.edu/~birch/klt/`.

[21] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography—A factorization method," Int. J. Comput. Vision, vol.9, no.2, pp.137–154, Nov. 1992.

[22] Y. Wu, Z. Zhang, T. S. Huang and J. Y. Lin, "Multibody grouping via orthogonal subspace decomposition, sequences under affine projection," Proc. IEEE Conf. Computer Vision Pattern Recog., vol.2, pp.695–701, Kauai, Hawaii, U.S.A., Dec. 2001.