PAPER

Object Tracking with Target and Background Samples

Chunsheng HUA[†], ***Nonmember***, **Haiyuan WU**[†a)], ***Member***, **Qian CHEN**[†], ***Nonmember***, ***and*** **Toshikazu WADA**[†], ***Member***

SUMMARY In this paper, we present a general object tracking method based on a newly proposed pixel-wise clustering algorithm. To track an object in a cluttered environment is a challenging issue because a target object may be in concave shape or have apertures (e.g. a hand or a comb). In those cases, it is difficult to separate the target from the background completely by simply modifying the shape of the search area. Our algorithm solves the problem by 1) describing the target object by a set of pixels; 2) using a K-means based algorithm to detect all target pixels. To realize stable and reliable detection of target pixels, we firstly use a 5D feature vector to describe both the color ("Y, U, V") and the position ("x, y") of each pixel uniformly. This enables the simultaneous adaptation to both the color and geometric features during tracking. Secondly, we use a variable ellipse model to describe the shape of the search area and to model the surrounding background. This guarantees the stable object tracking under various geometric transformations. The robust tracking is realized by classifying the pixels within the search area into "target" and "background" groups with a K-means clustering based algorithm that uses the "positive" and "negative" samples. We also propose a method that can detect the tracking failure and recover from it during tracking by making use of both the "positive" and "negative" samples. This feature makes our method become a more reliable tracking algorithm because it can discover the target once again when the target has become lost. Through the extensive experiments under various environments and conditions, the effectiveness and efficiency of the proposed algorithm is confirmed.

key words: *object tracking, K-means clustering, tracking failure detection and recovery, background interfusion*

1. Introduction

This paper describes a new approach to track an object based on the K-means clustering algorithm, which can achieve robust object tracking in many harsh conditions, such as cluttered background, target objects have apertures and non-rigid deformation, partial occlusion and illumination variance, etc.

Numerous powerful algorithms for object tracking have been reported since the last two decades. They can be categorized according to: 1) object representation; 2) search strategy; and 3) similarity (or dissimilarity) measurement. According to object representation, a target object can be described by: 1-a) object appearance [4], [5]; 1-b) color histogram [6], [7]; 1-c) image feature [3] and 1-d) target contour [13], [22]. The appearance-based representation and color histogram representation have to define a region to describe the target object. When some background pixels are mixed in to that defined region, tracking often fails. The contour-based representation, as well as feature-based representation, is not suitable for blurred or noisy image sequences and the initialization is not straight forward.

The search strategy can be classified into three types as: 2-a) brute force search within a search area [3]–[5]; 2-b) steepest ascent (descent) search [6], [7] and 2-c) random sampling search [9], [10]. Brute force search may be time consuming and will fail when a similar object exists in the search area. Steepest ascent (descent) search achieves good tracking performance when the smoothly distributed similarity (dissimilarity) distribution is available. However, if the distribution is over-smoothed, the local maxima (minima) may be removed that will cause the tracking failure when similar objects are located in close proximity. Random sampling search has the advantage in being able to recover from small tracking failure because it maintains the model parameter distribution.

The similarity measurement spreads in a wide range according to the object representation. In the case of appearance model (e.g. template), dissimilarity measurement SAD (Sum of Absolute Difference) and SSD (Sum of Squared Difference) are widely used. The Bhattacharyya distance [11] and Kullback-Lieber divergence [12], as well as Jeffly divergence, can be used as the dissimilarity measurement between color histograms. In the case of image feature and contour-based representations, Chamfer distance is also a good choice, however there are still many other measures.

Several tracking algorithms based on the background subtraction [2] are also available. Those works assume that the background is either still or is rigid and its 3D structure and motion are known before object tracking. Therefore, they can not be used when the background is changing or there are multiple moving objects.

Many objects contain apertures or have complex conceaved shape. For such objects, some background area will be visible through the apertures. If a target object is described by a convex solid shape, when the object moves in front of a complex background, background area through the apertures will be mixed into the object area, which will do harm to object tracking. (hereafter we call this phenomenon as the *background interfusion*). Most of the conventional tracking algorithms suffer from the background interfusion problem.

Manuscript received April 14, 2006.

Manuscript revised August 11, 2006.

[†]The authors are with the Faculty of System Engineering, Wakayama University, Wakayama-shi, 640–8510 Japan.

a) E-mail: wuhy@sys.wakayama-u.ac.jp

DOI: 10.1093/ietisy/e90–d.4.766

To solve this problem, we brought out a pixel-wise multi-color object tracking algorithm. It discriminates the mixed background pixels from the target object by applying K-means clustering algorithm to the pixels in the search area, and both the *positive* (target) and *negative* (background) samples are used in the clustering. With the help of both the positive and negative information, our algorithm achieves the self tracking failure detection and realizes the function of automatic recovery from the tracking failure. Furthermore, the recovered color is also confirmed by a Bayesian formulate for increasing the reliability of the tracking failure recovery.

The Mean-shift [6], [7] tracking algorithm is famous for its good performance. In that method, the tracking target is described by an ellipse and the object feature is described by a center weighted color histogram of the pixels in the ellipse. The object tracking is performed by searching for the ellipse area in the image that has the most similar color histogram to the one of the tracking target with a hill-climbing in the distribution of the color histogram similarity. However, several key issues are remained in it: 1) it is unsuitable for tracking monochromatic and planar objects. The color histogram of such objects will become a narrow peak that breaks the smooth color histogram assumption the Mean-shift method made. For this kind of objects, the mean-shift method will fail easily even when the illumination only changes a little; 2) it suffers from the background interfusion problem. Although the center weighted color histogram can reduce the influence of the background pixels surrounding the target, it can not discriminate the target from the mixed background pixels completely. Moreover, for objects that have apertures, the problem will become even serious than many traditional tracking method such as template matching because of the center weighted color histogram. Our algorithm *can* discriminate the target from the mixed background pixels completely. This is the most obvious advantage over the Mean-shift method.

2. Key Ideas

2.1 Key Ideas in This Research

As a general purpose tracking algorithm, it should be able to track an object when: 1) the shape of the object is complex and deformable; 2) the color(s) of the object varies during tracking caused by illumination variance. Also, we consider that the reliability of object tracking will be improved much if the tracking algorithm has the ability of detecting the tracking failure caused by some random events and recovering from it. In order to realize such a tracking algorithm, we:

(1). *Use the K-means clustering algorithm to perform a pixel-wise object tracking.* Compared with appearance model-based methods, the pixel-wise method: 1) Does not assume a prior object shape; 2) Is suitable for tracking non-rigid or wired objects.

(2). *Introduce the concept of negative (background) samples into the pixel-wise method.* Conventional pixel-wise algorithms only use positive (target object) samples to perform the pixel classification. It estimates the dissimilarity of a pixel to the positive samples and uses a threshold to determine if a pixel belongs to the target or not. However, during object tracking, both the color of the target object and its surrounding background will change. Thus a fixed threshold will not give good results during tracking. To solve this problem, we introduce the concept of negative samples that describe the background near the target object into the pixel-wise classification. The classification is performed by firstly evaluating the dissimilarity of a pixel to the positive samples (target) (d_T) and that to the negative samples (background) (d_B). The pixel will be classified as target if $d_T < d_B$. In this approach, both positive samples and negative samples are properly updated continuously during tracking. Since all target pixels can be detected during tracking, positive samples describing them best will also be available. For the same reason, the background pixels surrounding the target object are also available. Therefore, the negative samples that correctly describe them can also be obtained. The detail about how to obtain proper background samples for object tracking has been discussed in the Sects. 3.1, 4.1 and 3.3. Because both positive samples and negative samples can be properly updated continuously during tracking, our approach ensures that the pixel classification will be performed adaptively when the target and the background varies. Meanwhile, since our approach does not use any thresholds for pixel classification, problems such as unstableness caused by improper threshold values will not exist in our approach.

(3). *Use a 5D uniform feature space to describe image features.* Since both the position and the color of the target object as well as the background change during tracking, it is natural to process them simultaneously. In the proposed 5D feature space, the color and the position of a pixel are described uniformly by a vector $\mathbf{f} = [\mathbf{c}\ \mathbf{p}]^T$. Here $\mathbf{c} = [Y\ U\ V]^T$ describes the color and $\mathbf{p} = [x\ y]^T$ describes the position of a pixel (see Fig. 1). By performing pixel classification in this 5D feature space, both the movement and the color variance of the target object can be followed automatically during tracking.

(4). *Propose an automatic self tracking failure detection and recovery method.* The tracking failure detection is achieved by examining if the dramatic change of the color(s) of the target object has occurred between two adjacent im-

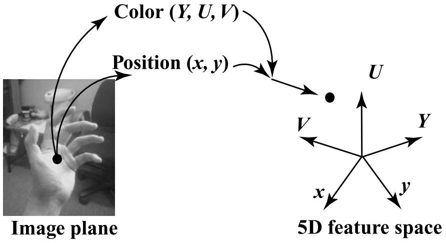


Fig. 1 Explanation for 5D feature vector.

age frames. The failure recovery is realized by finding out pixel(s) that is (are) most similar to the positive samples (target) determined in the previous image frame. The detail of this procedure is described in Sects. 3.4 and 3.5.

2.2 New Contributions in This Paper

There are also some researches reported that mentioned the idea of using both positive and negative information for robust tracking [1], [23]. Wada. etc [23] brought out an idea that the background interfusion problem could be solved by using negative information as well as positive information. They used the K-means clustering algorithm to classify the pixels into target cluster or background cluster. However, their idea was not formulized thus why it should work are not described clearly. Also, they did not give a method for obtaining negative information. Therefore, how to apply their idea to a real application was not clear. Hua. etc [1] formulized the idea in [23] and used a variable ellipse for describing the search area and gathering the negative information. But, 1) only monochromatic target object was considered, 2) the tracking failure detection in that work gives false alarm for objects having apertures thus causes unnecessary failure recovery to be carried out; 3) the recovered target color is not verified thus there was not a guarantee that the recovered color would be correct.

In this paper, we 1) formulate and extend the idea of [23]; 2) compared with Hua's work, our method can track objects that have multiple colors and shading; 3) bring out a new tracking failure detection method that can deal with objects that have multiple colors and apertures; 4) apply a Bayesian formulate to check if the recovered target color is correct or not.

3. Tracking Algorithm

3.1 Elliptical Search Area

The search area in this paper is defined as a region that contains the whole target object both in the previous and the current image frames. We use an ellipse to describe the search area. The shape and the position of the elliptical search area are determined according to the previous tracking result and the maximum velocity of the target object. The advantage of using an ellipse to describe the search area has been reported in [1]. Since all the pixels of the target object are in the elliptical search area, the pixels on the elliptical contour are background pixels. We use some pixels on the elliptical contour to represent negative samples (background) surrounding the target object during tracking.

3.2 Target Detection with K-Means Clustering

Target detection is carried out by classifying all the pixels in the elliptical search area as "target" or "background" with K-means clustering algorithm. In order to track an object having multiple colors, we prepare N target clusters. Each of them describes one of the N representative color part of the target object. N is the number of representative colors of the target object, which is given before starting the tracking job. In the case that the target object contains gradually changed colors, such as smooth shading effect of smooth curved surfaces, the distribution of the colors of the object are not completely random. Therefore, the colors can be divided into N groups according to their distribution. For each color group, we use the mean color as the representative color of it and use it to describe that color group. In this case, color grouping and mean color determination are carried out automatically during pixel classification with K-means clustering algorithm. We also prepare m background clusters which are defined by pixels on the elliptical contour.

The centers of both the i^{th} target cluster ($\mathbf{f}_T(i)$) and the j^{th} background cluster ($\mathbf{f}_B(j)$) are described by the 5D feature vectors as follows:

$$\begin{cases} \mathbf{f}_T(i) = [\mathbf{c}_T(i)\ \mathbf{p}_T(i)]^T, & i = 1\dots N \\ \mathbf{f}_B(j) = [\mathbf{c}_B(j)\ \mathbf{p}_B(j)]^T, & j = 1\dots m \end{cases}, \tag{1}$$

A pixel to be classified (unknown pixel) is described by $\mathbf{f}_u = [\mathbf{c}_u\ \mathbf{p}_u]^T$. In order to determine whether it belongs to *target* or to *background*, the distance from it to the nearest target cluster center (d_T) and that to the nearest background cluster center (d_B) are calculated as follows.

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

$$d_B = \min_{j=1\sim m} \left\{ \|\mathbf{f}_B(j) - \mathbf{f}_u\|^2 \right\}. \tag{3}$$

As shown in Fig. 2, if $d_T < d_B$, the pixel is classified as *target*, otherwise *background*. When the pixel has been classified as *target*, the cluster number (i) that the pixel belongs is recorded for the update procedure that will be described in the next section.

3.3 Update of Search Area

Since the target object moves and/or changes its shape, both the position and the shape of the search area need to be updated for tracking the target object in the next image frame. Here, the ellipse that describes the search area is determined

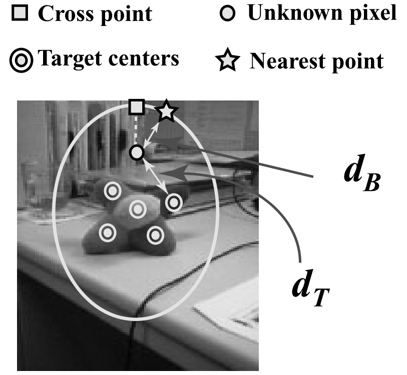


Fig. 2 Explanation for target clustering with multiple colors.

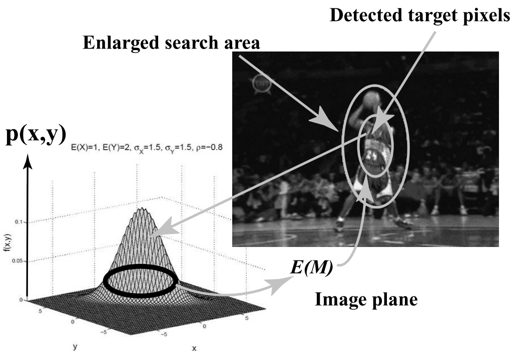


Fig. 3 Search area updated by the Mahalanobis distance.

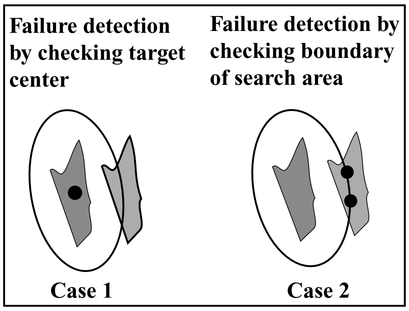

Failure detection by checking target center
Failure detection by checking boundary of search area
Case 1
Case 2

Fig. 4 Illustration for failure detection.

by using the pixel set of target object obtained from the current image frame. The pixels of the target object can be considered as a random distributed point set, thus they can be described by a probability distribution. Here we assume that the distribution can be described by a Gaussian probability density function (pdf) approximately [1]. Let $\mathbf{S_Z} = [\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_n]^T$ be the pixel set of the target object, where $\mathbf{Z}_i = [x_i, y_i]^T$ is the position of i-th pixel of the target object and n is the number of the target pixels. The Gaussian *pdf* describing the distribution of $\mathbf{S_Z}$ is given by:

$$\mathbf{S_Z} \sim \mathcal{N}(\mathbf{m}_Z, \Sigma_Z), \tag{4}$$

where $\mathbf{m}_Z$ is the mean and Σ_Z is the covariance matrix. The Mahalanobis distance from a vector $\mathbf{Z}$ to the mean $\mathbf{m}_Z$ is given as:

$$g(\mathbf{Z}) = [\mathbf{Z} - \mathbf{m}_Z]^T {\Sigma_Z}^{-1} [\mathbf{Z} - \mathbf{m}_Z]. \tag{5}$$

The ellipse $E(M)$ that will contain $M\%$ pixels of the whole target object is given by

$$g(\mathbf{Z}) = J, \tag{6}$$

where $J = -2\ln(1 - \frac{M}{100})$. (as shown in Fig. 3) We let M be big enough (e.g. 95) so that $E(M)$ will include the overwhelming majority of target pixels, thus $E(M)$ represents the approximated shape of the target object. Since target object moves, we enlarge $E(M)$ by k (e.g. 1.25) times and use it as the search area for the tracking in the next frame, where k is determined by considering the speed of the target object.

3.4 Self Tracking Failure Detection

In most tracking algorithm, target tracking is performed by searching a small area around the target detected in the previous frame for the target object. This approach works only when the result obtained in the previous frame is correct. Therefore, we consider that, after the object tracking has been carried out in the current frame, it is necessary to perform a verification procedure that checks whether the tracking result correct or not. If a tracking failure occurs and is detected, the current object tracking job should be interrupted and a failure recovery job should be started. Once the failure recovery has been performed successfully, we can restart the interrupted tracking job.

3.4.1 Failure Detection by Checking Target Center

If the target object is solid and it moves smoothly, one cluster center of a particular color part of the target object determined in the previous frame should still be on the target object of same color in the current frame. Therefore, after the detection of the target object in the current frame is completed, if the pixel in the current frame at the position of the cluster center determined in the previous frame is not having the similar color (as the case 1 of Fig. 4), we can consider that a tracking failure may have occurred in the current frame. This situation can be confirmed by the following equation:

$$f_C(T_i) = \begin{cases} 1; & \text{if } d_{Ti} > d_{PiB} \\ 0; & \text{otherwise} \end{cases}, \tag{7}$$

where $d_{Ti} = ||\mathbf{C}_{Ti}^{(t)} - \mathbf{C}_{Pi}^{(t+1)}||^2$, $d_{PiB} = \min_{j=1\sim m} ||\mathbf{C}_{Pi}^{(t+1)} - \mathbf{C}_{B}^{(t+1)}(j)||^2$. $\mathbf{C}_{Ti}^{(t)}$ is the color of the i-th cluster center in the previous frame. $\mathbf{C}_{Pi}^{(t+1)}$ is the color of pixel in the current frame at the position of the i-th cluster center in the previous frame. $\mathbf{C}_{B}^{(t+1)}(j)$ is the color of j-th background sample in the current frame. If $f_C(T_i) = 1$, it means that the color of pixel in the current frame at the position of the i-th cluster center in the previous frame looks more like background rather than the color of the previous cluster center thus indicates a tracking failure may have occurred.

However, in the cases that a target object has thin color patterns, is in a concave shape or has apertures, false alarm will be raised often. To cope with this situation, we propose one more method for checking tracking failure which will be described in the next sub-section.

3.4.2 Failure Detection by Checking the Boundary of Search Area

As described in Sect. 3.1, the ellipse contains the whole target object in the current frame and should still contain the whole target object in the next frame. Therefore, if the object goes across the boundary of the search area (elliptical contour), we can consider that a tracking failure may have occurred in the current frame. This may happen when the speed of the target object is faster than the predicted maximum speed. This situation can be confirmed by checking whether the colors of the pixels in the current frame on the elliptical contour determined in the previous frame are similar to the color of the target cluster centers (as case 2 of Fig. 4).

The condition that the color of a pixel in the current frame on the elliptical contour determined in the previous frame is similar to the color of one of the target cluster centers is given by

$$f_B(j) = \begin{cases} 1; & \text{if } \; d_{BTj} < d_{BBj} \\ 0; & \text{otherwise} \end{cases} \quad , \tag{8}$$

where $d_{BTj} = \min_{i=1\sim N} \|\mathbf{C}^t_{Ti} - \mathbf{C}^{(t+1)}_B(j)\|^2$, $d_{BBj} = \|\mathbf{C}^t_B(j) - \mathbf{C}^{(t+1)}_B(j)\|^2$. $\mathbf{C}^t_{Ti}$ is the color of i-th target cluster center determined in the previous frame, $\mathbf{C}^t_B(j)$ and $\mathbf{C}^{(t+1)}_B(j)$ are the color of j-th pixel on the elliptical contour in the previous and the current frame, respectively. The condition $f_B(j) = 1$ indicates that, in the current frame, the color of j^{th} background sample is more similar to the i^{th} target color than it to the j^{th} background sample in the previous frame. And this situation can be considered that the ellipse contour is overlapped with the target object thus indicates that parts of the target object moves out of the search area.

By checking the target center and the background samples, the probability of tracking failure can be estimated with the following equation:

$$P(fail) = 0.5\frac{\sum_{i=1}^{N} f_C(T_i)}{N} + 0.5\frac{\sum_{j=1}^{m} f_B(j)}{m}. \tag{9}$$

When $P(fail)$ is greater than 0.7 (an experimental value), we will judge that a tracking failure has occurred.

3.5 Tracking Failure Recovery

When a tracking failure occurred and was detected, some or all of the newly determined target cluster centers will be false ones. This means that the tracking of those target clusters are failed. Those false cluster centers can be detected by checking the condition indicated by Eq. (7).

The recovery from the tracking failure is achieved by re-determining each false target cluster center by searching for the pixel within the elliptical search area (*ellipse*) in the current frame that is most similar to that cluster center determined in the previous frame.

Let $\mathbf{f}^{(t)}_T$ be the 5-d feature vector of one of the lost cluster center determined in the previous frame t, $\mathbf{f}^{(t+1)}$ be one of the pixels in the search area in the current frame $t+1$, and $\mathbf{f}^{(t+1)}_T$ be the re-determined target cluster center. $\mathbf{f}^{(t+1)}_T$ can be determined by finding out the $\mathbf{f}^{(t+1)} \in$ *ellipse* that minimizing $d(\mathbf{f}^{(t+1)})$, where

$$d(\mathbf{f}^{(t+1)}) = \|\mathbf{f}^{(t+1)} - \mathbf{f}^{(t)}_T\|^2. \tag{10}$$

Since the color of a target cluster is being updated continuously during tracking, it may be quite different from the initial color of that target cluster. Because of this, it is necessary to check if a target cluster center has been recovered correctly by checking whether the color of the recovered target cluster center is reasonable. We use a Bayesian formulate to verify the reliability of the color of a recovered target cluster center.

$$P\Big(\mathbf{C}(k)|\mathbf{C}_{rec}\Big) = \frac{p\Big(\mathbf{C}_{rec}|\mathbf{C}(k)\Big)P\Big(\mathbf{C}(k)\Big)}{\Sigma_{i=1}^{N}\{p\Big(\mathbf{C}_{rec}|\mathbf{C}(i)\Big)P\Big(\mathbf{C}(i)\Big)\}}. \tag{11}$$

In Eq. (11), $\mathbf{C}_{rec}$ is the color of the recovered target cluster center, $\mathbf{C}(k)$ denotes the initial color of k^{th} target cluster center, $p\Big(\mathbf{C}_{rec}|\mathbf{C}(k)\Big)$ is the likelihood between $\mathbf{C}_{rec}$ and $\mathbf{C}(k)$, $P\Big(\mathbf{C}(k)\Big)$ is the prior probability of k^{th} target cluster decided by the ratio between the area of the k^{th} target cluster and the whole target area in the previous frame. By finding out k that maximizes the posterior $P\Big(\mathbf{C}(k)|\mathbf{C}_{rec}\Big)$, we determine target cluster that the color of its center has the maximum similarity to the color of the recovered target cluster center. If the k^{th} target cluster is the target cluster of which the center was tried to recovered, we consider the tracking failure recovery has been successful.

4. Experiment and Discussion

4.1 Manual Initialization

It is necessary to assign the number of clusters when using the K-means algorithm to classify a data set. In our tracking algorithm, in the first frame, we manually select N points on the object to be tracked and use them as the N initial target cluster centers. We let the initial ellipse (search area) be a circle. The center of circle is put at the centroid of the N initial target cluster centers. We manually select one point out of the object and let the circle cross it. In the following frames, the ellipse center is updated according to the result of target detection.

Here, we use m representative background samples selected from the ellipse contour. Theoretically, it is best to use all pixels on the boundary of the search area (ellipse contour) as representative background samples. However, since there are many pixels on the ellipse contour (several hundreds in common cases), the number of the clusters will become a huge one. This dramatically reduces the processing speed of pixel classification during tracking. In order to realize real-time processing speed, we let m be a small number. Through extensive experiment of tracking objects of wide classes, we found that $m = 9$ is a good choice for fast tracking while keeping the stable tracking performance. Of the 9 points, 8 points are resolved by the 8-equal division of the ellipse contour, and the 9th one is the cross point between the ellipse contour and the line connecting the pixel to be classified and the center of the ellipse center.

4.2 Efficiency of Our New Algorithm

To show the effectiveness of our new algorithm over the one

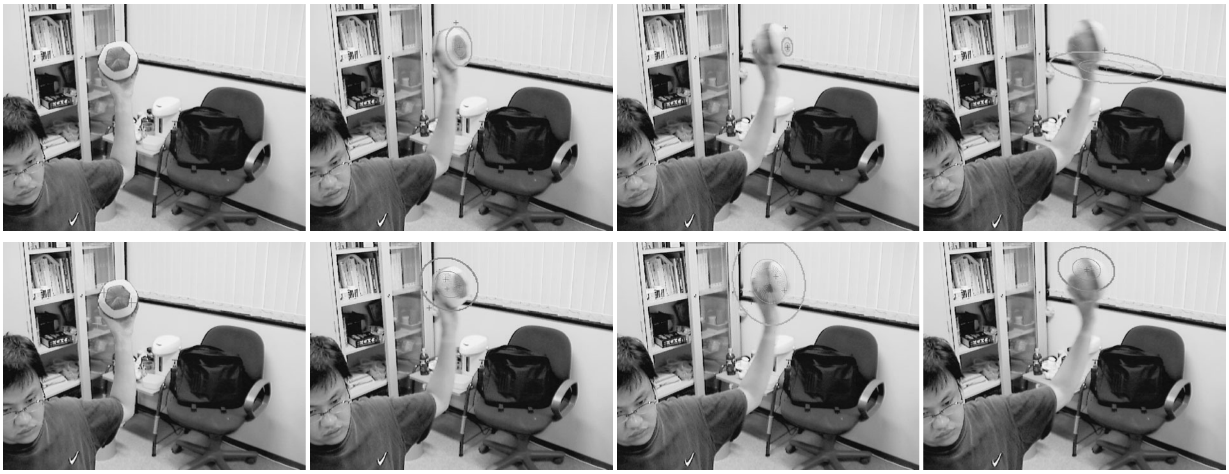


Fig. 5 Comparative experiment between our tracking algorithm and the referenced tracking algorithm in [1]. The upper row shows the tracking result of [1]. The lower row is the result of our new tracking algorithm.

described in [1], an experiment of tracking a multi-color object was performed. The experimental results are shown in Fig. 5. The upper row is the result obtained with algorithm described in [1]. Since only monochromatic target is considered in [1], it can only track one of the many color parts of the target object. When that color part become invisible caused by the rotation of target, the tracking was failed since Frame 010. The lower row shows the results obtained with our new algorithm. Since it can deal with the target object having multiple colors, it could track the target even when some parts of it became invisible. In Frame 010 in lower row, the ellipse contour turned green indicating that the disappearance of some target colors has been detected during tracking.

4.3 Comparative Experiments

To confirm the effectiveness of the proposed K-means tracker, we do some comparative experiments. They are taken among:

(a) SAD (Sum of Absolute Difference) template matching.

(b) Conventional Mean-shift Algorithm.

(c) Our K-means tracker.

They were applied to many objects under various sequences. Some targets for the comparative experiments are shown in Fig. 6, each target is response to each column in Table 1. In same table[†], "○" denotes "**Track Well**", "△" means "**Track Unstably**" and "×" represents "**Completely Fail**".

The target as shown in Fig. 6 (a) was used in the translation experiment, and all the three algorithms worked well.

In the experiment for the object rotation in plane, the target is the Fig. 6 (b). Since the template shape of SAD template matching is fixed, background pixels were gotten mixed with the template while the rotation. So the similarity measured between the template and the target appearance becomes lower and lower, which makes the object tracking unstable. The mean-shift algorithm can work well because it found out the most similar peak in the color histogram with hill-climbing computation, it is insensitive to the target rotation. Our K-means tracker works robustly because it detects the target object pixel by pixel and updates the search area according to the target detection result.

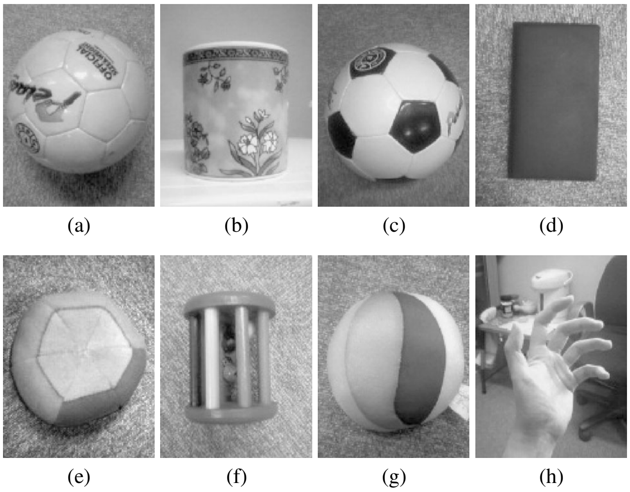


Fig. 6 Objects for the comparative experiments. Each is prepared for (a) Translation; (b) Rotation in tilt; (c) Rotation in pan and Scale; (d) Color shift; (e) Occlusion; (f) Wired object; (g) Textured light; (h) Non-rigid object.

The target as shown in Fig. 6 (c) was used in the experiment for the object rotation in depth. The reason why we select the Fig. 6 (c) is because that the appearance pattern of the target is undergo the change of rotation in depth. For the reasons mentioned in (1), the SAD template matching works

[†] The movies for these comparative experiments are visible at http://www.wakayama-u.ac.jp/˜wuhy/wu2_new.html

Table 1 Comparative experiments results. (Videos of these experiments can be downloaded from http://www.wakayama-u.ac.jp/˜wuhy/wu2_new.html)

	Translation	Rotation in plane	Rotation in depth	Scale	Color Shift	Partial Occlusion	Wired Object	Chess-board lighting	Non-rigid Object
SAD Template Matching	○	△	△	△	○	×	×	×	×
Mean-shift Algorithm	○	○	○	△	×	×	×	×	×
K-means Tracker	○	○	○	○	○	○	○	○	○

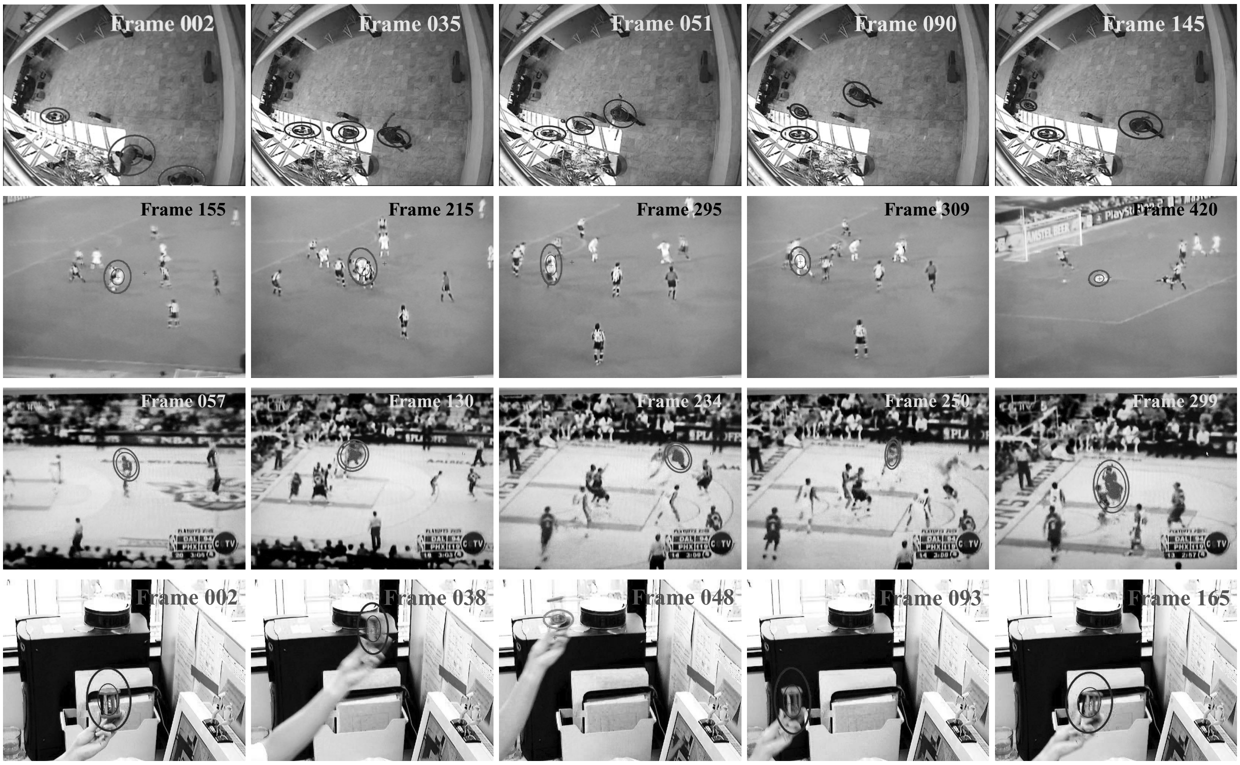


Fig. 7 Experiment results of K-means tracker with different targets under un-constraint conditions. Row 4 frame 048 shows the tracking failure detection and recovery.

unstably. Also as mentioned in (1), both the mean-shift algorithm and our K-means tracker can work well.

As for the scaling experiment (target is Fig. 6 (c) too). Because their templates are fixed, while scaling, the target appearance within the template will change greatly. Therefore, the similarity measured by the SAD template matching and mean-shift algorithm becomes low, which causes these two algorithms to work unstably. With the variable ellipse to restrict the search area, our K-means tracker could deal with the scaling well.

While the color-shift experiment, the target is Fig. 6 (d), which is monochromatic and contains highlight caused by its smooth surface. Because its color histogram is thin, when the illumination variance happens, mean-shift could hardly find an overlap part in the histograms between two frames, which lead to the tracking failure.

We used a target (Fig. 6 (e)) to perform the partial occlusion experiment. In this case, because the target appearance is greatly changed by a barrier, neither the SAD template matching nor the mean-shift could work. Because our K-means tracking algorithm detects each pixel as the target or background, it can track the visible part of target successfully.

In the wired object (Fig. 6 (f)) experiment, such target is difficult for many of the conventional tracking algorithms because the interfused background through object's apertures greatly changes the target features. The Mean-shift algorithm failed to track the target because the color histogram was changed. The SAD template matching failed because of the accumulated error caused by the background interfusion while template updating. Some images of our algorithm are available in Row 4 of Fig. 7.

We used a target (Fig. 6 (g)) for the chess-board lighting experiment. Here, the illumination variance includes two parts: (1) the un-continuous variance caused by the chess-board pattern projected in all directions; (2) gradual variance in the horizontal direction. The SAD template matching failed because of the part (1). The mean-shift algorithm failed because of both the part (1) and (2). Our algorithm was performed in the 5D feature space, so, it could follow the illumination variance well.

In the non-rigid object (Fig. 6 (h)) experiment. The following factors make the object tracking difficult: 1) ununiform illumination caused by the strong light from ceil and window aside. 2) background interfusion through the fingers. Both the SAD template matching and mean-shift algorithm failed to track the target, because of the illumination variance, background interfusion and target deformation.

4.4 Experiments under Real Sequence

As shown in Fig. 7, we also applied our K-means tracker to some other targets.

Row 1 is a surveillance sequence[†], in this case the challenging tasks are: 1) low image color saturation and the color differences between the target and background is small; 2) illumination and shape variance are distinguished.

Row 2 shows a soccer sequence[††], where non-rigid deformation, scaling and occlusion make it challenging for most of the conventional tracking algorithms.

The difficulties of Row 3 which is a basketball sequence are: complex colors of the surrounding audience, partial occlusion and the interfused background pixels, and these elements make it difficult for the prior-knowledge-based tracking algorithms to work robustly.

Row 4 shows the experiment with wired multi-color cage under complex background and illumination variance. The difficulties lie in: blurred image caused by high-speed movement, un-uniform illumination and background components. In frame 048, the ellipse contour is shown with different color from the other frames, which indicated that the tracking failure happened. With the recovery method mentioned Sect. 3.5, we can recover from this tracking failure as shown in following frames.

By classifying the target with target and background samples in the 5D feature space and using the variable ellipse to restrict the search area, our K-means tracker achieves the robust object tracking under such sequences.

All the experiments were taken with a desktop PC with 3.06GH Intel XEON CPU, and the image size was 640×480 pixels. When the target size varied from $140 \times 140 \sim 200 \times 200$ pixels, the processing speed of our algorithm was about 12~18 ms/frame.

5. Conclusion

In this paper, we have proposed a K-means tracker which is a general object tracking algorithm that can work in video-rate. Based on the pixel-wise clustering algorithm, we achieved robust object tracking for multi-color object with little prior information or assumption. With the update in the 5D feature space which describes the color and geometric feature, our algorithm becomes robust against the illumination variance. By applying K-means clustering algorithm to both the target and background samples, we can successfully discriminate the target from the surrounding or interfused background pixels. Also with target and background samples, our method achieves the automatic tracking failure detection and recovery, which helps us greatly in robust object tracking.

At present, since the initialization is performed manually, for feature work, we plan to perform automatic target detection as the initialization step, then combine this initialization with our K-means tracker to perform a complete automatic tracking system.

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) 18500131, (C)(2) 15500112.

References

[1] C. Hua, H. Wu, T. Wada, and Q. Chen, "K-means tracking with variable ellipse model," IPSJ Trans. CVIM, vol.46, no.Sig 15 (CVIM12), pp.59–68, 2005.

[2] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture model for real-time tracking," CVPR, pp.246–252, 1999.

[3] H.T. Nguyen and A. Semeulders, "Tracking aspects of the foreground against the background," ECCV, vol.2, pp.446–456, 2004.

[4] H.D. Crane and C.M. Steele, "Translation-tolerant mask matching using noncoherent reflective optics," Pattern Recognit., vol.1, no.2, pp.129–136, 1968.

[5] C. Gräßl, T. Zinßer, and H. Niemann, "Illumination insensitive template matching with hyperplanes," DAGM, pp.273–280, 2003.

[6] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," CVPR, vol.2, pp.142–149, 2000.

[7] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," IEEE Trans. Pattern Anal. Mach. Intell., vol.25, no.5, pp.564–577, 2003.

[8] M. Isard and A. Blake, "Condensation-conditional density propagation for visual tracking," IJCV, vol.29, no.1, pp.5–28, 1998.

[9] J. Deutscher, B. North, B. Bascle, and A. Blake, "Tracking through singularities and discontinuities by random sampling," ICCV, pp.1144–1149, 1999.

[10] K. Smith and D. Gatica-Perez, "Order matters: A distributed sampling method for multi-object tracking," BMVC, pp.25–32, London, Sept. 2004.

[11] F.C. Schweppe, "State space evaluation of the Bhattacharyya distance between two Gaussian processes," Information and Control, vol.11, no.3, pp.352–372, Sept. 1967.

[12] H.Z. Rafi and H. Soltanianzadeh, "Mutual information restoration of multispectral images," IWSSIP, 2003.

[13] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," IJCV, vol.1, pp.321–332, 1988.

[14] T. Zhao and R. Nevatia, "Tracking multiple humans in crowded environment," CVPR, vol.2, pp.406–413, 2004.

[15] Y. Wu, G. Hua, and T. Yu, "Switching observation models for contour tracking in clutter," CVPR, vol.1, pp.295–302, 2003.

[16] N.T. Siebel and S. Maybank, "Fusion of multiple tracking algorithms for robust people tracking," ECCV, vol.4, pp.373–387, 2002.

[17] 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.

[18] H. Tao, H.S. Sawhney, and R. Kumar, "Object tracking with Bayesian estimation of dynamic layer representations," IEEE Trans.

[†] the PETS2001 public database, University of Reading, UK, http://peipa.essex.ac.uk/ipa/pets/PETS2001

[††] This movie is taken from the European Champion League 2002–2003.

Pattern Anal. Mach. Intell., vol.24, no.1, pp.75–89, 2002.

[19] B. Heisele, U, Kreßel, and W. Ritter, "Tracking non-rigid moving objects based on color cluster flow," CVPR, pp.253–257, 1997.

[20] A. Agarwal and B. Triggs, "Tracking articulated motion using a mixture of autoregressive models," ECCV, vol.3, pp.54–65, 2004.

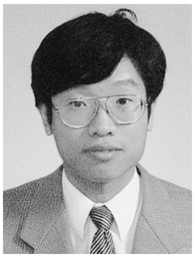
[21] J. Hartigan and M. Wong, "Algorithm AS136: A K-means clustering algorithm," Applied Statistics, vol.28, pp.100–108, 1979.

[22] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," ECCV, vol.1, pp.343–356, 1996.

[23] T. Wada, T. Hamatsuka, and T. Kato, "K-means tracking: A robust target tracking against background involution," MIRU2004, vol.2, pp.7–12, 2004.

Chunsheng Hua received his BE degree in electronic engineering from Shenyang University of Technology in 2001. He received his MS degree from the Department of Mechanical and System Engineering at Kyoto Institute of Technology in 2004. He started his Ph.D. course since 2004 until now. He is a student member of IEEE and IPSJ. His research interests include face recognition, color-based object tracking. He received the Award of IPSJ Digital Courier Funai Young Researchers in 2006.

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



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

Toshikazu Wada received his BEng degree in electronic engineering from Okayama University, MEng degree in computer science from Tokyo Institute of Technology and DEng degree in applied electronics from Tokyo Institute of Technology, in 1984, 1987, 1990, respectively. He is currently a professor in Department of Computer and Communication Science, 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 Engineerings (IEICE), Japan. He is a member of IPSJ, the Japanese Society for Artificial Intelligence and IEEE.