社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2005－CVIM－151 (1)
2005／11／17

K-means tracker: A multiple colors object tracking algorithm

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

This paper presents a K-means tracker, which is a novel visual tracking algorithm. This algorithm is robust against the interfused background, because it discriminates "target" pixels from "background" pixels in the tracking region by applying K-means clustering to both the positive and negative information of the target. To ensure the robustness of this algorithm, we apply the following ideas: 1) To represent the color similarity and spatial approximation of the target simultaneously, we use a $5D$ feature vector consisting of the position (x, y) and color (y, u, v) information for object tracking. The object tracking is performed and updated not only in the image space but also in the color space at the same time. Therefore, this adaptive nature guarantees our method to be robust against the target color change. 2) By using a variable ellipse model to restrict the target search area and represent the non-target pixels surrounding the target, the algorithm can cope with the changes of the scale and shape of the target object flexibly. 3) Even the tracking sometimes fails, this algorithm can automatically discover and recover from the tracking failure based on the positive and negative information. To capture motion-blur-free images of a moving object at video rate, we control a set of active cameras which are mounted on the pan-tilt units according to the results of the K-means tracker.

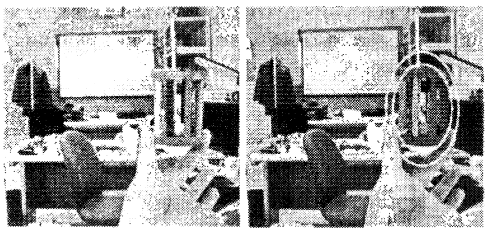

Fig. 1 An example result of tracking a multiple-color object in which background pixels are interfused.

1. Introduction

This research realizes an algorithm for tracking an object with multiple colors in real time. It concentrates on solving the problem caused by the interfused background that many tracking algorithms suffer from. Especially, this research realizes tracking failure detection and automatic recovery from the failure. A snapshot of our results is shown in **Fig.**1.

In the last two decades, numerous powerful algorithms for object tracking were developed[8)-26)]. Most of them can be mainly described according to three aspects: *searching area, target representation and similarity measurement.*

Template matching[12),13)] is one of the most famous object appearance model based tracking algorithm. It searches for the area in the input image that is the most similar to the registered template of the object. Because the algorithm using a fixed template will fail easily when the appearance of the object changes, algorithms using updatable template were introduced. However, at each update of template, some background pixels will be interfused into the template, which will cause the template to drift away from the object.

Comaniciu *et al* introduce the mean-shift algorithm[14)] (or kernel-based tracking[15)]) that uses a color histogram to describe an object. Since a color histogram is a viewpoint-insensitive feature, the mean-shift algorithm can track objects with pose changes and non-rigid deformation. However, since the estimation of the similarity between two colors histograms are unstable when the color distribution of an object is sparse or narrow (e.g. a planar monochromatic object), the performance of the algorithm becomes poor for tracking such objects.

Condensation[16)] (also called as particle filter, or MCMC[17)~20)]) realizes stable object tracking through cluster and occlusion by reasoning over a non-parametric distribution of joint state probability of multiple hypotheses.

Active contour[21)] performs well when tracking non-rigid object, because it can represent the precise target shape according to the target movement. However, this method requires that the object contour has to be defined or trained before tracking, which is not always possible in practice.

Because pixel-wise tracking algorithms[24),25)] do not use any object appearance models, they have more flexibility to cope with the deformation or the changes of scale and pose of objects. Most pixel-wise algorithms make use of the target information, but the precious non-target (background) informa-

† Wakayama University

tion is disregarded.

Through our experiment with the current tracking algorithms, we found out that the ability to discriminate the target from its surrounding background is the key element that finally resolves the tracking success or failure. However, the background surrounding the target may hardly have the fixed property because it will keep on changing to follow the target movement, time flowing, illumination changes and so on. Therefore, it is reasonable to think the following description will be an effective tracking algorithm: the target tracking should be accompanied with the continuous background discrimination, meanwhile the background property should also be continuously updated. Similar algorithms that make use of non-target information[8)-11)] have been introduced. With a stationary camera, moving objects can be effectively tracked in real-time by adaptive background subtraction[8)]. It can be generalized to situations where the video data can be easily stabilized, including purely rotating and zooming cameras, and the background can be modeled as an approximately planar surface[9)].

Collins *et al*[10)] propose a method for selecting the image feature that discriminates the object from its surrounding most conspicuously through a pre-defined feature set during tracking. Hieu *et al*[11)] propose a tracking algorithm by a discriminative model. In that method, the object and its surrounding background are divided into small patches, which are described by a set of Gabor features. A LDA classifier is trained by the object and background patches. However, during the update of object/background templates, there is no guarantee that an unseen patch can be classified correctly.

Most methods assume that the target is a solid appearing object, which can be described by an entire area (e.g. a rectangle or an ellipse) surrounded by background. However, in practice, many objects can not be described in that way. For example, human bodies, tea pots, steel towers of power line or bicycles have apertures. For such objects, background appears not only around the object but also through its apertures. Therefore, methods that consider the area of target object as an entire one will fail because some background may be mis-regarded as target. While pixel-wise method classifies target and background at pixel-level, so it can eliminate the influence of background interfused through apertures..

As well known, object tracking may fail even with a powerful algorithm. We consider that a mechanism for detecting tracking failure is indispensable to successful object tracking. Having this mechanism, the tracking failures detection and recovery will become possible, e.g. by searching for the object in a wider area.

In[7)], we have proposed a tracking algorithm based on the K-means clustering and a variable ellipse that represents for the background information (*hereafter called as the non-target information*). In that paper, we use the variable ellipse model to represent the target shape, and meantime this ellipse also restricts the searching area. K-means clustering is used as the similarity measurement to discriminate the target pixels from those non-target ones. Since the pixel-wise method is applied, the K-means tracker can work well on the target with interfused background. This tracking algorithm also achieves the self tracking failure detection and recovery automatically, because both the target and non-target information are applied. Because both the target color and position are simultaneously updated in a $5D$ feature space, this K-means tracker especially work well against the illumination changes.

In this new work, we mainly improve our system against several problems remained in[7)] as the following aspects:

(1) The previous monochromatic tracker is extended to the multiple-color tracker (N colors, theoretically $N = \infty$).

(2) To deal with object with special structures (such CD), the original tracking failure detection is replaced by an improved one, which will exam both the target centers and the non-target pixels on the ellipse contour.

(3) After the tracking failure recovery, the recovered colors are compared with the pre-defined ones through a Bayesian formulation. This step will confirm us that the recovered colors are correct or not.

To capture motion-blur-free images of a moving object at video rate, we control an active camera mounted on a pan-tilt unit according to the results of the K-means tracker.

2. Multi-color Tracking Algorithm

In[7)], the monochromatic target (or object containing similar colors) could be stably tracked with a variable ellipse (in **Fig**.2 we call it the *Absolute Target Area*, hereafter ATA) to represent the non-target pixels surround the target and restrict the searching area. However, this algorithm is limited to the monochromatic object, in the real world, most objects that contain multiple colors can not be processed by this method.

2.1 Basic Algorithm

Since pixel-wise tracking algorithm does not need the object appearance model, it can track the non-rigid object and ware object well. However, most of the present pixel-wise methods only pay attention to the target property, the object tracking in those works is performed by searching for a region that has the highest similarity to the target object. In such case, whether the tracked object is correct or not is often performed by comparing the object's similarity with an absolute threshold, which is not robust under continuous changing condition.

On thinking of this problem, we introduce the concept of negative samples into the pixel-wise tracking algorithm. With the negative samples, whether an object is the target or not is established by comparing its similarity to the positive samples (here, it is the target centers) with that to the negative samples (non-target center). In such case, the similarity measurement becomes dynamic because both the positive and negative samples are keep on changing, which lead our method to be able to work robustly under different conditions. In this paper, the similarity measurement is performed by comparing the distance to the positive and negative centers, other measurement can also be adapted.

In this algorithm, each pixel in an image is described by a 5D feature vector $\mathbf{f} = [\mathbf{c}\ \ \mathbf{p}]^T$, where, $\mathbf{c} = [Y\ U\ V]^T$ describes the color similarity and $\mathbf{p} = [x\ y]^T$ describes the position approximation of the pixel. As shown in **Fig.**2, we assume that a target object has N colors. If the mean of the target pixels having i^{th} color in the 5D feature space has been obtained, we call it as the target cluster center and describe it by $\mathbf{f}_T(i) = [\mathbf{c}_T(i)\ \mathbf{p}_T(i)]^T$, $i = 1 \sim N$. Non-target pixels on the ellipse are presented as $\mathbf{f}_{Ne}(j) = [\mathbf{c}_{Ne}(j)\ \mathbf{p}_{Ne}(j)]^T, j = 1 \sim m$, where m is the number of the selected pixels on the ellipse, and an *unknown* pixel is described by $\mathbf{f}_u = [\mathbf{c}_u\ \mathbf{p}_u]^T$.

An unknown pixel $\mathbf{f}_u$ can be classified by comparing the shortest distance d_T from it to the N target cluster centers

$$d_T = \underset{i=1\sim N}{argmin}\left\{\|\mathbf{f}_T(i) - \mathbf{f}_u\|^2\right\}, \qquad (1)$$

and the shortest distance d_N from $\mathbf{f}_u$ to the m pixels on the ellipse

$$d_N = \underset{j=1\sim E}{argmin}\left\{\|\mathbf{f}_{Ne}(j) - \mathbf{f}_u\|^2\right\} \qquad (2)$$

If $d_T < d_N$, $\mathbf{f}_u$ is classified as a target pixel, otherwise a non-target pixel.

2.2 Selection of Non-target pixels

There $m = 9$ non-target pixels selected from the ellipse contour. 8 of them are determined by the 8-equal division of the ellipse at 45° interval. The rest one is the cross point shown in **Fig.**2. For an unknown pixels, its cross point is performed by the ellipse contour and the radius connected by the unknown pixel and the ellipse center.

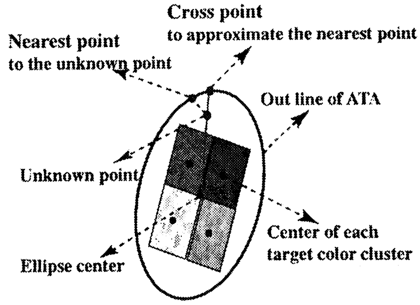

Fig. 2 The ellipse model for clustering multi-color target

While the tracking process, since the object keeps on moving, the background non-target pixels surround it will also be continuously moved. Therefore, it is important that the location of non-target pixels should be changeable. In this paper, the non-target pixels are dynamically updated according to the distribution of the detected target pixels. This ability guarantees our algorithm to deal with various geometric transformations, such as scaling, rotation, revolution, etc. The update of non-target pixels (it can also be called as the update of ATA) can be found in[7].

2.3 Update of Target Centers

During each iteration of K-means clustering, the detected target centers are updated in both the 2D position space and the 3D color space to follow the illumination changes. However, since rapid color shifting (e.g. reflecting) will destroy the target property, we use a simple averaging filter to gradually decrease the influence of rapid color changes as:

$$f_T^{(t)}(i) = \gamma f_T^{(new)}(i) + (1-\gamma) f_T^{(Init)}(i) \qquad (3)$$

$0 < \gamma < 1$ is the pre-defined coefficient(e.g. 0.6 an experimental value) , the superscript (t), (new), $(Init)$ denote the time, $f_T^{(new)}(i)$ represents the clustering result of i^{th} target color in frame t, $f_T^{(Init)}(i)$ means its initial feature. With this equation, when rapid color shift happens, our method can resist such influence, and when the target color turns back it will continue the correct process.

3. Self Tracking Failure Detection

As shown in **Fig.**3, we define that: in the current frame if the target center disappears from where it has been in the previous frame, we consider the tracking failure has occurred.

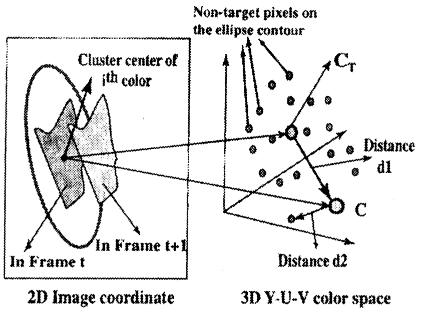

Fig. 3 Tracking failure detection for Target color

In **Fig.**3, we let $\mathbf{c}_T$ be the color of i^{th} color cluster center (x_i, y_i) in the previous frame and $\mathbf{c}$ be the color of the pixel at (x_i, y_i) in current frame, $d_1 = ||\mathbf{c}_T - \mathbf{c}||^2$ and $d_2 = \underset{j=1\sim m}{argmin}||\mathbf{c}_T - \mathbf{c}_N(j)||^2$. The failure of tracking i^{th} color cluster can be detected by checking if $d_1 > d_2$ stands. If this assumption stands, we let $failure(T) = 1$.

However, in fact it is quite easy for the 2D centers of the target colors to be overlapped at the same point or the 2D target center just lies at the apertures of target object (like the CD). Meanwhile, because the background may contain similar or the same color as that of the target, if only the target centers are examined, it will easily to the unnecessary or the wrong failure detection result. Therefore, we consider that not only the target center but the non-target pixels on the ellipse contour must be examined to check if the tracking failure happens or not.

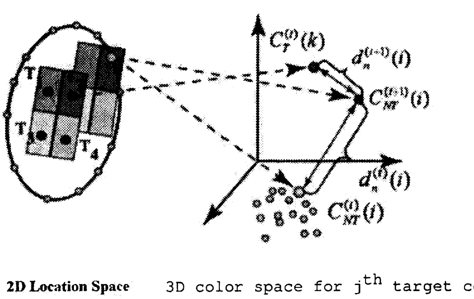

Fig. 4 Tracking failure detection for Non-target color

As shown in **Fig.**4, we consider that in the normal case the selected n points non-target pixels (here $n = 36$ it means these sample points are selected at 10° interval from the ellipse contour) should be far away from the target centers in the 3D color space, and if over 40% of the n non-target points are closer to a target center than to their original color position, in such case we define the color detection for non-target colors is failed . Only if both the target detection and the non-target detection are failed simultaneously, we will say the target object is lost.

In **Fig.**4, $C_T^{(t)}(k)$ denotes the k^{th} target color center in frame t, $C_{NT}^{(t)}(i)$ denotes the color of i^{th} non-target points ($i = 1 \sim n$) in frame t, and $C_{NT}^{(t+1)}(i)$ means the color of the same non-target point mentioned above in frame $t+1$. $d^{(t)}(i)$ is the color distance from $C_{NT}^{(t)}(i)$ to $C_{NT}^{(t+1)}(i)$ and $d^{t+1}(i)$ is the distance from $C_{NT}^{(t+1)}(i)$ to $C_T^{(t)}(k)$. When $d^{(t+1)}(i) < d^{(t)}(i)$, it means that in frame $t+1$ the color contained by the i^{th} selected non-target point has higher probability to the target color, at this time we defined this non-target point is failed and let $failure_{NT}(i) = 1$ The whole tracking failure detection is performed as:

$$P(fail) = 0.5 * failure(T) + 0.5 * \frac{\Sigma_{i=1}^{n} failure_{NT}(i)}{n} \tag{4}$$

When $P(fail)$ is over 0.7 (experimental value), we will say that target object is lost. In the next section will describe the method to recover from this tracking failure.

4. Failure Recovery

Assuming target object does not move too fast, the j^{th} color cluster of it should not completely disappear from the *ATA* determined in the previous frame. When the tracking failure of j^{th} color has been detected, it can be recovered by finding $\mathbf{f_p}$ that has the minimum $d^{(t+1)}(i)$

$$d(\mathbf{f_p}) = \underset{i=1\sim E}{argmin}\ d^{(t+1)}(i) \tag{5}$$

here, $d^{(t+1)}(i)$ is the 5D distance from the previous target center to any pixel, $E \in ATA$ means the number of all pixels within the ellipse, the pixel that contains $\mathbf{f}_p$ will be considered as the recovered new j^{th}target center.

The reliance of the recovered cluster center of j^{th} color is verified by using a Bayesian probability formulation. Since the initial colors are recorded at the beginning of object tracking, the posterior probability of k^{th} ($k = 1 \sim N$) recorded color given the

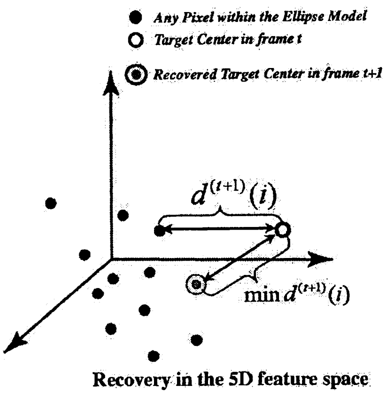

Fig. 5 Failure Recovery for j^{th} Target Center

recovered color is calculated as

$$P(\mathbf{c}(k)|\mathbf{c}_{rec}) = \frac{p(\mathbf{c}_{rec}|\mathbf{c}(k))P(\mathbf{c}(k))}{\sum_{i=1}^{N}\{p(\mathbf{c}_{rec}|\mathbf{c}(i))P(\mathbf{c}(i))\}}, \quad (6)$$

where $\mathbf{c}_{rec}$ is the recovered color, $p(\mathbf{c}_{rec}|\mathbf{c}(i))$ can be estimated by assuming a normal distribution of i^{th} color, $P(\mathbf{c}(k))$ is the ratio of the area of k^{th} color cluster to the area of the target object.

The verification is carried out by finding k that maximizes Eq.(6) and checking if $k = j$. If $k \neq j$, the recovery is considered as a failure.

When the target pixels can not be described by a convex solid pattern, false alarm may be raised during tracking failure detection. Although a false alarm will cause an unnecessary tracking failure recovery, it will not do harm to the performance of object tracking. Only if all the N colors of target object are lost and none of them can be recovered, our K-means tracker will fail. But there is very little probability that such case will occur, therefore, our algorithm can work robustly.

5. Active Cameras Controlled by K-means Tracker

It is difficult to monitor a moving object with a still camera. Since the object moves, its image may be blurred or out of focus, its size in the image may be improperly large or small. Moreover, it may disappear from the image. (See **Fig.**6)

Here, we present a new method for capturing high quality images of an object that moves fast in a complex background by using two active cameras whose pan, tilt, focus and zoom can be controlled. The "high quality image sequence" here means that the image of the object is in focus and not blurred, the S/N ratio is high enough (the image is not too dark) and both the size and the position of the object in the image is kept unchanged. (See **Fig.**7)

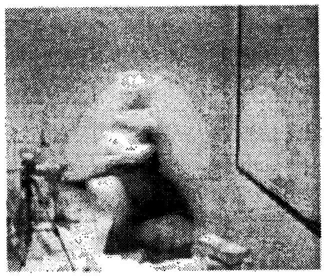

Fig. 6 Moving object image taken from the still camera

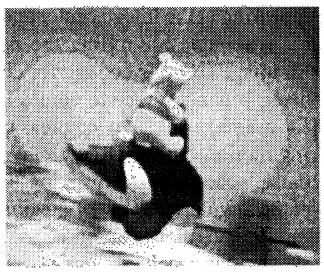

Fig. 7 Moving object taken by our active camera

To achieve this goal, we use the output of the K-means tracker to control the angular position and speed of the pan-tilt units by employing PID control scheme. The purpose of this control is to make the object appearing at the image center and its speed in image space to be zero. This operation has the effect of stabilizing the object in the image. This effect makes it possible to take motion-blur-free images with relatively low shutter speed, which ensures the high S/N ratio of the image. By using two cameras, binocular stereo vision algorithm can be used to obtain 3D position and velocity of the object. These results are used for adjust the focus and the zoom to ensure that the object appeared in focus and in the desired size.

5.1 Image stabilization through visual feedback control

In order to stabilize the object in the images, we mount the camera to a motor droved pan-tilt unit. The block diagram of our system is shown in Fig.8.

The pan-tilt unit is carefully designed and adjusted to ensure that the optical center of the camera will not move the pan or tilt angle changes. We call it as "Fixed Viewpoint Active Camera". By using the K-means tracker, we can obtain the position and the velocity of the object in image space. We use a

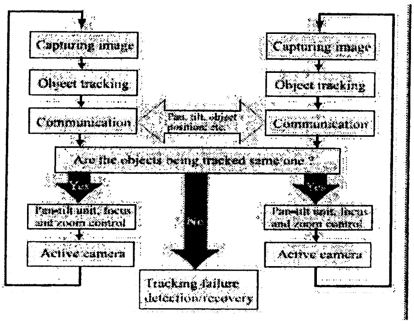

Fig. 8 Block diagram of the system

visual feedback control algorithm to make the object keep appearing at the image center and its speed being zero. Although controlling the position of the object in image space is enough for chasing the moving object, it is necessary to control the speed of the object in image space to suppress oscillation for capturing motion-blur-free images. We employ the PID control scheme for controlling the pan-tilt unit. We let the angular velocity of the object be the P component, the angular position be the I component and the angular acceleration be the D component. Then the PID control can be used for controlling the angular speed and position of the pan-tilt unit simultaneously.

5.2 Binocular active camera system

3D information of an object is very useful for stable object tracking. The 3D position can be used for adjusting focus and zoom of the camera, the 3D position and velocity can be used to predict the velocity in the image space, etc. In this research, we use binocular active stereo camera (Fig .9) to estimate 3D information of the object.

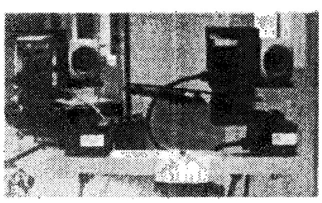

Fig. 9 Binocular Active Cameras

When the geometric relation between the two active cameras has been calibrated and the information about the rotational transform of each camera is shared by two active camera control module, it is possible to gaze at the same point on the object while tracking it and estimating its 3D position and velocity.

Also, by sharing the tracking information between the two cameras, the epipolar constraint can be used to predict the position where the object should appear at in one camera from the position of the object in the image of the other camera. Therefore, if the object tracking failed in one camera, it can start an target searching module that tries to find out the object around the position of the object predicted from the tracking information of the other camera.

6. Experiment

The K-means tracker was applied to many image sequences, some of the experiment samples are shown in **Fig.**10. Here, the evaluation of our K-means tracker is performed among the SAD template matching and the mean-shift tracking algorithm[15].

(1) SAD Template Matching: Sum of the Absolute Difference Template Matching performs the similarity measurement by comparing the sum of the absolute difference of the texture features. In our experiment, texture feature is the color information.

(2) Mean-shift algorithm[15] uses the color histogram to describe the target property. With hill-climbing method, the target position is estimated at the place that has the maximum similarity to the defined target histogram.

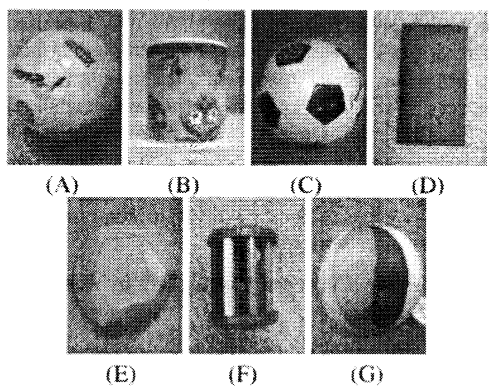

Fig. 10 Some other experiment objects.

In **Tab.**1 ☆, we show the comparative experiment results among these three methods. In this table:

(1) ○ denotes "Can Track the target well";

(2) △ represents "Can Track, but Unstably";

(3) × means "Can Not Track".

In the first column, the target object is the A in

☆ The movies of these experiments are stored at: http://www.wakayama-u.ac.jp/ wuhy/wu2_new.html

Table 1 Comparative Experiments Results

	Translation	2D Rotation	3D Rotation	Scale	Color Shift	Partial Occlusion	Ware Object	Textured lighting
SAD Template Matching	○	△	△	△	○	×	×	×
Mean-shift Algorithm	○	○	○	△	×	×	×	×
K-means Tracker	○	○	○	○	○	○	○	○

Fig.10. While translation, all these three methods can track the target well.

In the second column, the target object is the B in **Fig.**10. For the geometric 2D rotation, the SAD template matching tracked unstably because, while the update of template, background pixels were interfused into the template, which changed the texture feature and finally led to the gradual template shift.

In the third column, the target object is the C in **Fig.**10. With the 3D rotation, the SAD template matching did not track well because the template only followed the initialized special black pattern. When the football revolute, the template moved with the special black pattern and shift from the target, when the target returned back it turned back also.

In the fourth column, we made the experiment with the C in **Fig.**10. under greatly scaling. Because the texture feature would change as the object become big or small and the SAD template matching can not change the size of rectangle, at this time template matching could track the target but quite unstably. For the same reason, mean-shift algorithm did not work well, too.

In the fifth column, to exam the three methods in the case of color shift, we selected the target object as the D in **Fig.**10. Mean-shift algorithm completely failed, because, in the color histogram, monochromatic object will be a thin line or a quite narrow range. When color shift happened, this thin line was moved greatly in the color histogram, which led to the tracking failure.

In the sixth column, the comparative experiment was performed with the E in **Fig.**10 to exam the performance of the three methods in the case of partial occlusion. As for SAD template matching, when occlusion happened, the color and edge texture within the template were completely changed, therefore the object finally failed. For mean-shift method, in such case, the color within the ellipse was completely different from the initial target color, thus the tracking failed finally.

In the seventh column, the target object is the F in **Fig.**10. As for the SAD template matching, since the background pixels kept on be interfused into the template through the ware object, the accumulated position error will occur while the update of template. And finally, this accumulated position error shifted the template away from the target object. Since the background was not uniform, even the ware object moved a little, the color histogram within the ellipse would be greatly deformed. Such reason resulted in the tracking failure of mean-shift algorithm.

In the last column, the target object is G in Fig.10. The texture lighting included: in the horizontal direction continuous illumination changes happened; in the vertical direction uniform white-black pattern happened. Although SAD template matching updated the texture within the template, it failed because of the rapid pattern changes in the vertical direction, which changed the texture feature. As for mean-shift, since the vertical pattern changes were uniform and the color histogram would not be greatly changed in this direction, it could track the target at the beginning of the experiment. However, because the continuous illumination changes simultaneously happened in the horizontal direction, it failed when the target moved from the light area to the dark area for the shape of color histogram was changed.

All the experiments were taken on a desktop PC with an Intel Xeon 3.06Ghz CPU. The image size was 640×480 pixels. In the cases that the target object size verified from 140×160 to 200×230 pixels, and the number of its colors was less than six, the processing time was $0.013 \sim 0.018$ sec/frame. Through the real application experiment, our method was proved to be able to work at the video-rate speed.

7. Conclusion

In this paper, we have proposed the K-means tracker, which can track a multiple-color target by discriminating the target pixels from non-target pixels with the fastened K-means clustering algorithm.

With the concept of non-target, the proposed method can selfly detect the tracking failure by checking both the target centers and the non-target pixels on the ellipse contour. Our algorithm can also recover from the tracking failure automatically and then the recovered target color is verified by a Bayesian formulate to confirm the correctness of the recovery process. The application of both the positive (target) and negative (non-target) information makes this algorithm robust against the interfused background. We have implemented a prototype system based on the algorithm and confirmed the robustness and video-rate processing through extensive experiments.

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

References

1) B.Heisele "Motion-based Object Detection and Tracking in Color Image Sequence," *ACCV*, (2000)
2) Z.Khan, T.Balch and F.Dellaert "An MCMC-based Particle Filter for Tracking Multiple Interacting Targets," *ECCV*, Vol.4, pp.279-290, (2004)
3) K.Toyama, A.Blake "Probabilistic Tracking in a Metric Space," *ICCV*, Vol.2, pp.50-57, (2001)
4) J.Hartigan, "Clustering Algorithms", *John Wiley & Sons*, New York, NY, (1975)
5) M.Isard, A.Blake, "Contour tracking by stochastic propagation of conditional density", *ECCV*, Vol.1, pp.343-356, (1996)
6) 和田俊和, 濱塚俊明, 加藤丈和 "K-means トラッキング：背景混入に対して頑健な対象追跡法", *MIRU*, Vol.2, pp.7-12, (2004)
7) Chunsheng Hua, H. Wu, Q. Chen, T. Wada " K-means tracking with variable ellipse model", *IPSJ CVIM* Vol.146, pp.?-?, (2005)
8) Stauffer, C. and Grimson, W.E.L, " Adaptive Background Mixture Models for Real-time Tracking ", CVPR, pp.246-252 (1999)
9) Irani, M. and Anandan, P., " A Unified Approach to Moving Object Detection in 2D and 3D Scenes ", PAMI, Vol.20, No.6, pp.577-589, (1998)
10) R. Collins and Y. Liu, " On-line Selection of Discriminative Tracking Feature ", ICCV, Vol.1 pp.346-352 (2003)
11) H.T.Nguyen and A. Semeulders, " Tracking aspects of the foreground against the background ", ECCV, Vol.2 pp.446-456 (2004)
12) H.D.Crane and C.M.Steele, " Translation-tolerant mask matching using noncoherent reflective optics ", PR, Vol.1, No.2, pp.129-136, (1968)
13) C.Gräßl, T.Zinßer, H.Niemann, " Illumination Insensitive Template Matching with Hyperplanes ", *DAGM*, pp.273-280, (2003)
14) D.Comaniciu, V.Ramesh and P.Meer, " Real-time Tracking of Non-rigid Objects using Mean Shift ", *CVPR*, Vol.2, pp.142-149, (2000)
15) D.Comaniciu, V.Ramesh and P.Meer, " Kernel-Based Object Tracking ", *PAMI*, Vol.25, No.5, pp.564-577, (2003)
16) M. Isard, and A. Blake, " Condensation-conditional density propagation for visual tracking ", IJCV, Vol.29, No.1, pp.5-28, (1998)
17) T.Zhao, R.Nevatia, " Tracking Multiple Humans in Crowed Enviroment ", *CVPR*, Vol.2, pp.406-413, (2004)
18) N.T.Siebel and S.Maybank, " Fusion of Multiple Tracking Algorithms for Robust People Tracking ", *ECCV*, Vol.IV, pp.373-387, (2002)
19) J.Vermaak, P.Pérez, M.Gangnet and A.Blake, " Towards Improved Observation Models for Visual Tracking: Selective Adaptation ", *ECCV*, Vol.1 pp.645-660, (2002)
20) Y.Wu, G.Hua, T.Yu, " Switching Observation Models for Contour Tracking in Clutter ", *CVPR*, Vol.1, pp.295-302, (2003)
21) M.Kass, A. Witkin and D. Terzopoulos, " Snakes: active contour models ", *IJCV*, Vol.1, pp.321-332, (1988)
22) H.Sidenbladh, M.J.Black, " Learning Image Statistics for Bayesian Tracking ", *ICCV*, Vol.2, pp.709-716, (2001)
23) H.Tao, H.S.Sawhney and R.Kumar, " Object Tracking with Bayesian Estimation of Dynamic Layer Representations ", *PAMI*, Vol.24, No.1, pp.75-89, (2002)
24) B.Heisele, U,Kreßel, W.Ritter, " Tracking Non-Rigid Moving Objects Based on Color Cluster Flow ", *CVPR*, pp.253-257 (1997)
25) A.Agarwal and B.Triggs, " Tracking Articulated Motion using a Mixture of Autoregressive Models ", *ECCV*, Vol.3, pp.54-65, (2004)
26) J.Hartigan, M.Wong, " Algorithm AS136: A K-Means Clustering Algorithm ", *Applied Statistics*, Vol.28, pp.100-108, (1979)