K-meansトラッカー：
失敗を自動的に検出・回復する対象追跡法

華春生† 呉海元† 陳謙† 和田俊和†

† 〒640-8510 和歌山県和歌山市栄谷930 和歌山大学システム工学部情報通信システム学科
E-mail: †{hua, wuhy, chen, twada}@vrl.sys.wakayama-u.ac.jp

あらまし 本稿では、「K-meansトラッカー」と名づけた対象追跡法を提案する。この方法は K-means クラスタリングによって箱円モデル内の各画素をターゲットと非ターゲットに分けることによって安定な追跡が継続できるという簡便性を持っている。提案手法は 3 つのアイデアを同時に採用することによって追跡の安定性を確保する：1）追跡対象を構成する画素の色の類似性と空間的接近性を同時に表現するため、各画素を x,y の空間情報をと Y.U.V の色情報からなる 5 次元ベクトルとして扱う。これは対象の空間的追跡だけでなく、色属性の更新も同時に保つことを意味している。これによって、追跡対象の色が変わても安定に追跡できる。2）ターゲットの検出結果に基づいて非ターゲットを表現する可変箱円モデルのパラメータをフレームごとに更新する。これによって、ターゲットの大きさ形状が大きく変わってもロバストに追跡できる。3）一部の追跡が失敗しても、ターゲットと非ターゲット情報を重ねて、追跡の失敗を自動的に検出・回復する機能を持たせることによって追跡の安定性を保障している。また、提案手法では、ビデオレコードで追跡を実現するために、K-means クラスタリング法の代わりに、より高速な \( N+1 \) 1 クラスタリングを提案する。多数の実画像系列の実験結果より、他の手法に比べて本手法が背景の混入、追跡対象の変形、色の変化、などに対して極めて簡便であることを確認した。

キーワード 複数色対象の追跡、背景混入、可変箱円モデル、K-means クラスタリング

K-means Tracker:
A Failure Detectable and Recoverable Tracking Method Against Interfused Background

Chunsheng HUA†, Haiyuan WU†, Qian CHEN†, and Toshikazu WADA†
† Faculty of System Engineering Wakayama University Wakayama City
E-mail: †{hua, wuhy, chen, twada}@vrl.sys.wakayama-u.ac.jp

Abstract This paper presents a K-means tracker, which is a novel visual tracking algorithm. One of the most distinctive points of this algorithm is the robustness against interfused background, because this algorithm discriminates “target” pixels from “background” pixels in the tracking region based on K-means clustering using both positive and negative information of the target. Since the clustering is performed in the 5D feature space of position \((x,y)\) and color \((y,u,v)\), target clusters automatically follow the target positions as well as the target colors. This adaptive nature guarantees the robustness against target color change. By using a variable ellipse model to represent both the target shape and the surrounding non-target pixels, the algorithm can cope with the change of the scale and the shape of the target object flexibly. In spite of this robustness, the tracking sometimes fails. In such cases, the algorithm automatically detects the tracking failure and recovers from it based on the positive and negative information. We achieve the video-rate processing by replacing the original K-means clustering algorithm with a newly developed \( N+1 \) 1 clustering algorithm. We have implemented a prototype system based on the algorithm and confirmed the effectiveness through extensive experiments.

Key words Tracking of multiple-color target, Interfused background, Variable ellipse model, K-means clustering
1. Introduction

This research realizes an algorithm for tracking an object having 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 [1]-[19].

Template matching [5], [6] 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 [7] (or kernel-based tracking [8]) 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 [9] (also called as particle filter, or MCMC [10]-[13]) realizes stable object tracking through cluster and occlusion by reasoning over a non-parametric distribution of joint state probability of multiple hypotheses.

Active contour [14] performs well when tracking non-rigid object. 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 [17], [18] 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) information is disregarded.

There are some algorithms that make use of non-target information [1]-[4]. With a stationary camera, moving objects can be effectively tracked in real-time by adaptive background subtraction [1]. 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 [2].

Collins et al [3] 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 [4] 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 this paper we propose a K-means clustering [19] based pixel-wise algorithm for object tracking (hereafter called as K-means tracker). Our K-means tracker can track multiple-color target objects robustly even when the object appearance changes dramatically. It can track rigid or non-rigid objects with apertures and has the abilities to detect tracking failure and recover from it. It classifies pixels in the tracking area (e.g. an ellipse) into target or non-target by applying a K-means clustering, based on a joint color-location feature vector. Our K-means tracker has the following contributions:
(1) By utilizing not only the target information but also the non-target information, it can detect tracking failure and recover from it automatically.

(2) We use \( N \) points to present the \( N \) colors of the target object and a variable ellipse model to represent the surrounding non-target pixels (background). The ellipse shape is adjusted according to the result of target detection, which makes object tracking robust against the change of scale and shape of the target.

(3) Theoretically, the distances from a pixel to \( N+E \) clusters (\( E \) is the number of non-target clusters) have to be calculated in order to tell whether the pixel belongs to target or not by K-means clustering. Since both the number and colors of non-target pixels are generally unpredictable, \( E \) may be a huge number. In this paper, by introducing a model called “Absolute Target Area” for representing both the target object and the surrounding not-target pixels, the \( N+E \) clustering can be replaced by \( N+1 \) clustering without degrading the performance of pixel classification. With this improvement, the computation time can be greatly reduced and the object tracking in video-rate becomes possible. Moreover, it improves the stability of object tracking when some background pixels have colors similar to the target object.

We select the pixel-wise algorithm for object tracking, because it is superior to the object appearance model based algorithm in dealing with object appearance changes. Unlike the existing pixel-wise tracking methods, our \( K \)-means tracker processes both positive samples (hereafter target) and negative samples (hereafter non-target) simultaneously.

1.1 Absolute target area

The boundary that separates the target region from non-target region is a key feature for object tracking. We call it as “real target boundary” (hereafter RTB). However, since the shape of RTB may be complex and the detection of RTB is generally difficult, it is not suitable to use RTB directly in object tracking.

Here we propose a model called “absolute target area” (hereafter ATA) to describe the target object and its surrounding background. The ATA is an area that contains all pixels of the target object inside and all the points on this outline are the representative non-target points around the target object.

The ATA also has the following characteristics: 1) Its outline is described by one or a set of simple and smooth closed curves; 2) It is a good approximation of RTB, that means, the outline of the ATA is not too far away from RTB. The first characteristic is to make the computation for object tracking easy and fast, and to make the model insensitive to the image noise or small change of the target appearance. The second one is for keeping a good accuracy of object’s location during tracking.

In this paper we use an ellipse to describe the ATA. This is because it satisfies the firstly mentioned characteristic of ATA and its rich degrees of freedom of deformation make it satisfying the second characteristic for most objects. Also, its simple geometric property and mathematical description reduce the computation for object tracking. Certainly, the description of ATA need not be restricted to a single ellipse. Other description may also be used when necessary.

As shown in Fig.2, the ellipse center is located at the centroid of target object, and the ellipse describes the outline of ATA that represents the non-target pixels surrounding the RTB approximately.

1.2 Pixel classification in ATA

A pixel in an image is described by a 5D feature vector \( f = [c \ p]^T \), where, \( c = [Y U V]^T \) describes the color and \( p = [x y]^T \) describes the position of the pixel.

The distance between two pixels \( f_1 \) and \( f_2 \) in the 5D feature space is defined with a joint metric as:

\[
\|f_1 - f_2\|^2 = \|c_1 - c_2\|^2 + w_p\|p_1 - p_2\|^2,
\]

where \( w_p \) is a pre-defined coefficient.

Because the ellipse describing the outline of ATA is an approximation of the RTB, it contains all the target pixels inside, meantime it may also contain some non-target pixels (Fig.2). In order to ensure a successful object tracking, it is necessary to remove those non-target pixels. This can be accomplished by using a K-means clustering algorithm.

As shown in Fig.3, 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 target cluster center and describe it by \( T_r(i) = [c_r(i) \ p_r(i)]^T \), \( i = 1 \sim N \). Non-target pixels on the ellipse are described as \( f_{Nc}(j) = [c_{Nc}(j) \ p_{Nc}(j)]^T \), \( j = 1 \sim E \), where \( E \) is the number of pixels on the ellipse, and an unknown pixel is described by \( f_u = [c_u \ p_u]^T \).

If an unknown pixel is a target pixel, it should be near to one of the \( N \) target cluster centers; otherwise, it should be near to one of the \( E \) non-target pixels on the ellipse. According to this consideration, an unknown pixel \( f_u \) can be
the non-target pixels in the pixels on edges, the number of such pixels can be considered as inﬁnitesimal compared to the entire image. Thus it is impossible to build a compact model containing only \(E_{\text{small}}\) clusters (\(E_{\text{small}} \ll E\)) to describe the non-target pixels on the ellipse.

Therefore, the K-means clustering can be very computational expensive, especially when the ellipse is big and so is \(E\). Mathematically, there are inﬁnite points on an ellipse so \(E\) can be considered as inﬁnite (\(E \to \infty\)).

In general, it is reasonable to assume that the color of pixels has some spatial smoothness, which means, the colors of two pixels will not have much difference if the 2D distance between them is short. Although this will not be true for the pixels on edges, the number of such pixels can be considered as inﬁnitesimal compared to the entire image. Thus such pixels will have very little inﬂuence on a statistics based algorithm and can be neglected safely.

Since the outline of ATA is an approximation of RTB, the non-target pixels in ATA should be near to the ellipse. Therefore, as shown in Fig.3, if an unknown pixel in ATA is a non-target pixel, its color should be similar to the color of the nearest point on the ellipse. Thus \(d_N\) in Eq.(3) can be approximated by the distance from the unknown pixel to the nearest point on the ellipse and Eq.(3) can be simpliﬁed as

\[
d_{NT} \cong \|f_{N_n} - f_u\|^2, \tag{4}
\]

where, \(f_{N_n}\) is the feature vector of the nearest point. So the computation cost for estimating \(d_N\) is reduced from \(E\) to 1 and the total computation cost for classifying an unknown pixel is reduced from \(N+E\) to \(N+1\). For this reason, we name it as \(N+1_\infty\)-means clustering.

As shown in Fig.3, the line connecting the ellipse center and the unknown point intersects with the ellipse at the cross point. Because the 2D distance from an unknown pixel in ATA to the nearest point on the ellipse is short, the distance between the nearest point and the cross point (\(f_{N_c}\)) is also short. Thus \(f_{N_n}\) in Eq.(4) can be replaced by \(f_{N_c}\) safely. This makes the \(N+1_\infty\)-means clustering even faster.

2. Automatic Update of ATA

When all target pixels have been detected in a new frame of image, the ATA has to be updated to satisfy its characteristics.

In this paper, we assume that the distribution of the pixels of a target object can be described by a normal distribution. Therefore, a dynamic Gaussian probability function is used to describe the appearance of the target object. Because this is a statistical representation, it has the following advantages over a concrete geometric model to describe the appearance of that object: 1) It is insensitive to the small deformation or the geometrical transformation of the object; 2) It reduces the effect of mis-detection of some target pixels due to image noise, etc.

The Gaussian probability density function of a target object describing by a set of pixels \(Z = [Z_1, Z_2, \ldots, Z_n]\) can be expressed as:

\[
Z \sim \mathcal{N}(m_Z, \Sigma_Z), \tag{5}
\]

where \(Z_i = [x_i, y_i]^T \in S_T\), \(m_Z\) is the mean and \(\Sigma_Z\) denotes the covariance matrix. The Mahalanobis distance of a vector \(Z\) to the mean \(m_Z\) is given by

\[
g(Z) = [Z - m_Z]^T \Sigma_Z^{-1} [Z - m_Z]. \tag{6}
\]

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

\[
g(Z) = K, \tag{7}
\]

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

We let \(M\) be big enough (e.g. 93) so that \(E(M)\) will contain almost all the target pixels. The outline of ATA is obtained by enlarging \(E(M)\) by \(k\) (e.g. 1.25) times to ensure that all target pixels will fall in ATA and to leave enough space so that most target pixels will still be in ATA in the next frame (see Fig.4).
3. Tracking failure detection

In the current frame, if an object suddenly disappeared from where it had been in the previous frame, we consider that a tracking failure has occurred.

In the previous frame, after the pixel classification in ATA has been completed, the 2D image coordinates of $i^{th}$ color cluster center $\{x_i, y_i\}; i = 1, ..., N$ is obtained. Assuming that the target pixels of $i^{th}$ color can be described by a convex solid pattern approximately, in the current frame, if the color of pixel at $(x_i, y_i)$ is obviously different from $i^{th}$ color, we will know that the $i^{th}$ color cluster has moved too far away and we consider the tracking of $i^{th}$ color cluster has failed.

In Fig.5, $c_T$ is the color of $i^{th}$ color cluster center in the previous frame and $c$ is the color of the pixel at $(x_i, y_i)$ in current frame, $d_1 = ||c_T - c||^2$ and $d_2 = \arg\min_{j=1\sim E} ||c_T - c_N(j)||^2$. The failure of tracking $i^{th}$ color cluster can be detected by checking if $d_1 > d_2$ stands.

4. Tracking failure recovery

Assuming target object does not move too fast, the $i^{th}$ color cluster of it should not completely disappear from the ATA determined in the previous frame. When the tracking failure of $i^{th}$ color has been detected, it can be recovered by finding $f_p$ that minimizes

$$d(f_p) = ||f_T(i) - f_p||^2, \quad (8)$$

where $p \in ATA$, then using $f_p$ as the new cluster center of $i^{th}$ color.

The reliableness of the recovered cluster center of $i^{th}$ 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 recovered color is calculated as

$$P(c(i)|c_{rec}) = \frac{P(c_{rec}|c(i))P(c(i))}{\sum_{j=1}^{N} P(c_{rec}|c(j))P(c(j))}, \quad (9)$$

where $c_{rec}$ is the recovered color, $P(c_{rec}|c(i))$ can be estimated by assuming a normal distribution of $i^{th}$ color, $P(c(i))$ 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.(9) and checking if $k = i$. If $k \neq i$, 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. Experimental results

5.1 Initialization

At the beginning of object tracking, we select $N$ pixels in a $N$-color target object manually. Each pixel is used as the initial center of each color cluster. We let the centroid of these selected pixels be the initial centroid of the target. We then select another pixel out of the target object. The circle passing that pixel and whose center is at the initial target center is used as the initial outline of ATA.

The shape and position of the ATA is updated continuously according to the result of pixel classification during tracking.

5.2 Experiments with video sequences

The K-means tracker was applied to many image sequences. Some experiment samples are shown in Fig.6. Here, we present some representative results.

Comparative experiments were carried out among: 1): Sum of absolute difference based template tracking; 2): Mean-shift tracking algorithm; 3): Our K-means tracker; to show the effectiveness of our algorithm.

Some of the comparative experimental results are shown in Fig.7 and Fig.8. In Fig.7(a), the comb has rich 3D rotation and translation; in Fig.7(b), the running boy sequence demonstrates the robustness of our algorithm against occlu-
interfused background pixels. The reasons for the tracking failure of template matching are the template drift caused by background pixels interfused into the template in Fig.7(a),(b). As for mean-shift, the reasons of tracking failure are: in Fig.7(a), the color histogram in the template was polluted by the interfused background pixels; in Fig.7(b), occlusion, illumination change and the interfusion of background pixels happened simultaneously.

In Fig.8, a soccer sequence\(^\text{13}\) is a challenge because it shows a typical non-rigid object. Template matching failed due to the interfused background pixels led by the non-rigid movement of human body; mean-shift tracking algorithm was unsuccessful because of the illumination changes and interfused background pixels.

Fig.9 shows some other experimental results. For the partially transparent tea pot, even the background had the similar color to the object, our K-means tracker could track it successfully. The head tracking experiment dealt with scaling, partial occlusion and revolution. The tracking of hand dealt with various deformations such as fisting, scratching, and topological changes. Tracking iris was an example to deal with illumination changes.

All the experiments were taken on a PC with an Intel Xeon 3.06Ghz CPU. The image size was \(640 \times 480\) pixels. In the cases that the target object size verified from 140 to 200 pixels, and the number of its colors was less than six, the processing time was \(0.013 \sim 0.018\) sec/frame.

6. Conclusions

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 our \(N+1\)\(_\infty\)-means clustering algorithm. With the concept of non-target, the proposed method can detect the tracking failure and recover from it automatically, which 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. The experiments with human body, head, hand and eyes showed that our algorithm can be also applied to human interface.

文 献


(注1): The soccer movie is taken from the UEFA EURO 2004.
(a) Comparative experiment with Comb

(b) Comparative experiment with a running boy

図 7  Row 1: Template matching; Row 2: Mean-shift tracking algorithm; Row 3: Our K-means tracker
8 Comparative experiment with a soccer player.