*Regular Paper*

# K-means Clustering Based Pixel-wise Object Tracking

CHUNSHENG HUA,†,☆ HAIYUAN WU ,† QIAN CHEN † and TOSHIKAZU WADA †

This paper brings out a robust pixel-wise object tracking algorithm which is based on the K-means clustering algorithm. In order to achieve the robust object tracking under complex condition (such as wired objects, cluttered background), a new reliability-based K-means clustering algorithm is applied to remove the noise background pixel (which is neigher similar to the target nor the background samples) from the target object. According to the triangular relationship among an unknown pixle and its two nearest cluster centers (target and background), the normal pixel (target or background one) will be assigned with high reliability value and correctly classified, while noise pixels will be given low reliability value and ignored. A radial sampling method is also brought out for improving both the processing speed and the robustness of this algorithm. According to the proposed algorithm, we have set up a real video-rate object tracking system. Through the extensive experiments, the effectiveness and advantages of this reliability-based K-means tracking algorithm are confirmed.

## 1. Introduction

Years of vision research have yielded in lots of powerful object tracking algorithms. Numerous powerful and effective algorithms have been applied into object tracking, such as the Kalman filter [7], template matching [8], EM algorithm [11], CONDENSATION [9] (also called as the "particle filter") algorithm, dynamic Bayesian network [10], mean shift [12] and iterative clustering [6], etc.

The template matching algorithm [18] uses the sum of grey value of the predefined template to describe the target feature. The object tracking is achieved by searching over the image to find out the region that has the maximum similarity to the predefined template. Because the grey value of template is used to describe the target feature, the original template matching is very sensitive to the illumination changes and target deformation. Improvements on template matching [8] have been reported.

Background subtraction is able to track the target object by checking the differences between the observed image and the predefined background model. Therefore, it is especially suitable to work indoors, such as the subway station, airport, etc. Although adaptive background subtraction [19] has been proposed, background subtraction still suffers from the random noise (such as rains or snows) and is not suitable to work in the dynamic scene. The same problem also exists in the optical flow method [20] which tracks the target object by examining the characteristics of flow vectors of the moving target over the surrounding background.

Comaniciu [12], *et al.* use the color distribution of the target to describe the target feature, and they achieve the robust object tracking for the non-rigid objects by using the mean-shift algorithm. One of the key issues in mean shift algorithm is to produce the center-weighted image (which is also a 3D color histogram) at time with the Bhattacharyya coefficient. In this center-weighted image, the pixels on the target object have high weight, while pixels on the background have low weight. By applying the hill-climbing calculation to find out the mode of 3D histogram, mean shift algorithm achieves the robust object tracking under cluttered condition. But means shift algorithm is not suitable for the monochromatic and planar objects, because such objects usually appear like a vertical line or a narrow peak in the color histogram; even the illumination just changes a little, such narrow peak will be drifted dramatically and hill-climbing method will fail because there is no overlap of the target template in two adjacent frames.

CONDENSATION [9] can track the target through occlusion and clutter by reasoning over a state-space of multiple hypotheses. Because it combines the random sampling techniques with the posterior probability of the target object together, it achieves very robust object tracking.

However, although these algorithms are quite powerful, they only concentrate on the similarity between the target model and an unknown region/pixel. In order to measure such similarity, a threshold is usually applied into these algorithms. Since the target object may move under the clut-

☆ Presently with the ISIR of Osaka University
Presently with E-mail: huachunsheng@gmail.com
† Faculty of System Engineering, Wakayama University.
E-mail: wuhy, chen@sys.wakayama-u.ac.jp, twada@ieee.org

tered condition, it is difficult to select the proper threshold to work stably under all conditions. Furthermore, there is no guarantee that if the object with the maximum similarity is really the target one or not.

Collins [16] *et al.* bring out an idea that: while object tracking, the most important thing is the ability to discriminate the target object from its surrounding background. They propose a method to switch the mean-shift tracking algorithm among the different linear combination of the *RGB* colors which can select the very features that distinguish the object most from the surrounding background. Therefore, the kernel of this paper is how to select the best combination of *RGB* colors for discrimination. The combination that achieves the maximum ratio between the color distribution of target and that of the background is considered as that best features. Improved performance compared with the standard mean-shift algorithm has been reported in that paper. Even so, the color histogram has very little identification power and this method appears to work only when the target is solid and its appearance does not change dramatically. Meanwhile, in the case of high dimensional features like textures, the large number of combination of colors (in fact, each image contains 49 such color combinations) will prevent it from achieving the real-time performance.

Similar ideas have been applied by Nguyen [17] and Zhang [21]. In [21], they describe the target and background features with their color histograms. An unknown pixel is classified into target or background by comparing the bin values of its color between the target histogram and the background one. Therefore, when some parts of the surrounding background contain similar color to that of target, this ratio will become very low and some target pixels will be regarded as the background ones. Furthermore, because they assume that both the target and background are solid, the long-term performance of this paper is not ensured. Meanwhile, this work can not be applied to track the object with apertures because the mixed background will pollute the target histogram. In [17], although the more powerful Gabor filter is used to discriminate the target from the background, the performance of that work in the long term is suspected because the target is assumed to be solid. When the target is non-rigid, there will be no guarantee that the update of target template is correct, and the multi-scale problem is also remained in [17].

Therefore, obviously almost all of the mentioned tracking algorithms share one common problem: *when the target object is non-rigid and/or contains apertures, background pixels will be mixed into the target object; when the target object moves under the cluttered background condition, the continuously mixed background pixels will greatly degrade the purity of the target feature (such as color or texture)*. In this paper, this phenomenon is called as ***background interfusion***.

Hua [13–15] *et al.* bring out the *K-means tracker* to solve the background interfusion by introducing the negative samples as well as positive samples into object tracking. Within an ellipse model that is used to restrict the search area and represent the background samples, each pixel is classified into target or background groups with the K-means clustering by using those target and background samples described in a color-position feature space. By updating the search area according to the distribution of detected target pixels, K-means tracker can successfully deal with the target deformation and scaling. To make the K-means tracker work more robustly, tracking failure detection and recovery processes are also brought out in [13–15].

However, some inherent weaknesses in K-means clustering algorithm degrade the performance of the K-means tracker (the detail are described in Section 2.2 and **Fig**.1). To solve these problems, a new reliability-based K-means clustering (called as RK-means) is brought out in this paper. In object tracking, each pixel will be assigned with a reliability value according to the triangular relationship among that pixel, its nearest target center and its nearest representative background center (which is selected from a variable ellipse model). This reliability value indicates how reliable the classification of a pixel will be. Noise pixels that should not be classified as a target pixel or a background one will be given a low reliability value, thus will be ignored. Embedding this reliability into K-means clustering algorithm, the discrimination between the target pixels and the noise pixels becomes possible and this makes the object tracking more robust than the K-means tracker.

The following sections mainly explain this new reliability-based K-means clustering and its application in object tracking, the details description about 5D image feature, the update of search area, tracking failure detection and recovery can be found in [13–15].

## 2. K-means Clustering Based Object Tracking

### 2.1 Definition of target object

In this research, because the target object is described by the color and position information (as described in [13–15]), the tracked object is defined by its existence (its position and color information). It means, despite of the shape or size variance, the object tracking is considered as successful if the color and position of tracked object is correct.

### 2.2 Reliability-based K-means Clustering

Although K-means clustering is one of the most popular pixel-wise clustering algorithms and can be directly applied into object tracking, two problems degrade its performance: 1) When the noise data not belonging to any cluster exist, K-means clustering will wrongly classify them into some predefined clusters; 2) the wrongly assumed number of clusters sometimes leads to the wrong clustering result. Although some researchers brought out some improvements [1–5] on K-means clustering, while object tracking such problems still affect the performance of K-means clustering.


(a) Input image (b) Result of K-means

**Fig. 1** In [13], K-means tracker will suffer from the noise data.

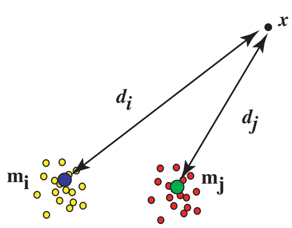



**Fig. 2** When the noise data exists, K-means clustering will give wrong conclusion that $x$ belongs to $\mathbf{m}_j$.

The problem caused by the noise data can be explained as: the K-means clustering only classifies the unknown data according to its distance to the cluster centers. As shown in **Fig**.2, K-means clustering algorithm only check $d_i$ and $d_j$ (which are the distance among an unknown data $\mathbf{x}$ and the cluster centers $\mathbf{m}_i, \mathbf{m}_j$) to judge which cluster $\mathbf{x}$ should belong to, even if $\mathbf{x}$ is more like to be a noise data.

In order to solve the mentioned problems, we introduce the reliability estimation into K-means clustering. The *reliability* value for each data is used to judge if it is reasonable to classify an unknown data into one cluster or not.

Because the noise data are always distant from the data belonging to the initialized clusters (called as normal data), the distance from noise data to its nearest cluster center should be longer than that of the normal data. Therefore, the distance between a data and its closest cluster center can be used to classify the unknown data. However, a standard is necessary for estimating how far a data is from its nearest cluster center. Here, we use the triangular relationship among an unknown data and its two nearest cluster centers to measure if it is reliable or not to classify this data into some clusters. A high reliability value should be assigned to the normal data, and a low value to the noise one.

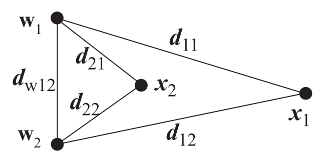



**Fig. 3** The relationship between data vectors and their two closest cluster centers.

As shown in **Fig**.3, $\mathbf{w}_1$ and $\mathbf{w}_2$ are the cluster centers, $\mathbf{x}_1$ and $\mathbf{x}_2$ are the unknown data vectors. According to $d_{11}$ and $d_{21}$, we can only judge $\mathbf{x}_2$ is closer to $\mathbf{w}_1$ than $\mathbf{x}_1$. But we can not determine which of $\mathbf{x}_1$ and $\mathbf{x}_2$ is the noise data vector. However, according to the shape of triangle $\triangle\mathbf{x}_1\mathbf{w}_1\mathbf{w}_2$ and $\triangle\mathbf{x}_2\mathbf{w}_1\mathbf{w}_2$, we can draw a conclusion that, compared with $\mathbf{x}_2$, $\mathbf{x}_1$ has higher probability to be a noise data vector.

While clustering an arbitrary data vector $\mathbf{x}_k$ of data set $\mathbf{X}$ ( $\{\mathbf{x}_k; k = 1, \ldots, n\}$), the reliability value of $\mathbf{x}_k$ is defined as the ratio of the distance between its two closest cluster centers to the sum of the distance from $\mathbf{x}_k$ to the two cluster centers:

$$\mathcal{R}(\mathbf{x}_k) = \frac{\| \mathbf{w}_{f(\mathbf{x}_k)} - \mathbf{w}_{s(\mathbf{x}_k)} \|}{d_{kf} + d_{ks}}, \tag{1}$$

where

$$\begin{aligned} d_{kf} &= \| \mathbf{x}_k - \mathbf{w}_{f(\mathbf{x}_k)} \|, \\ d_{ks} &= \| \mathbf{x}_k - \mathbf{w}_{s(\mathbf{x}_k)} \| . \end{aligned} \tag{2}$$

$f(\mathbf{x}_k)$ and $s(\mathbf{x}_k)$ are the subscript of the closest and the secondly closest cluster centers to $\mathbf{x}_k$:

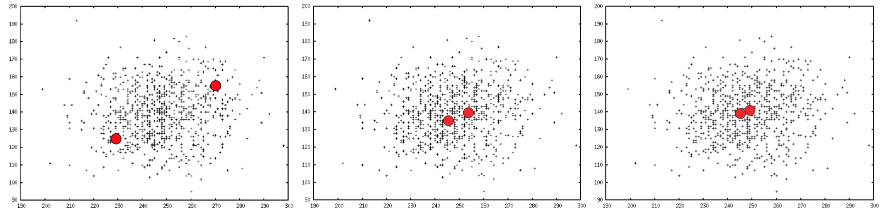
$$\begin{aligned} f(\mathbf{x}_k) &= \underset{i=1,\ldots,c}{\operatorname{argmin}} (\| \mathbf{x}_k - \mathbf{w}_i \|) \\ s(\mathbf{x}_k) &= \underset{i=1,\ldots,c, i \neq f(\mathbf{x}_k)}{\operatorname{argmin}} (\| \mathbf{x}_k - \mathbf{w}_i \|) . \end{aligned} \tag{3}$$

**Fig**.4 shows RK-means can detect the outliers.

The degree ($\mu_{kf}$) of a data vector $\mathbf{x}_k$ belonging to its closest cluster $f(\mathbf{x}_k)$ is computed from $d_{kf}$ and $d_{ks}$:

$$\mu_{kf} = \frac{d_{ks}}{d_{kf} + d_{ks}}. \tag{4}$$

Since $\mathcal{R}(\mathbf{x}_k)$ indicates how reliable that $\mathbf{x}_k$ can



Initial state Iteration 5 Iteration 9

**Fig. 6** In one crowd of data vectors, RK-means will merge two initial cluster centers together.

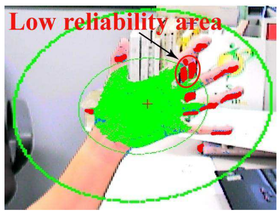



**Fig. 4** Outliers are detected and painted with red by our RK-means algorithm.

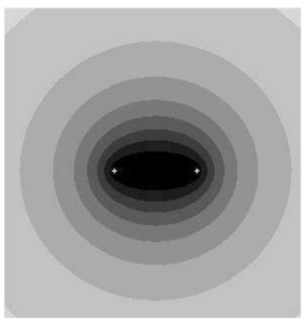


**Fig. 5** Reliability field around two cluster centers. The reliability values are shown by intensity.

be classified, the probability that $\mathbf{x}_k$ belongs to its closest cluster can be computed as the product of $\mathcal{R}(\mathbf{x}_k)$ and $\mu_{kf}$.

$$t_{kf} = \mathcal{R}(\mathbf{x}_k) * \mu_{kf} \tag{5}$$

$t_{kf}$ denotes that: under the reliability $\mathcal{R}(\mathbf{x}_k)$, the probability that $\mathbf{x}_k$ belongs to its closest cluster.

### 2.3 Data grouping

Assuming the number of clusters and the initial cluster centers are given, the data grouping in RK-means clustering algorithm is carried out as:

1. For each data vector $\mathbf{x}_k$, compute its probability of belonging to its nearest cluster center as Eq.(5).
2. Update the clusters by minimizing the following objective function:

$$J_{rkm}(\mathbf{w}) = \sum_{k=1}^{n} t_{kf} \parallel \mathbf{x}_k - \mathbf{w}_{f(\mathbf{x}_k)} \parallel^2 . \tag{6}$$

The cluster centers $\mathbf{w}$ are obtained by solving the equation

$$\frac{\partial J_{rkm}(\mathbf{w})}{\partial \mathbf{w}} = 0. \tag{7}$$

The existence of the solution to Eq.(7) can be proved easily if the Euclidean distance is assumed. To solve this equation, we first compute an approximate $\mathbf{w}$ with the following equation:

$$\mathbf{w}_j = \frac{\sum_{k=1}^{n} \delta_j(\mathbf{x}_k) t_{kf} \mathbf{x}_k}{\sum_{k=1}^{n} \delta_j(\mathbf{x}_k) t_{kf}}, \tag{8}$$

where

$$\delta_j(\mathbf{x}_k) = \begin{cases} 1 & \text{if } j = f(\mathbf{x}_k) \\ 0 & \text{otherwise} \end{cases} . \tag{9}$$

Then $\mathbf{w}$ can be obtained by applying Newton's algorithm using the result of Eq.(8) as the initial values. Since there is a dependency between the cluster centers and the probability of belongingness, the step one and two are performed iteratively until $\mathbf{w}$ converges.

### 2.4 Redundant cluster deletion

The definition of redundant cluster includes two cases: 1) the assumed number of clusters is smaller than the real one that the dataset has, where redundant real clusters exist; 2) the assumed number of clusters is greater than the real number of clusters, where redundant assumed clusters exist.

In case (1), the RKM algorithm can easily delete the pixels of redundant real clusters by assigning them low reliability with Eq.(1). That is because such pixels are always far away from any of the initial cluster centers and the output of Eq.(1) indicates that they should not belong to any initial clusters. **Fig**.10 shows the performance of RKM in such case.

In case (2), when the assumed number of clusters is greater than the one that a dataset really has, there will be some redundant assumed clusters. Such redundant assumed clusters will scramble for the data vectors that should belong to one cluster, thus those data vectors will be divided into two or more clusters forcibly. Since this partition does not match with the natural structure of those data vectors, the clustering will become unstable and sensitive to noise.

To solve this problem, we merge two redundant clusters into one cluster according to their variance and reliability.

**Fig**.5 illustrates a reliability field in a two-dimension space around two cluster centers. The

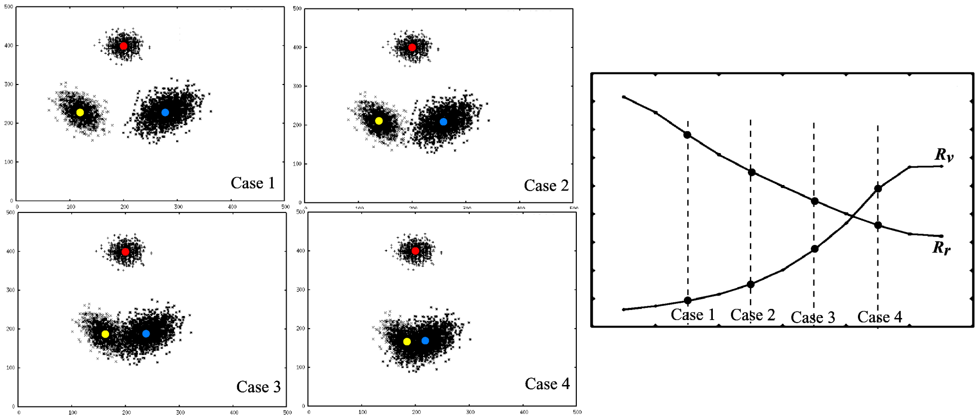



**Fig. 7** Investigation on the possibility of merging two clusters.

data vectors located on the line connecting the two cluster centers will have higher reliability values than the other data vectors. Such data vectors will have the effect to attract the two cluster centers together, and this effect will become stronger when the two cluster centers really get closer.

**Fig**.6 shows an experimental result of clustering one crowd of data vectors with RK-means clustering algorithm, when given two initial clusters. Here the two cluster centers (red circle) get closer and closer as the iteration increases. Then, the average reliability of the data vectors of the two clusters will decrease gradually according to Eq.(1).

We considered that it is possible to judge if two clusters should be merged by checking their average reliability and variance. Although the variance of a data set of $n$ dimensional vectors are generally described by its covariance matrix, here we compute a single value to describe the dispersion of the data vectors of cluster $i$ as following:

$$v(i) = \frac{\sum_{\mathbf{x}_k \in \mathrm{cluster}(i)} (\mathbf{x}_k - \overline{\mathbf{w}_i})^2}{N}, \tag{10}$$

where $N$ is the number of data vectors in cluster $i$. The average reliability of cluster $i$ can be computed by:

$$r(i) = \frac{\sum_{\mathbf{x}_k \in \mathrm{cluster}(i)} \mathcal{R}(\mathbf{x}_k)}{N}. \tag{11}$$

In order to establish a standard for checking if two clusters should be merged into one big cluster or not, we did some experiments to segment various simulated data set with the method described in section 2.2, and analyzed the relation between the average reliability, the dispersion and the distributions of the data vectors.

In **Fig**.7, Case 1∼4 show the clustering results of data sets under different distribution. In order to check if two clusters should be merged or not, we compare the average reliability and the dispersion before and after merge. The result of this comparison is obtained by the following equations:

$$\begin{cases} R_v(i,j) & = \frac{v(i)+v(j)}{v(i\cup j)} \\ R_r(i,j) & = \frac{r(i)+r(j)}{r(i\cup j)} \end{cases}, \tag{12}$$

where $i \cup j$ indicates the merged cluster from cluster $i$ and $j$.

The graph in the right part of **Fig**.7 shows the results of $R_v$ and $R_r$ of the two clusters. By analyzing these results, we discovered that the ratio of $R_r$ to $R_v$ can be used to judge if two clusters should be merged:

$$Merge(i,j) = P_m(i,j) = \frac{R_r(i,j)}{R_v(i,j)}. \tag{13}$$

If $P_m(i,j) \leq 1$ then the cluster $i$ and the cluster $j$ should be merged. Then, the redundant cluster deletion can be carried out as the following steps:

1. For each cluster ($i$), find out its nearest cluster $j$ then check if $P_m(i,j) \leq 1$.
2. If such cluster pairs do not exist, terminate this procedure.
3. Find out the pair that the distance between the two cluster centers is shortest and let $i$ and $j$ denote the number of the smaller and bigger clusters, respectively.
4. Merge the data vectors in the cluster $j$ into cluster $i$. This can be done by assigning the cluster number $i$ to all the data vectors of cluster $j$, and using $(\mathbf{w}_i + \mathbf{w}_j)/2$ as the initial cluster center of the new cluster $i$, which is the cluster merged from cluster $i$ and $j$.
5. remove the cluster $j$.
6. repeat step 1.

This procedure should be executed after each update of cluster centers. So, we add the redundant cluster deletion procedure as the third step into the process of data grouping described in section 2.2.

### 2.5 The RK-means clustering algorithm

The RK-means clustering algorithm is summarized as:

1) Initialization
i) give the number of clusters $c$ and
ii) give an initial value to each cluster center $\mathbf{w}_i, i = 1,\ldots,c$.
The initialization can be performed manually or by using an external method.
2) Iteration of data grouping
while $\mathbf{w}_i, i = 1,\ldots,c$ do not reach fixed points,
Do
i) calculate $f(\mathbf{x}_k)$ and $s(\mathbf{x}_k)$ for each $\mathbf{x}_k$.
ii) update $\mathbf{w}_i, i = 1,\ldots,c$ by solving Eq.(7).
iii) delete redundant cluster with the method described in subsection 2.4

### 2.6 Object Tracking with Reliability-based K-means Clustering

By applying the *RK-means clustering* algorithm into object tracking, we bring out the *RK-means tracker*, which is a pixel-wise algorithm that remove the target pixels from the surrounding background pixels and the mixed noise pixels.

Correspondently, the target centers are described as $\mathbf{f}_T(i) \quad i = 1 \sim K$, the representative background clusters are represented as $\mathbf{f}_B(j) \quad j = 1 \sim m$, and an unknown pixel is described as $\mathbf{f}_u$. Detailed description about $\mathbf{f}$ can be found in [13–15].

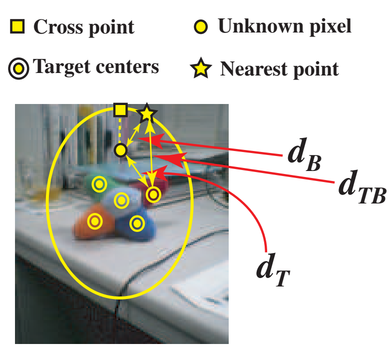



**Fig. 8** RK-means clustering on multi-color object with target and background samples.

As shown in **Fig**.8, the classification for an unknown pixel is performed between its nearest target and background clusters. The minimum dissimilarity from $\mathbf{f}_u$ to the target clusters will be calculated as:

$$d_T = \min_{i=1\sim K} \parallel \mathbf{f}_u - \mathbf{f}_T(i) \parallel, \tag{14}$$

correspondently, we can get the nearest target cluster center $\mathbf{f}_T(i)$ to $\mathbf{f}_u$. The minimum dissimilarity between $\mathbf{f}_u$ and the surrounding background samples (which are selected from the ellipse contour) is computed as:

$$d_B = \min_{j=1\sim m} \parallel \mathbf{f}_u - \mathbf{f}_B(j) \parallel, \tag{15}$$

the dissimilarity between the nearest target center and the nearest background sample ($\mathbf{f}_B(j)$) is calculated as follows:

$$d_{TB} = \parallel \mathbf{f}_T(i) - \mathbf{f}_B(j) \parallel . \tag{16}$$

The reliability of clustering $\mathbf{f}_u$ into target clusters or background clusters is estimated as:

$$\mathcal{R}(\mathbf{f}_u) = \frac{d_{TB}}{d_T + d_B}, \tag{17}$$

Then the probability that $\mathbf{f}_u$ belongs to the $i^{th}$ the target cluster is calculated as:

$$\mu_T^{(i)}(\mathbf{f}_u) = \mathcal{R}(\mathbf{f}_u) * \frac{d_B}{d_T + d_B}. \tag{18}$$

By applying $\mu_T^{(i)}(\mathbf{f}_u)$ into Eq.(8), the new target center $\mathbf{f}_T^{(new)}(i)$ can be calculated. To deal with the abrupt color shift caused by reflection or illumination changes, we select the method mention in [14] to get the final target color $\mathbf{f}_T^{(t)}(i)$ from $\mathbf{f}_T^{(new)}(i)$. Detailed description of the update of search area, tracking failure detection and recovery can be found in [13–15]

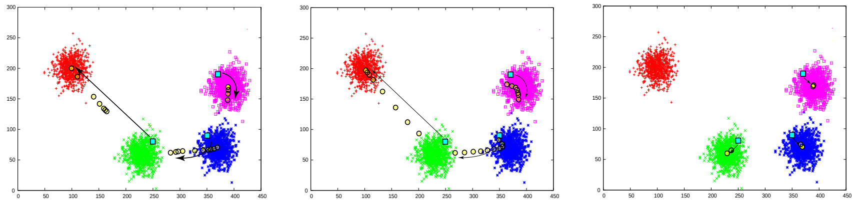
## 3. Radial Sampling

In order to reduce the computational cost of RK-means tracker, one of the most effective methods is to reduce the volume of the dataset which will be processed while clustering. Therefore, the sampling technique is required for this purpose. Here, we proposed a radial sampling method (as shown in the bottom of **Fig**.9) which can improve both the speed and robustness of our tracking algorithm.

The radial sampling is set up according to such an assumption: when the target object is solid and included within the search area, the closer a pixel is to the center of the search area, the more possible it is a target pixel. In this research, because the search area is described by an ellipse model, this assumption becomes: the closer to the ellipse center, the more possible a pixel is a target pixel. In this research, because the search area is updated according to the distribution of the detected target pixels, the target center is the center of search area. Therefore, this assumption is reasonable.

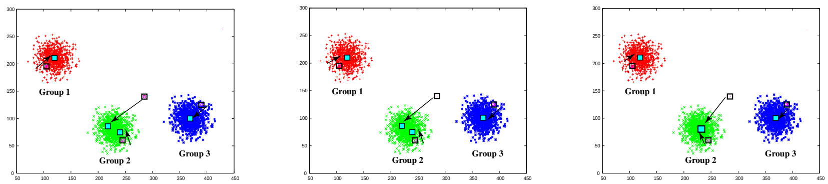
This radial sampling method is performed in the following way: all the sample points are selected from the ellipse radius at constant intervals; each ellipse radius is produced at the intervals of constant degree.

With this radial sampling, the ellipse center has high density and ellipse boundary has low density



(A) EKM (B) FKM (C) RKM

**Fig. 10** Clustering results when an undefined cluster exists.




(A) EKM (B) FKM (C) RKM

**Fig. 11** Clustering results when a redundant cluster exists.

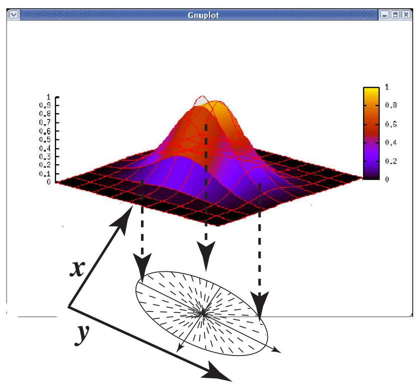



**Fig. 9** Radial sampling and the center-weighted density map.

as shown in **Fig**.9. Since there are few sample points selected from the regions far away from the ellipse center, the affection of noise pixels (or the mixed background pixels) has been reduced, thus improving not only the processing speed but also the robustness of the RK-means clustering.

## 4. Experiment and discussion

For concision, hereafter the Hard (or existing) K-means clustering, Fuzzy K-means and our RK-means are abbreviated as *EKM*, *FKM* and *RKM*, respectively. In all the experiments, we give the same initial number of clusters and the cluster centers points for the same image sequence, and all the algorithms run at the same iterations for one image frame.

### 4.1 Evaluating the effectiveness of RK-means

We compared the clustering results of RKM with those of EKM and FKM within the same simulated data sets.

**Fig**.10 shows the performance of the proposed RKM in the case that the assumed number of clusters is smaller than the real one that a dataset has. In this figure, four clusters exist but the assumed number of clusters is three, and the sky blue "□" denotes the initial cluster centers and the yellow "●" shows the updated cluster centers during iteration, and the resulted cluster centers are the ones pointed by arrows. Here, both the EKM and FKM failed to give correct clustering result according to the initial points. Our RKM algorithm correctly found three clusters while ignored the fourth redundant cluster according to the initial position. That is because RKM will give extremely low reliability to the data vectors in the fourth redundant cluster according to Eq.(1). While updating the cluster centers, such low-reliability data vectors would have little affection on the correct clusters and could be ignored, thus, the RKM could correctly find the three cluster centers.

As shown in **Fig**.11, we give four initial clusters for the data set only having three groups of crowded data vectors. The hollow "□" indicates the initial cluster centers, and the solid sky blue "□" shows the resulted cluster centers. The EKM and FKM algorithm brutally divided the data vectors of *group 2* into two separated clusters (**Fig**.11(A) and (B)). Meanwhile, the RKM removed one redundant cluster during the iteration of data grouping and gave only one cluster for that data group (see **Fig**.11 (C)).

### 4.2 Convergence of the RK-means

One important thing that needs to be confirmed is whether the RKM clustering algorithm converges. We used the IRIS dataset☆ that is a com-

☆ http://www.ics.uci.edu/~ mlearn/databases/iris/iris.data mon database often used for testing data grouping algorithms to check the convergence of RKM. The IRIS dataset has 150 data points. It is divided into three groups and two of them are overlapping. Each group contains 50 data points. Each point has four attributes.

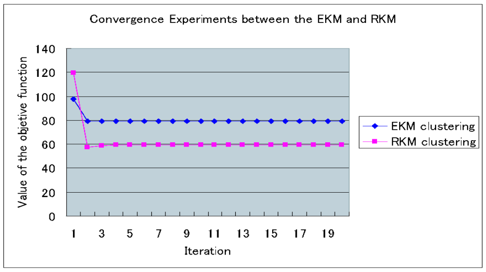



**Fig. 12** Convergence of the EKM and the RKM algorithms.

From the experimental result shown in **Fig**.12, we confirmed that the convergence of the RKM algorithm is as good as the EKM algorithm, and the speed of convergence is not slower than the EKM.

### 4.3 Affection of the cluster volume on RKM

Another important element that will affect the performance of RKM is the volume (sometimes the size) of each cluster.

The left image of **Fig**.13 shows a case that both the distribution and volume of one cluster increase. Here, the distribution of the left cluster is a circle, and both its distribution and volume are fixed, the distribution of the right cluster is an ellipse and its distribution will increase as its volume increase. In the graph of **Fig**.13, the horizontal axis shows the ratio between the volume of each cluster (and this ratio will become larger and larger), the vertical axis is the average reliability of each cluster. The red line shows the changes of the average reliability of the left cluster and the green one shows that of the right cluster. From this image, it is obvious that the increase of the distribution and volume of one cluster has very little affection on the average reliability of each cluster calculated by the RKM. In other words, the RKM is insensitive to the changes of volume and distribution of clusters.

**Fig**.14 shows the comparative experiment on the displacement between the EKM and RKM in the case of **Fig**.13. The vertical axis shows the displacement (pixel), the horizontal axis is the same as **Fig**.13. The red line shows the displacement of the EKM and the green line shows that of the RKM. (A) shows the comparative results of RKM and EKM on the left cluster of **Fig**.13, correspondently, (B) shows the results on the right cluster. According to both graphics, obviously the RKM works better than the EKM in this case. That is because the EKM will mistakenly take some outliers as the pixels belonging to some clusters. While the RKM can give low reliability to such outliers.

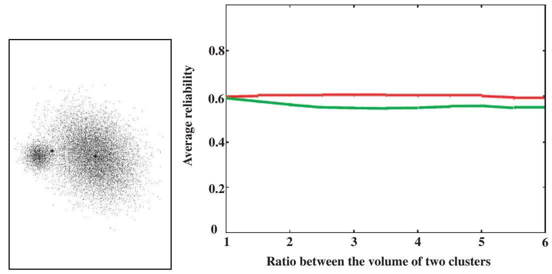



**Fig. 13** Experiment of the RK-means clustering under variable distribution. In the right graph, the horizontal axis shows the ratio between the volumes of two clusters, the vertical axis shows the average reliability of each cluster. The red curve shows the average reliability of the left cluster, and the green one shows that of the right cluster.

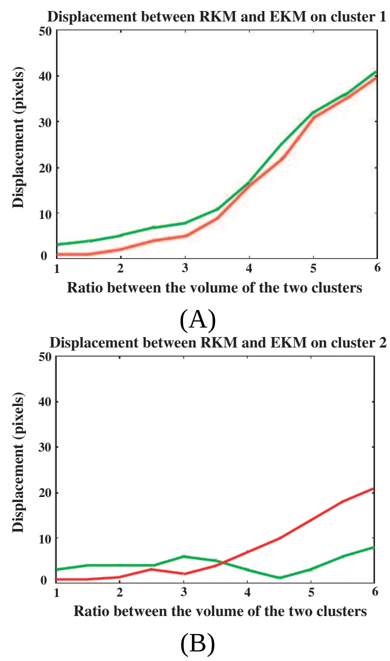
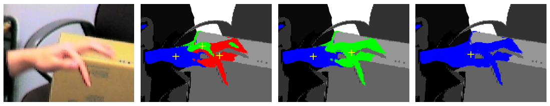



**Fig. 14** The displacements between the EKM and RKM on each cluster in the case of **Fig**.13. Green curve: the displacement of RKM; Red Curve: the displacement of EKM

**Fig**.15 shows the result of an image segmentation experiment for testing the effect of Section 2.4. **Fig**.15(A) was extracted from frame 285 in an image sequence used for the experiments shown in **Fig**.18. The yellow cross indicates the target cluster centers. The target clusters become less during the iteration of data grouping, as shown in (B)$\sim$(D).



(A) (B) (C) (D)

**Fig. 15** Image segmentation with redundant cluster deletion. A: Input Image; B: Iteration 1; C: Iteration 2; D: Iteration 3.

These results show that the RKM algorithm can automatically delete the redundant clusters.

## 4.4 Object tracking with RK-means tracker

### 4.4.1 Initialization

Different from the existing initialization method (like selecting rectangle or ellipse region around the target object), the initialization of RK-means tracker is performed by manually selecting several initial target points that can represent the main target colors. That is because conventional initialization methods can only define the target position, they can not provide the correct number of target colors which is needed. Since automatic color detection has beyond object tracking, we manually perform this initialization.

After getting the initial search area, the other background points are selected from the circle contour at the fixed interval of 45°. Therefore, $m$ of Eq.(15) is eight. Because the background sample points are selected from the boundary of search area, updating the search area is equal to updating the background samples. More detailed description about selecting and updating the background samples are available in [13–15].

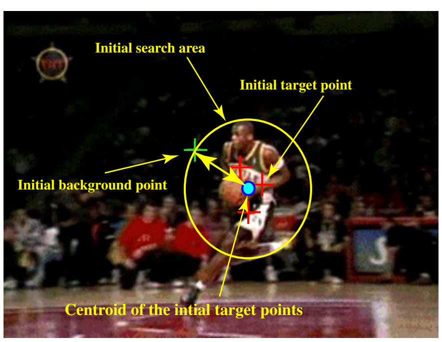



**Fig. 16** Initialization for multi-color object tracking.

As shown in **Fig**.16, in the first image, $K$ points are manually selected as the initial target points, where the number of K depends on the main colors contained by the target object. Then, the centroid of these initial target points will be calculated and be regarded as the center of the initial search area. Thirdly, one initial background point out of the target object will be selected. Finally, an initial search area will be constructed, which is a circle in the first image. The center of search area is located at the centroid of the initial target points and its radius is the distance between the initial background point and the center of circle.

### 4.4.2 Performance of RK-means tracker

Hua *et al.* [13] have compared☆ K-means tracker with some famous tracking algorithms (like template matching [18] and mean shift [12]). Here we only compare our RK-means tracker with the K-means tracker.

**Fig**.17 shows the experimental result of RK-means tracker in the case that the target person disguises himself while tracking. Applying the RK-means clustering between the target and background samples, the RK-means tracker can distinguish the obstacles (like sun glasses, towel, hat) from the target. By updating the search area dynamically, the RK-means tracker can also deal with the rotation of target object.

**Fig**.18 shows a sequence of hand tracking. As for the K-means tracker, the tracking failed since frame 285. In frame 285, since the color of some surrounding background parts (e.g. a corrugated carton) was similar to the skin color for some degree, the k-means tracker mistakenly took them as the target pixels. This caused the update of the search area to fail, so did the tracking. As for our RK-means tracker, in frame 285, the influence of the background parts having color similar to the skin color was effectively repressed through the reliability evaluation. As the result of it, the RK-means tracker could detect the target area (e.g. hand) and update the search area correctly.

**Fig**.19 shows the performance of RK-means tracker when the target person is fighting against another person ☆☆. Since two persons are grabbling with each other, the other person is mixed into the search, which is difficult for the conventional algorithms to work correctly. But the RKM algorithm can successfully discriminate the target person from another one, because the pixels of another person has low reliability to be considered as the target ones.

**Fig**.20 shows the performance of RK-means tracker on multi-color object. In this sequence, as for being the nonrigid human body, the target shape changed greatly and the colors of both target and background had been changed by the abrupt camera flash (Frame 52). Our RK-means tracker correctly

☆ http://vrl.sys.wakayama-u.ac.jp/VRL/studyresult/study_result_3_en.html

☆☆ The video is taken from PETS-ECCV2004.

Frame 0009 Frame 0341 Frame 0779 Frame 1080

**Fig. 17** Disguise experiment. The RK-means tracker is applied to track the human head where the target person disguises himself.

Results of the K-means tracker. The missing target center and similar color of background make it failed since Frame 285.

Results of the RK-means tracker.

Frame 220 Frame 258 Frame 285 Frame 305

**Fig. 18** Results of comparative experiment with K-means tracker [13] and RK-means tracker under complex scenes.

detected and recovered from this color changes, where the ellipse contour was painted with green to indicate this recovery.

All the experiments were performed with a desktop PC with a 3.06GHZ Intel XEON CPU, and the image size was $640 \times 480$ pixels. When the target size varied from $100 \times 100 \sim 200 \times 200$ pixels, the processing time of our algorithm was about $9 \sim 15$ms/frame.

## 5. Conclusion

In this paper, we have proposed a robust pixel-wise object tracking algorithm which is based on a new reliability-based K-means clustering algorithm (called as RK-means tracker). By applying the RK-means clustering into both the target and background samples, the RK-means tracker can give low reliability value to both the background and noise pixels to delete them, thus it achieves the robust object tracking under the cluttered condition. By dynamically updating the search area according to the distribution of the target pixels, the RK-means tracker can follow the target deformation smoothly. When the target is partial lost, the tracking failure detection and recovery processes are brought out to solve this problem. Through the extensive experiments, we have confirmed that the effectiveness and advantages of the RK-means tracker.

An object tracking algorithm requires an initialization, which can be performed manually or by using the information provided by some object detection techniques. In the case of RKM tracker, the information for this initialization is the centers and the colors of each pixels groups of a target object. However, most existing object detection algorithms only provide the position and the shape (often as a rectangle or an ellipse) of the detected object. In order to use them for the initialization of the RKM tracker, an additional process that segment the object into several groups of pixels, each has an unique color, is required.

**Acknowledgments** This research is partially supported by the Ministry of Education, Culture, Sports, Science and Technology, Grant in-Aid

Frame 055 Frame 107 Frame 125 Frame 168

**Fig. 19** Applying the RKM into object tracking. This video is taken from the public database of PETS-ECCV2004, and in this sequence the target person is fighting against another person. The RKM-based tracking system has successfully tracked the target person against the other person.

Frame 001 Frame 052 Frame 062 Frame 091

**Fig. 20** Performance of RKM tracker on the multi-color object under complex background condition

for Scientific Research (A)(2), 16200014 and (C) 18500131, (C) (2) 15500112.

## References


1) J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithm, Plenum Press, New York, 1981
2) R. Krishnapuram and J.M. Keller, A Possibilistic Approach to Clustering, *IEEE Trans. Fuz. Sys.*, Vol. 1, No. 2, pp.98-110, 1993.
3) Chintalapudi K.K., Kam M., A noise-resistant fuzzy C means algorithm for clustering, *FUZZ-IEEE*, Vol.2, pp.1458-1463, 1998.
4) J.M. Jolion, etc., Robust Clustering with Applications in Computer Vision, *PAMI*, vol. 13, No. 8, 1991.
5) Ron Zass and Amnon Shashua A Unifying approach to Hard and Probabilistic Clustering, *ICCV*, Vol 1, pp.294-301, 2005.
6) J.Hartigan, M.Wong, Algorithm AS136: A K-means clustering algorithm, *Applied Statistics*, vol.28, pp.100-108, 1979.
7) N.Peterfreund, Robust tracking of position and velocity with Kalman snakes, *PAMI*, Vol.22, pp.564-569, June 2000.
8) C.Gr*ä*βl, etc., Illumination Insensitive Template Matching with Hyperplanes,*DAGM*, pp.273-280, 2003
9) M.Isard and A.Blake, CONDENSATION-Conditional density propagation for visual tracking, *IJCV*, Vol.29, No.1, pp.5-28, 1998.
10) K.Toyama, A.Blake, Probabilistic Tracking in a Metric Space,*ICCV*, Vol.2, pp.50-57, 2001
11) A.D.Japson, etc., Robust Online Appearance Model for Visual Tracking, *PAMI*, Vol.25, No.10, 2003
12) D.Comaniciu, V.Ramesh and P.Meer: Kernel-Based Object Tracking,*PAMI*, Vol.25, No.5, pp.564-577, 2003
13) C.Hua, H.Wu, T.Wada, Q.Chen, K-means Tracking with Variable Ellipse Model, *IPSJ Transactions on CVIM*, Vol.46, No.Sig 15(CVIM12), pp.59-68, 2005
14) C.Hua, H.Wu, Q.Chen, T.Wada, K-means Tracker: A General Algorithm for Tracking People, *Journal of Multimedia*, Vol.1, Issue 4, pp. 46-53, 2006
15) C.Hua, H.Wu, Q.Chen, T.Wada, Object Tracking with Target and Background Samples, *IEICE*,Vol.E90-D, No.4, pp.766-774, 2007
16) R. Collins and Y. Liu: "On-line Selection of Discriminative Tracking Feature", *ICCV*, Vol.2 pp.346-352 2003
17) H.T.Nguyen and A. Semeulders: "Tracking aspects of the foreground against the background", *ECCV*, Vol.2 pp.446-456 2004
18) A. Rosenfeld and A.C. Kak: " Digital Picture Processing, Computer Science and Applied Mathematics", *Academic Press*, New York, 1976
19) Stauffer, C. and Grimson, W.E.L, "Adaptive Background Mixture Models for Real-time Tracking", *CVPR*, pp.246-252, 1999
20) J.Barron, D.Fleet, S.Beauchemin: "Performance of optical flow techniques", *IJCV*, Vol.2, No.1, pp. 42-77, 1994
21) C.Zhang, Y.Rui, "Robust Visual Tracking via Pixels Classification and Integration", *ICPR*, pp. 37-42,

2006

(Received September 03, 2007)
(Accepted March 11, 2008)

**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 received his Ph.D. degree in computer vision from the graduated school of system engineering at Wakayama University in 2007. He is a member of IPSJ. Now, he is working at the Institute of Scientific Industrial Research of Osaka University as a post doctor. He received the Funai award of IPSJ Digital Courier for young researchers in 2006. His research interests include face detection and recognition, color-based object tracking.

**Haiyuan Wu** received the 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, IEICE, 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 and 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 the IEICE, IPSJ, the Japanese Society for Artificial Intelligence and IEEE.