Vol. 46 No. SIG 15(CVIM 12) IPSJ Transactions on Computer Vision and Image Media Oct. 2005

Regular Paper

K-means Tracking with Variable Ellipse Model

Chunsheng Hua,† Haiyuan Wu,† Toshikazu Wada†
and Qian Chen†

We have proposed a K-means clustering based target tracking method, compared with the template matching, which can work robustly *when tracking an object with hole through which the background can be seen* (e.g., *mosquito coil*) (*hereafter we call this problem as the* **background interfusion** *or the* **interfused background**). This paper presents a new method for solving the drawbacks of the previous method, i.e., low speed, instability caused by the change of shape and size. Our new tracking model consists of a single target center, and a variable ellipse model for representing non-target pixels. The contributions of our new method are: 1) The original K-means clustering is replaced by a 2_∞-means clustering, and the non-target cluster center is adaptively picked up from the pixels on the ellipse. This modification reduces the number of distance computation and improves the stability of the target detection as well. 2) The ellipse parameters are adaptively adjusted according to the target detection result. This adaptation improves the robustness against the scale and shape changes of the target. Through the various experiments, we confirmed that our new method improves speed and robustness of our original method.

1. Introduction

Object tracking is an important task in Computer Vision. After years of researches, many efficient methods have been proposed. We can roughly categorize the present tracking methods into three broad classes: *template-based object tracking, probabilistic object tracking* and *pixel-wise tracking*.

The most famous object tracking method is *template matching* [1),2)]. In that work, the object translation is estimated by searching for a small region which has the maximum similarity to the predefined target template. However, when using a constant template, this method suffers from the change of object appearance such as scaling, rotation, etc. If the updatable template is used, it suffers from the positioning error accumulated during the template updating, because the interfusion of background pixels into the template is unavoidable. To solve this problem, Comaniciu et al. introduce the mean-shift method [3)] (or Kernel-based tracking method [4)]). In their method, the color histogram of the target object is calculated through the kernel density estimation. This histogram is used as the feature to describe the target object. Since color histogram is a viewpoint-invariant feature, it is stable to the changes of scale, pose and shape of the target object. The similarity between two features calculated from the target object and a supposed target region in the image respectively will decrease gently when the displacement between it and the target object increases. The object tracking is performed by finding out a region whose color histogram has the maximum similarity to that of the target object. In this way, the tracking becomes stable even for a non-rigid target. However, mean-shift method suffers from the narrow color distribution in the color histogram of the monochromatic target. The reason is that the histogram will change dramatically when the illumination changes, even if the change is small. Another drawback is that its performance is not good when tracking a thin or flat object (like a pencil or book). The tracking will fail when the target rotates in 3D. This is because the shape of ellipse template is not allowed to be changed during the tracking. Moreover, since mean-shift tracking can be considered as a kind of template matching with an updatable template, it shares the same drawback that the small positioning error will be accumulated (shown as **Fig. 1**).

Among the *probabilistic tracking methods*, contour-based method [9),11),12),18)] has the attraction of its good performance for non-rigid objects. Siebel, et al. [9)] combine the motion detector, region tracker, head detector and active shape tracker together to perform a pedestrian tracking system. Its success depends on an assumption that the background without people (moving objects) is known in advance. Since

† Wakayama University

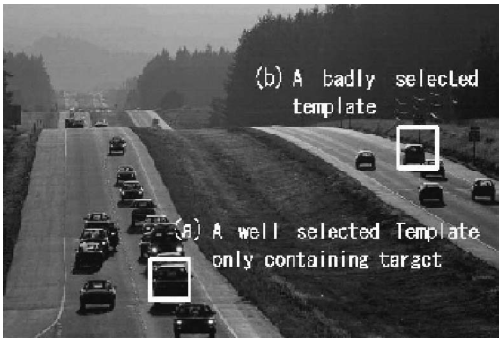


Fig. 1 When the background is interfused into the template, template matching is easy to fail.

contour-based tracking needs the prior information, the object contour has to be defined or trained before tracking, which is not always available in practice. Also, when the object contour is complicated, the initial target contour definition is often laborious and difficult to be made automatically.

CONDENSATION (also called as the particle filter, or MCMC) [7),8),10),13)] has attracted the attention of many researchers. The main idea is that, based on the combination of the parameter vector and statistic computation, object tracking is performed by predicting the object position from the previous information and verifying the existence of object at the predicted position. However it is not straight forward to select a proper feature vector for constructing a CONDENSATION-based real-time tracking system. Khan, et al. [7)] extend this method to deal with the problem caused by the interacting targets. They use a Markov random field (MRF) to model those interactions by adding an interaction factor to the importance weights in a joint particle filter. One drawback of their method is that tracking failure will occur when the targets overlap.

Compared with those model-based tracking methods, *pixel-wise tracking method* can be considered as a data-driven method, which does not require any prior model of the target. Heisele, et al. [5),6)] use a parallel K-means clustering algorithm [17)] to segment the color image sequence and find out the moving region as the target object. However, the huge number of clusters makes their method computationally expensive. Another K-means based autoregressive model is proposed by Agarwal et al. [15)]. Because the clustering is only performed to the positive samples, the tracking failure can not be detected, thus failure recovery will be impossible.

Our research group [21)] has presented a *K-means tracking* method. In that method, the image pixels are divided into target and non-target pixels by applying K-means clustering to a predefined target point and several non-target points. This approach makes the object tracking become robust against the background interfusion. With the help of non-target information, this method has the ability of failure detection and automatic failure recovery. However, since the non-target points can only translate in 2D, it can not deal with the appearance changes of target object, i.e., size, pose, etc. Also, the computational cost is proportional to the number of non-target points.

To solve those problems, we make a lot of improvements over the previous method. In this new research, we assume that: 1) The target object should be monochromatic or contain similar colors (e.g., skin color). There is no special limitation for the target shape, and its free change is also allowed (e.g., hands); 2) The background can be anything, but there should not be large area containing similar colors to the target object. In our new method, we define a variable ellipse model for the object tracking. The ellipse center is used to describe the target cluster center and the ellipse contour is applied to represent the centers of non-target clusters, meanwhile the ellipse region restricts the searching area for K-means clustering. The main contributions are: 1) The original k-means clustering in the previous method is replaced by a 2_∞-means clustering, and the non-target cluster centers are automatically selected from the pixels on the ellipse model. This modification reduces the number of distance computation and improves the stability of target detection. 2) The ellipse parameters are adaptively adjusted according to the detected target pixels. This adaptation improves the robustness against the scale and shape changes of the target.

2. K-means Tracking [21)]

2.1 Basic Idea

To overcome the drawbacks of the conventional pixel-wise tracking methods, our group has presented a *K-means tracking* method [21)], which divides the input pixels into target object and non-target objects by using the K-means clustering algorithm [17),19)]. For clustering, each pixel is represented by a five-dimensional fea-

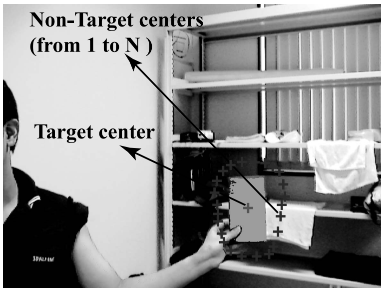


Fig. 2 A sample of the previous method.

ture vector $\mathbf{f} = (Y, U, V, x, y)$ to describe its color similarity and spatial approximation in the image. Therefore, as shown in **Fig. 2**, the feature vector of the target center is $\mathbf{f}_{(T)} = (Y_{(T)}, U_{(T)}, V_{(T)}, x_{(T)}, y_{(T)})$ and the centers of none-target points are $\mathbf{f}_{(nj)} = (Y_{(nj)}, U_{(nj)}, V_{(nj)}, x_{(nj)}, y_{(nj)})$. Here, $j = 1 \sim N$ is the index of the non-target center.

So, at time t, the distance $(d_T(t))$ from an unknown point $\mathbf{f}_{(u)}$ to the target center $\mathbf{f}_{(T)}$ and the minimum distance $(d_{NT}(t))$ from unknown point to non-target centers $\mathbf{f}_{(nj)}$ are expressed as follows:

$$d_T(t) = \|\mathbf{f}_{(u)}(t) - \mathbf{f}_{(T)}(t)\|^2, \quad (1)$$

$$d_{NT}(t) = \underset{j=1\sim N}{argmin}\|\mathbf{f}_{(u)}(t) - \mathbf{f}_{(nj)}(t)\|^2, \quad (2)$$

and this clustering procedure will produce a set of feature vectors $T(t)$ as target pixels:

$$T(t) = 1 \quad if \quad \{d_T(t) < d_{NT}(t)\}. \quad (3)$$

At time $t+1$, if the target was lost, we can achieve failure detection as:

$$\text{if} \quad \|\mathbf{f}_{(T)}(t+1) - \mathbf{f}_{(T)}(t)\|^2 > \underset{j=1\sim N}{argmin}\|\mathbf{f}_{(T)}(t+1) - \mathbf{f}_{(nj)}(t)\|^2 \quad \text{then} \quad S(t+1) = 1. \quad (4)$$

On knowing the target is lost, we can use the non-target information to achieve failure recovery (details in Ref. 21)).

The advantages of this research include: 1) Because of using feature vectors of both target point (positive data) and non-target points (negative data), it is robust against background interfusion, which makes our method different from the conventional pixel-wise tracking methods. 2) With Eq. (4), our method can achieve failure detection, which is very important for object tracking; 3) We can achieve failure recovery; 4) Because the update is performed by the K-means clustering algorithm in the 5D feature space, both the target and non-target centers are updated, thus this method has the ability to deal with the illumination changes.

2.2 Drawbacks of the Previous Method

However, there are still several drawbacks of our previous method: 1) The processing result is heavily dependent on the choosing of the initial non-target points, i.e., position, number N. From Eq. (2) we can see that the calculation cost is in proportion to N, which reduces the processing speed. 2) If many non-target points are chosen, the tracking result will be easily influenced by the background, when the background has the similar colors to the target object. 3) Since the non-target points can only translate in 2D, it can not deal with change of appearance of target object, i.e., size, pose, etc.

3. K-means Tracking with Ellipse Model

3.1 Ellipse Model

The boundary that separates the target region from the non-target region is a key feature for object tracking/detection/recognition, and image segmentation. Here we call it the "*real target boundary*" (hereafter *RTB*). However, it is very difficult or even impossible to detect the *RTB* perfectly in practice.

Here, we propose a model named "*absolute target region*" (hereafter *ATR*) to describe the target object and its surrounding background. The *ATR* is an area that contains all the target object pixels, meanwhile all the pixels on the outline of *ATR* are the representative non-target points around the target object.

The *ATR* also has the following characteristics: 1) Its outline should be described by simple and smooth closed curves; 2) It should be a good approximation to *RTB*, that means the outline of *ATR* is close to the *RTB*. The first characteristic makes the computation for object tracking easy and fast, meantime it also makes the *ATR* insensitive to the image noise or small changes of the target appearance. The second one contributes to the good accuracy of object's location during tracking.

Thus, in this paper we use an ellipse model to describe the *ATR* (**Fig. 3**), because it satisfies the above mentioned two characteristics. Meantime, it has the five parameters to represent the shape and direction of the target object, which is important for object tracking. In this paper: 1) We use the ellipse center to represent the target cluster center and the continuous points on the ellipse contour are used to represent the non-target points around the tar-

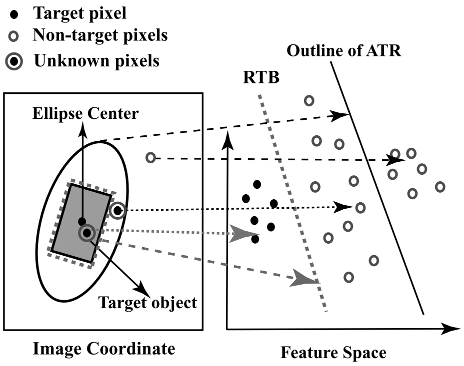


Fig. 3 The ellipse model for describing *ATR*.

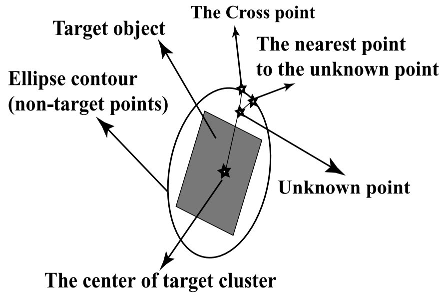


Fig. 4 2_∞*-means Clustering.*

get object; 2) The ellipse model will be updated according to the target clustering result (details in Section 4); 3) We use the ellipse region to restrict the searching area. Target detection is performed by using a K-means clustering algorithm within this region to classify each pixel as the target cluster or non-target cluster. Thus the ellipse does not need to be fitted to the target object, and the background pixels in the ellipse will not affect the performance of object tracking.

3.2 Local Color Smoothness Assumption

In general, it is reasonable to assume that the differences of the brightness and colors between two pixels are small, if the 2D geometric distance between them is small. Although this will not be true for pixels on the image or color edges, the number of such edge pixels can be considered as infinitesimal compared with the total number of pixels in an image. Such exceptions will have very little contribution to the final result of a statistics based processing with this assumption, thus they can be ignored without any problems.

3.3 2_∞-means Clustering

Supposing that we have already had an ellipse model for describing the *ATR* of the target object, we will know that all the target pixels are inside the ellipse and the ellipse outline can separate the region containing all the target pixels from its vicinity. However, since this ellipse is an approximation of the *RTB*, there are some non-target pixels included in the ellipse region. In order to ensure the correctness of object tracking, it is necessary to detect those non-target pixels and all target pixels in the ellipse region.

Because the property of target pixels is known (or pre-defined), it is easy to obtain a measurement to indicate how well a pixel is similar to be a target pixel. However, since the background around the target object changes continuously it is impossible to have a fixed (or pre-defined) description about the non-target pixels. Moreover, it is impossible to know how many clusters is enough to describe non-target pixels.

In the image, the ellipse contour consists of E points, if the ellipse size is large, the E can be a huge number. And the computational cost of Eq. (2) will become $N = E \to \infty$, in this case it is difficult to achieve the object tracking at the video-rate speed.

Because the ellipse outline is a good approximation of the *RTB*, the 2D distance between any non-target pixel in the ellipse region and the nearest point on the ellipse contour should be small. Based on the local color smoothness assumption, the differences of the brightness and the color between any non-target pixel and the nearest pixel on the ellipse outline should also be small.

If a pixel is a non-target pixel, it should belong to the same cluster as the nearest pixel on the ellipse contour belongs to.

Therefore, an unknown pixel only needs to be compared between the target cluster center and the nearest pixel on the ellipse contour to tell whether the unknown pixel is target pixel or not (See **Fig. 4**). This is indeed a "*two means clustering*", and the computation in Eq. (2) will be:

$$d_{NT}(t) = \|\mathbf{f}_{(u)}(t) - \mathbf{f}_{n1}(t)\|^2 \qquad (5)$$

In this case, the computational cost for non-target points clustering is reduced from $E \to \infty$ to 1. And the whole clustering becomes $1{+}1_\infty$-means clustering (Here, the 1 before "+" means the target cluster number and the 1 after "+" means the non-target cluster number. 1_∞ means we select one pixel from $E \to \infty$ to calculate). For this reason, we call this approach as 2_∞-*means clustering*. As shown in Fig. 4, because the 2D distance from an unknown point to its nearest point on the ellipse is short, the distance from the nearest point to the cross point is also short. The cross point is determined by the ellipse contour and the line connected by the ellipse center and the unknown pixel. Thus f_{n1} in Eq. (5) can be replaced by f_{nc}. This makes the 2_∞-means clustering even faster.

When the background contains similar colors to the target object, because only the cross point is calculated, the influence of such background pixels will be restricted to the pixels on the ellipse radius (While, as for our previous K-means clustering method, all the pixels in the searching area will be affected). Therefore, with this 2_∞-means clustering, the robustness of our new method has been greatly improved.

4. Ellipse Model Adaptation

The ellipse model, which describes the search area in the next frame, has to be adjusted according to the detected target pixels with the K-means clustering. This is because both the shape and the position of the target may change during tracking,

In order to determine the ellipse model, we need a description about the target pixels. A polygon or a closed curve will not suitable for this purpose because the target shape may be complex and its boundary may be ambiguous. On the other hand, during tracking, the appearance of the target object in the image is influenced by many factors such as the lighting condition, the material and the pose of the target object, image noise, and so on. Therefore, the target pixels can be considered as a random distributed point set. Here we use a probability distribution to describe the target pixels in the image. The probability distribution is described by a Gaussian probability density function (*pdf*) approximately. This is reasonable because most tracking targets are opaque solid object and they will appear in a restricted area and not scatter everywhere in the image. Also, in engineering field Gaussian *pdf* is widely used for describing random distributions, which often leads to stable and robust systems.

The target pixel description with Gaussian *pdf* also has the following advantages: 1) It is robust to the small deformation of the target; 2) It reduces the influence of mis-detected target pixels caused by image noise, etc.

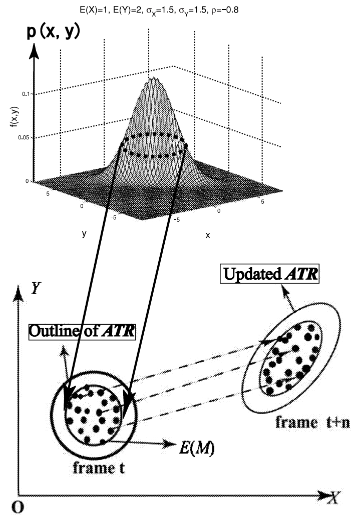


Fig. 5 The outline of ATR and the ellipse updated by Mahalanobis distance.

Here the Gaussian *pdf* of a random vector $\mathbf{Z} = [Z_1, Z_2.....Zn]^T$ is defined as:

$$Z \sim \mathcal{N}(\mathbf{m}_Z, \Sigma_Z), \qquad (6)$$

in this paper, we make $Z_i = \{x_i, y_i\}^T$ (x_i and y_i represent the detected target pixels), $\mathbf{m}_Z$ is the mean and Σ_Z denotes the covariance matrix. The Mahalanobis distance of a vector $\mathbf{Z}$ to the mean $\mathbf{m}_Z$ is given by

$$g(\mathbf{Z}) = [\mathbf{Z} - \mathbf{m}_Z]^T {\Sigma_Z}^{-1} [\mathbf{Z} - \mathbf{m}_Z]. \qquad (7)$$

The minimum ellipse ($E(M)$) that contains at least M% of the target pixels is given by

$$g(\mathbf{Z}) = J, \qquad (8)$$

where $J = -2\ln(1 - \frac{M}{100})$.

We let M be big enough (e.g., 95) so that $E(M)$ will contain 95% target pixels. $\frac{M}{100} = 0.95$ is the probability of $Z_i = \{x_i, y_i\}^T$ that will fall in $2|\Sigma_Z|$ area centered at $\mathbf{m}_Z$. The outlier out of that area will be filtered as noises.

The outline of ATR is obtained by enlarging $E(M)$ by k times (e.g., $k = 1.25$, it is the best value by our experience) to ensure that all target pixels will be included in ATR and to leave enough searching area so that most target pixels will still be in ATR at the next frame (see **Fig. 5**).

5. Experiments

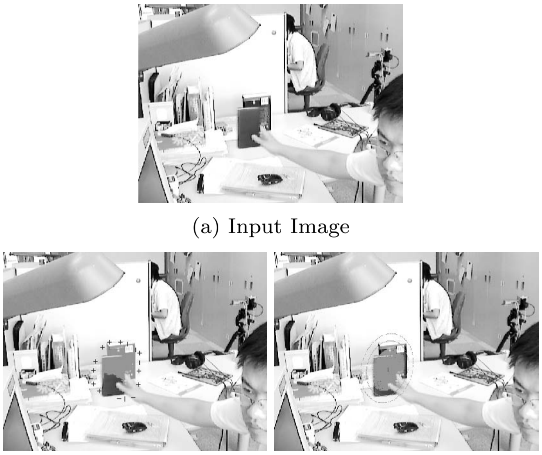
5.1 Initialization and Tracking

In the first image, we select any pixel in the target object as the target point and take another pixel outside the object as the non-target point. We consider the target point as the center, the distance from target point to the non-target point as radius to make a circle as the initial ellipse (the circle must include all the pixels of the target object).

While tracking, the shape and position of the ellipse is updated by clustering result.

5.2 Effectiveness of our New Method

As shown in **Fig. 6** (a), the target object is a small blue notebook. In the background, there is a book parts of which contains similar colors to the target. As for our previous method, since all the predefined non-target points will be calculated (Eq. (2)), when the non-target points contain similar colors to the target object, all the pixels in the defined mask [21] will be greatly affected as shown in Fig. 6 (b). As for our new method, since 2_∞-means clustering (Eq. (5)) is applied, only the cross point on the ellipse contour will be considered. When the cross point has similar color to the target object, only the pixels on the ellipse radius will be affected. Thus the robustness of our new method is greatly improved as shown in Fig. 6 (c).



(a) Input Image

(b) Our Previous Method (c) Our New Method

Fig. 6 Effectiveness of the 2_∞-means clustering.

5.3 Comparative Experiments

To evaluate the performance of our new method, we performed the comparative experiments among the following algorithms:

Method (1) Sum of Absolute Difference (SAD) based template matching,
Method (2) Mean-shift object tracking,
Method (3) Previous K-means tracking,
Method (4) Our new K-means tracking.

Figures 7, **8** and **9** are arranged as: Row 1: SAD-based template matching; Row 2: Mean-shift method; Row 3: Our previous K-means tracking; Row 4: Our present K-means tracking.

We have used many kinds of objects to test our system. Some of the tested objects are: a comb, mosquito coil and high-speed moving

(a) Frame 001 (b) Frame 053 (c) Frame 098 (d) Frame 128 (e) Frame 156

Fig. 7 Comparative experiment with Comb.

(a) Frame 002 (b) Frame 041 (c) Frame 132 (d) Frame 188 (e) Frame 234

Fig. 8 Comparative experiment with mosquito coil.

(a) Frame 002 (b) Frame 041 (c) Frame 132 (d) Frame 188 (e) Frame 234

Fig. 9 Comparative experiment with high-speed moving head.

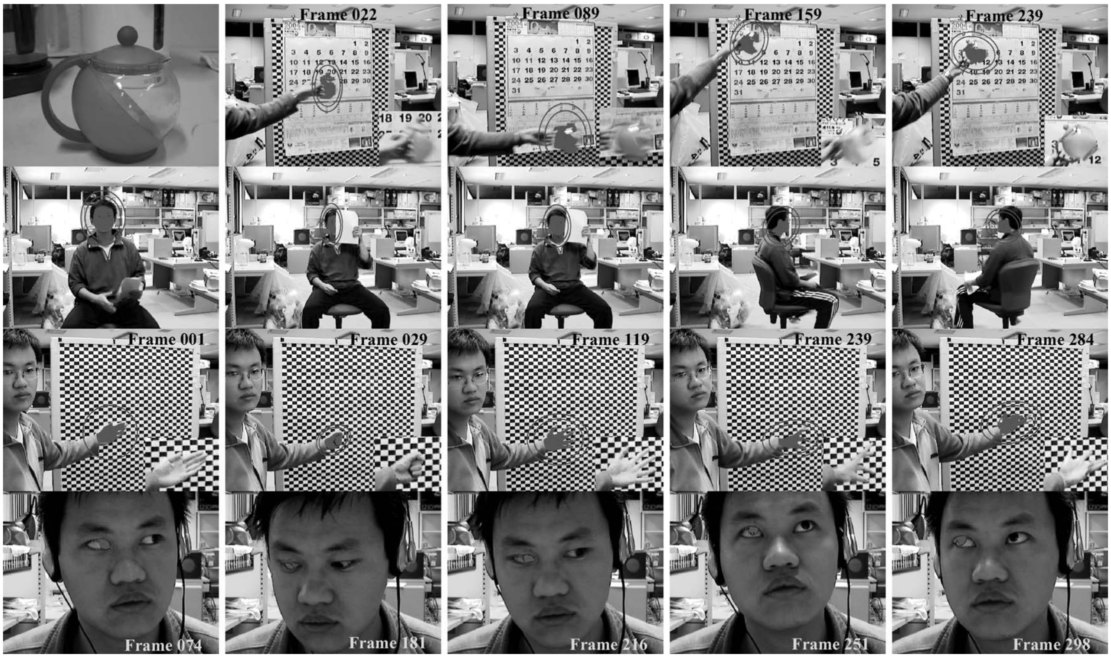


Fig. 10 Some tracking results of our K-means tracker. Row 1: Kettle moved at high-speed. Row 2: Head with scaling, partially occlusion, and revolution. Row 3: Hand with topological changes. Row 4: Moving eye and head.

head. Since comb and mosquito coil have apertures, the background interfusion makes them become the challenging objects. The high-speed moving head contains shape changing and 3D transformation. From the comparative experiments, we could see:

In Figs. 7 and 8, because of the apertures in the object, the background was always interfused into the template. While tracking, when the newly interfused background was different from the initial background:

As for Method (1), the background interfusion changed the texture in the template;

As for Method (2), the background interfusion changed the color histogram in the template;

Method (3) could not deal with 3D transformation.

So all of them finally failed.

On the contrary, Method (4) could track successfully, because it used pixel-wise method and updated the ellipse contour dynamically. In Fig. 7 Row 4, since the ellipse shape in frame $t+1$ is updated according to the target clustering result in frame t, in the frame $t+1$, if the target size shrinks rapidly, the ellipse size may be too large compared to the target object. However, unless the object moves out of the ellipse, our method will work without any problem.

In Fig. 9, Method (1), (2) failed because of the feature changing caused by 3D transformation and the distortion of high-speed moving head. Method (3) suffered from 3D transformation and too many non-target points definition. And only Method (4) tracked fast and robustly.

Through Figs. 7, 8, 9, we can see that our new K-means tracking method has successfully resolved the problem caused by the rotation ☆ and scaling ☆☆ of the target object. By using the variable ellipse model, our new method is robust against the 3D deformation of the target object.

In **Fig. 10**, we tested our new K-means tracking method with a kettle (whose moiety is transparent), a head, a hand and eyes. For the kettle, even the background had the similar colors to the object, the proposed method could track successfully. The head tracking experiment dealt with scaling, partly occlusion, and revolution. Hand tracking experiment dealt with various transformations from the initial pose such as fisting, scratching, topological

☆ Fig. 7 (Frame 53); Fig. 9 (Frame 41)

☆☆ Fig. 9 (since Frame 132)

changes, which were difficult for the contour-based method to track. The eyes experiment is an example for the illumination changes. The experiments with head, hand and eyes showed the potential application of our algorithm in human interface.

All the experiments were taken with an Intel Xeon 3.06 GHZ CPU. The image size was 640×480 pixels, and when the target size was between 140×140 and 200×200 pixels, the processing time was from $0.014 \sim 0.02$ sec/frame.

6. Conclusion

In this paper, we present an improved k-means tracking method under color image sequence. Inheriting the advantages of our previous system, with a single target point and a variable ellipse model which is adaptively controlled by the target detection result. We improve our tracking system in speed, stability and the scale or shape of the target object. Real-time K-means tracking becomes possible.

Acknowledgments This research is partially supported by the Ministry of Education, Culture, Sports, Science and Technology, Grantin-Aid for Scientific Research (A)(2), 16200014 and (C)(2) 16500112.

References

1) Crane, H.D. and Steele, C.M.: Translation-tolerant mask matching using noncoherent reflective optics, *Pattern Recognition*, Vol.1, No.2, pp.129–136 (1968).
2) Gr*äβ*l, C., Zin*β*er, T. and Niemann, H.: Illumination Insensitive Template Matching with Hyperplanes, *DAGM*, pp.273–280 (2003).
3) Comaniciu, D., Ramesh, V. and Meer, P.: Real-time Tracking of Non-rigid Objects using Mean Shift, *CVPR*, Vol.2, pp.142–149 (2000).
4) Comaniciu, D., Ramesh, V. and Meer, P.: Kernel-Based Object Tracking, *PAMI*, Vol.25, No.5, pp.564–577 (2003).
5) Heisele, B.: Motion-based Object Detection and Tracking in Color Image Sequence, *ACCV* (2000).
6) Heisele, B., Kre*β*el, U. and Ritter, W.: Tracking Non-Rigid Moving Objects Based on Color Cluster Flow, *CVPR*, pp.253–257 (1997).
7) Khan, Z., Balch, T. and Dellaert, F.: An MCMC-based Particle Filter for Tracking Multiple Interacting Targets, *ECCV*, Vol.4, pp.279–290 (2004).
8) Zhao, T. and Nevatia, R.: Tracking Multiple Humans in Crowed Enviroment, *Pro CVPR'04*, Vol.2, pp.406–413, Washington, D.C (2004).
9) Siebel, N.T. and Maybank, S.: Fusion of Multiple Tracking Algorithms for Robust People Tracking, *ECCV*, Vol.IV, pp.373–387, Springer-Verlag (2002).
10) Vermaak, J., P*é*rez, P., Gangnet, M. and Blake, A.: Towards Improved Observation Models for Visual Tracking: Selective Adaptation, *ECCV*, Vol.1, pp.645–660 (2002).
11) Wu, Y., Hua, G. and Yu, T.: Switching Observation Models for Contour Tracking in Clutter, *CVPR*, Vol.1, pp.295–302 (2003).
12) Li, P. and Zhang, T.: Visual Contour Tracking Based on Particle Filter, *Image Vision Computing* (2003).
13) Sidenbladh, H. and Black, M.J.: Learning Image Statistics for Bayesian Tracking, *ICCV*, Vol.2, pp.709–716 (2001).
14) Tao, H., Sawhney, H.S. and Kumar, R.: Object Tracking with Bayesian Estimation of Dynamic Layer Representations, *PAMI*, Vol.24, No.1, pp.75–89 (2002).
15) Agarwal, A. and Triggs, B.: Tracking Articulated Motion using a Mixture of Autoregressive Models, *ECCV*, Vol.3, pp.54–65 (2004).
16) Toyama, K. and Blake, A.: Probabilistic Tracking in a Metric Space, *ICCV*, Vol.2, pp.50–57 (2001).
17) Hartigan, J. and Wong, M.: Algorithm AS136: A K-Means Clustering Algorithm, *Applied Statistics*, Vol.28, pp.100–108 (1979).
18) Yilmaz, A., Li, X. and Shah, M.: Object Contour Tracking Using Level Sets, *ACCV* (2004).
19) Hartigan, J.: Clustering Algorithms, *John Wiley & Sons*, New York, NY (1975).
20) Isard, M. and Blake, A.: Contour tracking by stochastic propagation of conditional density, *ECCV*, Vol.1, pp.343–356 (1996).
21) Wada, T., Hamatsuka, T. and Kato, T.: K-means Tracking: A Robust Target Tracking against Background Involution, *MIRU*, Vol.2, pp.7–12 (2004)

(Received January 31, 2005)
(Accepted July 18, 2005)

(Editor in Charge: *Ryo Kuratsume*)

Chunsheng Hua was born in 1978 and received his BE degree in electronic engineering from Shenyang Polytechnic University in 2001. He received his MS degree from Department of Mechanical and System Engineering at Kyoto Institute of Technology. He started his Ph.D. course since 2004 until now. His research interests include face recognition, color-based object tracking.

Haiyuan WU received the Ph.D. degree from Osaka university in 1997. From 1996–2002, she was research associate at Kyoto Institute of Technology. Since 2002, she has been associate professor at Wakayama university. She is a member of IPSJ, IEICE, ISCIE and the Human Interface.

Toshikazu WADA received the BEng degree in electrical engineering from Okayama University, the MEng degree in computer science from Tokyo Institute of Technology, and the DEng degree in applied electronics from Tokyo Institute of Technology, in 1984, 1987, and 1990, respectively. He is currently a professor in the Department of Computer and Communication Sciences, Wakayama University. His research interests include pattern recognition, computer vision, image understanding, and artificial intelligence. He received the Marr Prize at the International Conference on Computer Vision in 1995, Yamashita Memorial Research Award from the Information Processing Society Japan (IPSJ), and the Excellent Paper Award from the Institute of Electronics, Information, and Communication Engineers (IEICE), Japan. He is a member of IEICE, IPSJ, the Japanese Society for Artificial Intelligence, and IEEE.

Qian CHEN received the Ph.D. degree from Osaka university in 1992. From 1992–1994, he was a researcher at Laboratory of Image Information and Science and Technology. From 1994–1995, he was research associate at Osaka university. From 1995–1997, he was research associate at Nara Institute of Science and Technology. Since 1997, he has been associate professor at Wakayama university. He is a member of IPSJ and RSJ.