Robust Tracking Using Particle Filter with a Hybrid Feature

Xinyue ZHAO†††a), Yutaka SATOH†††b), Hidenori TAKAUJI††††c), Nonmembers, and Shun’ichi KANEKO††††d), Member

SUMMARY This paper presents a novel method for robust object tracking in video sequences using a hybrid feature-based observation model in a particle filtering framework. An ideal observation model should have both high ability to accurately distinguish objects from the background and high reliability to identify the detected objects. Traditional features are better at solving the former problem but weak in solving the latter one. To overcome that, we adopt a robust and dynamic feature called Grayscale Arranging Pairs (GAP), which has high discriminative ability even under conditions of severe illumination variation and dynamic background elements. Together with the GAP feature, we also adopt the color histogram feature in order to take advantage of traditional features in resolving the first problem. At the same time, an efficient and simple integration method is used to combine the GAP feature with color information. Comparative experiments demonstrate that object tracking with our integrated features performs well even when objects go across complex backgrounds.

key words: object tracking, hybrid feature, particle filter, grayscale arranging pairs (GAP), color histograms

1. Introduction

Tracking is a topic of great interest in a wide variety of computer vision applications, ranging from video compression to mobile robot navigation. Tracking real-world objects is a challenging task due to the presence of noise, occlusion, clutter, dynamic background elements and confusing background colors. Various of tracking algorithms have been proposed and implemented to overcome these difficulties, such as mean shift [1], Kalman filter [2], and particle filters [3].

Among these algorithms, the particle filter has attracted considerable attention in recent years because of its powerful ability to deal with general non-linear and non-Gaussian problems. In the framework of a particle filter, one of the most important parts is the observation model. The commonly used observation models built for particle filtering tracking are edge-based [4], color-based [5], [6], and contour-based features [7]. However, algorithms relying on only one feature are less robust and suffer from various limitations in complex scenarios. The color feature is robust to noise and partial occlusion but suffers from illumination changes and the presence of confusing background colors. The edge and contour features are more robust to illumination variation compared to the color feature but are much sensitive to background clutter.

Some researchers have focused on utilizing different features together for effective tracking, such as a combination of the above features [8], [9], so that each feature can compensate each other for the weaknesses of the others. Generally, ideal features for tracking should meet the following requirements: high ability to accurately distinguish objects from the background, and high reliability to identify the detected objects. The often mentioned features are more effective in handling temporal situations than spatial relationships; hence, they are weak in distinguishing objects from the background. Furthermore, they describe the absolute properties (for example, the color, edge, or shape) of individual objects without consideration of the correlations between objects and the background.

To overcome this problem, we utilize a novel robust feature called Grayscale Arranging Pairs (GAP) [10], [11], which was originally proposed for background subtraction. Compared with other background subtraction methods, the GAP feature builds a more accurate and robust background model that is flexible enough to handle different sets of complex conditions. In this study, we improve the GAP feature to make it suitable for object tracking. Because of the outstanding performance of the GAP feature in extracting the foreground from the background, it has high sensitivity in distinguishing objects from the background, even in a complex environment (such as conditions with severe illumination changes and dynamic backgrounds). It makes use of both temporal information and global spatial information by considering stable relationships of intensity among multiple point pairs. Moreover, it represents the relative properties between objects and environment, which varies according to the positions of objects. Thus, the GAP feature provides better discrimination in many situations where a simple feature (such as the color feature) may fail, for example, under similar background conditions.

Together with the GAP feature, we also utilize the color histogram feature [6], which has been widely utilized and
is good at realizing performance to identify objects. Color provides robust and high performance in the description of objects, but it is weak in discriminating foreground objects from the background. However, this weakness can be compensated well by the GAP feature.

In this work, we introduce a hybrid observation model to integrate the GAP feature and color histogram feature in a static camera environment. The model produces a good representation of the discrimination capabilities between objects and the background, and discernment capabilities on an object itself. The two main contributions of this paper are: first, a robust and dynamic feature called GAP is adopted for object tracking which to remedy the defects of other features; second, we integrate the GAP feature with the color feature in a simple and effective manner, resulting good quality of tracking.

The remainder of the paper is structured as follows: Section 2 discusses related works and the function of particle filter. Section 3 introduces a hybrid features for tracking, including the GAP feature and the color histogram feature. Section 4 describes experiments used to validate the algorithm. Section 5 contains concluding remarks.

2. Related Works

2.1 Tracking Algorithms

In this section we focus on various observation models and techniques for object tracking based on particle filter. Many algorithms have been proposed for object tracking relying on a single feature, which is chosen according to the applications. Color distribution is robust to noise and partial occlusion and is proposed in many works [5], [6], [12]. Nummario [6] used an adaptive particle filter based on the color histogram feature. Pérez [5] also proposed a particle filter based on the color histogram feature that introduced an interesting extension to multiple-part modeling, incorporation of background information, and multiple target tracking. A discriminant-EM algorithm, proposed in [12], formulates the non-stationary color tracking problem as a transductive learning problem. Gaussian mixture models are used in this algorithm to model color distributions. Edge is another good choice for tracking tasks. The histogram of oriented gradients (HOG) [4] is an example of the technology based on histograms of edge orientations. In addition, other characteristics such as texture and contour are employed to represent objects [7].

However, using a single cue for tracking is insufficient to deal with a wide variety of environmental conditions. Recently, because of the many advantages of integrating features for tracking, several publications have described multiple features for tracking. Pérez et al. [8] introduced generic mechanisms for data fusion by using a particle filter based on layer sampling. This method searches sequentially in each direction rather than the entire space and hence is efficient in the case where the state space can be partitioned for searching. Spengler and Schiele[13] integrated multiple cues and proposed self-adaptation of each cue during tracking. In [14], Lipton et al. used shape and color information to detect and track multiple people and vehicles in a cluttered scene and monitor activities over a large area and extended time periods. Wang et al. [9] used on-line feature evaluation of a combined feature set for object tracking. Some researchers also integrated color information with motion direction and other information, which provides good tracking results [15].

2.2 Particle Filter

2.2.1 Basic Concept

Particle filters, also known as Sequential Monte Carlo methods, are sophisticated model estimation techniques based on statistical simulation. Let \( x_{t-1} \) denote the state of a tracked object at time \( t - 1 \), \( z_{t-1} \) be an observation at \( t - 1 \), and \( z_{1:t-1} \) denote a set of all observations up to \( t - 1 \). From a Bayesian viewpoint, all interesting information about the target’s state \( x_{t-1} \) is encompassed by its posterior \( p(x_{t-1}|z_{1:t-1}) \). During tracking, this posterior is recursively estimated as the new observation \( z_t \) arrives, which is realized in two major stages: prediction (1) and update (2):

\[
p(x_t|z_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})dx_{t-1}. \tag{1}
\]

\[
p(x_t|z_{1:t}) \propto p(z_t|x_t)p(x_t|z_{1:t-1}). \tag{2}
\]

Recursions (1) and (2) for the posterior require a specification of a dynamic motion model that describes the state evolution \( p(x_t|x_{t-1}) \) and a model that evaluates the likelihood of any state given the observation \( p(z_t|x_t) \).

2.2.2 Dynamic Motion Model

The state vector of a single object typically consists of kinematic and regional parameters. In this paper, in order to provide a comprehensive representation of the target, we have represented the target shape by an ellipse region. We have employed a first order motion model in order to predict the target’s position in subsequent frames. Therefore, the state vector has been defined to include the position and size of the ellipse region of the target object in the image, as well as its velocity and changes in size. Thus, the state of our tracker is defined as

\[
x_t = \{u_t, v_t, U_t, V_t, u_{\dot{t}}, v_{\dot{t}}, \dot{c}_t\}, \tag{3}
\]

where the components \( \{u_t, v_t, U_t, V_t\} \) specify the static part of the state, including the locations \( (u_t, v_t) \) and lengths of half axes \( (U_t, V_t) \) of particles, while the components \( \{u_{\dot{t}}, v_{\dot{t}}, \dot{c}_t\} \) specify the dynamic part describing the velocity \( (u_{\dot{t}}, v_{\dot{t}}) \), and the scale change factor \( \dot{c}_t \) of the particles. Hence the dynamic motion model is denoted as the following update equation

\[
x_t = Ax_{t-1} + W_{t-1}. \tag{4}
\]
where $A$ defines the deterministic component of the model and $W_{t-1}$ is a multivariate Gaussian random variable. Since we use the first order model, $A$ describes a region with constant velocity and scale change. $W_{t-1}$ includes the unknown dynamic components (e.g., acceleration) and the components that we may not be able to include in the model [16].

2.2.3 Likelihood Function

The posterior $p(x_{t-1}|z_{1:t-1}) \approx \{x^{(m)}_{t-1}, w^{(m)}_{t-1}\}_{m=1, \ldots, M}$ at time $t - 1$ is estimated by a cloud of $M$ weighted particles with the state $x^{(m)}_{t-1}$ and the respective weight $w^{(m)}_{t-1}$. At time $t$, the particles are first re-sampled according to their weights. Then, they are propagated according to the dynamic model in (4) to obtain a representation of the prediction $p(x|z_{1:t-1})$. Finally, a weight is assigned to each particle according to the likelihood function $w^{(m)}_{t} \propto p(z|x^{(m)}_{t})$. All weights are normalized to sum to one, and the posterior at time $t$ is approximated by a new weighted particle set $p(x|z_{t}) \approx \{x^{(m)}_{t}, w^{(m)}_{t}\}_{m=1, \ldots, M}$. The procedure of determining likelihood is based on feature similarity. The features of the target region are compared with those of other candidate particle regions extracted in the last frame. In this paper, we will introduce a hybrid feature for determining likelihood. The details of the likelihood function will be introduced in Sect. 3.

2.2.4 State Estimation

As mentioned above, at each time $t$, a new set of the weighted $M$ particle samples $\{x^{(m)}_{t}, w^{(m)}_{t}\}_{m=1, \ldots, M}$ is obtained. Then from this set of weighted samples, the current state $\hat{x}_{t}$ can be estimated as the weighted mean

$$\hat{x}_{t} = \sum_{m=1}^{M} w_{t}^{(m)} x_{t}^{(m)}.$$  \hspace{1cm} (5)

3. Observation Models

The traditional features used in observation models consider only the information of features based on pixel attributes such as color, typically over the object region, ignoring the spatial relationship among pixels. This makes them weak in distinguishing between objects and the complex environment, such as conditions with illumination variation and confusing background. To resolve these problems, we adopt the GAP feature, which considers a correlative relationship among pixels in both time and space domains. It has very high sensitivity in discriminating objects from the background. Furthermore, it has good ability to resist the influence of illumination changes and most cases of dynamic backgrounds. Moreover, to make use of the good performance of traditional features in the description of objects, we utilize one of the most effective features—the color histogram feature. Thus, by adopting two features simultaneously, we can make full use of the advantages of both, thereby realizing better performance.

3.1 Likelihood Based on GAP

The concept of the GAP operator was first proposed in [10], [11] for background subtraction. When the intensity of a pixel varies mainly due to the background fluctuation (illumination variations, waving leaves, and camera vibration), it is difficult to predict its temporal variation. However, the problem becomes simple by using point pairs with a stable relationship. Based on such consideration, the GAP feature makes use of the properties of stable point pairs, showing a strong performance in resisting the influence of environmental fluctuations.

Compared with other background subtraction methods, the GAP feature has two advantages. First, it is more robust to the influence of severe illumination changes and background motion. Some approaches use background models built by historical judgments of each pixel. Although pixel correlations are considered in the modeling, other approaches typically use information between neighboring pixels or pixel blocks, whereas, the GAP feature shows that useful pixel correlations are not limited to neighboring pixels or pixel blocks. Using both temporal and spatial information, the GAP feature utilizes point pairs with a constant intensity difference throughout the entire image, so that illumination variance and some background motion, such as common conditions of swaying leaves, repetitive camera jitter, and slow moving clouds, can be well tolerated. Second, the GAP feature has higher accuracy. It utilizes multiple point pairs, rather than a single point pair, since multiple point pairs distributed dispersely in the spatial domain can eliminate errors caused by a single point pair and provide more robust and credible information for classification.

Because of the outstanding performance in extracting the foreground from the background, GAP has higher sensitivity than other background subtraction methods for distinguishing objects from the background. In addition, the common features describe the absolute properties (for example, the color, edge, or shape) of an individual object without considering the correlations between the objects and the background. A core part of the GAP feature varies according to the positions of objects in environment, which represents the relative properties between objects and environment.

The GAP feature denotes the probability of one pixel belonging to the foreground. In our framework, unlike many traditional methods which consider the history states of a pixel to decide whether it belongs to the foreground, we do this by considering the relationship between this pixel and several statistically chosen reference pixels. For a target pixel $P$, suppose we have $N$ positive reference pixels which statistically have higher intensity than $P$, and $N$ negative reference pixels which statistically have lower intensity than $P$ (refer to [11] for the details of statistically choosing the reference pixels). In a new frame, $P$ is classified as the background when its intensity is normal: lower than those of the positive reference pixels and higher than those
of the negative reference pixels; contrarily, \( P \) is classified as the foreground when its intensity is abnormal: higher than those of the positive reference pixels or lower than those of the negative reference pixels. In details, two probabilities that concern whether \( P \) belongs to the foreground are calculated as follows: positive probability \( \xi^+ = n^+ / N \) (The number \( n^+ / N \) denotes that the intensity of \( P \) is higher than that of \( n^+ \) out of \( N \) positive reference pixels) and negative probability \( \xi^- = n^- / N \) (The number \( n^- / N \) denotes that the intensity of \( P \) is lower than that of \( n^- \) out of \( N \) negative reference pixels). These two probabilities compose the GAP feature, and \( 0 \leq \xi^\pm \leq 1 \). The GAP feature is shown in Fig. 1 for a vivid view.

In the framework of target tracking, in an off-line operation, we first decide the locations of the reference pixels using training images. Then, in the on-line operation, for each pixel, we can obtain the positive probability \( \xi^+ \) and the negative probability \( \xi^- \) of the GAP feature by using the above steps for each input frame. The GAP feature is a characteristic of each pixel. In the first frame, when the target model is designed, we can calculate the histogram of the GAP feature of pixels included in the model region. Since we used the fixed target model, the histogram of the GAP feature for the target model is constant among the tracking. Then, in a new frame, when the particle is produced, the histogram of the GAP feature can be calculated in the same manner. The histograms of the particle and the target model can be used to calculate similarity. Here we adopt a histogram with the \( u \)-bin (\( u \in \{-U, \cdots, -1, 1, \cdots, U\} \)), and \( 2U \) is the total number of bins) to represent the GAP feature.

The procedure for building the histogram of the GAP feature is shown in Fig. 2. For the convenience of the explanation, in this paper, we define \( L^+ \) which has the same function as that of \( \xi^+ \), and is given as

\[
L^+ = \begin{cases} 
\pm(1 - \xi^+) \cdot U & (\xi^+ < 1), \\
\pm 1 & (\xi^+ = 1),
\end{cases}
\]

where we use the top integral function for easy of calculation. Here, \( U \) denotes the half number of bins in the histogram.

In the next step, histograms of \( L^+ \) and \( L^- \) in the particle region are calculated independently. Then, normalized histograms are produced to obtain an equal treatment in each particle region. Finally, the normalized histograms of \( L^+ \) and \( L^- \) are attached together simply to form the histogram of the GAP feature. The histogram of the GAP feature in the state \( x_t \) is denoted as \( F_{\text{GAP}} \) and can be calculated using the equation

\[
F_{\text{GAP}}^{(u)} = \gamma \cdot \left( \sum_{L^- = -U}^{1} \frac{h(L^-)}{A} \delta(L^- - u) + \sum_{L^+ = 1}^{U} \frac{h(L^+)}{A} \delta(L^+ - u) \right),
\]

where \( A \) is the number of pixels in each region for normalization, \( \delta \) is the Kronecker delta function, \( h \) is the histogram function and \( \gamma \) is used to ensure that \( \sum_{u=-U}^{U} F_{\text{GAP}}^{(u)} = 1 \).

Figure 2 (a) shows an example of the histogram of the GAP feature for a walker. \( L^+ \) shows the relatively brighter parts of objects compared to the background and \( L^- \) shows the relatively darker parts. \( L^+ \) and \( L^- \) act together to distinguish objects from the background. The smaller value between \( L^+ \) and \( L^- \) indicates the probability of the pixels being foreground pixels. As the value \( \min(|L^+|, |L^-|) \) approaches 1, the probability that the pixels are foreground pixels increases. \( P_1 \) is an example of a foreground pixel of the walker. As expected, \( L^+(P_1) = +8 \) and \( L^-(P_1) = -1 \) (in our experiment, \( U = 8 \)). Figure 2 (b) shows another case where the particle circle does not include any walker. In this case, given \( P_2 \) for example, we get \( L^+(P_2) = +7, L^-(P_2) = -8 \), which indicates that the probability that the pixel is a foreground pixel is low.

In the case with \( M \) particles, at time \( t \), we denote \( F_{\text{GAP}}^* = \{F_{\text{GAP}}^{(u)}\}_{u=-U, \cdots, U} \) as the target GAP feature model and \( F_{\text{GAP}}(x_t^{(m)}) \) as the candidate GAP feature model with a hypothesized state \( x_t^{(m)} \) of each particle. The likelihood between \( F_{\text{GAP}}^* \) and \( F_{\text{GAP}}(x_t^{(m)}) \) can be measured by the Bhattacharyya distance

\[
\rho_{\text{GAP}}(x_t^{(m)}) = \left( 1 - \sum_{u=-U}^{U} \frac{F_{\text{GAP}}^*(u) F_{\text{GAP}}(x_t^{(m)}/u)}{1} \right)^{1/2}.
\]
3.2 Likelihood Based on Color Histogram

Color provides many cues and it achieves robustness to non-rigidity, rotation and partial occlusion of objects. We utilize the HSV color space to make the algorithm less sensitive to illumination changes, which is more robust than the RGB representation. The bin index $b(y_i)$ assigns the color at the location $y_i$ to the corresponding bin. To increase the reliability when boundary pixels of an object are occluded, we use a weighting function $k(r)$ in which pixels that are far from the region center are assigned smaller weights. Then, the color feature $F_{\text{color}} = \{F_{\text{color}}^{(v)}\}_{v=1}^V$ at the center point $y$ of state $x_t$ is calculated as

$$F_{\text{color}}^{(v)} = f \sum_{i=1}^A k\left( \frac{y - y_i}{A} \right) b(y_i) - v, \quad (9)$$

where $A$ is the number of pixels in the region, and $f$ is a normalization constant ensuring $\sum_{v=1}^V F_{\text{color}}^{(v)} = 1$.

The similarity of color histograms between the template and the current frame is also computed using the Bhattacharyya distance. The likelihood of the color feature can be denoted as $\rho_{\text{color}}(x_t^{(m)})$ in the state $x_t^{(m)}$, which is calculated similarly using the formulation in Eq. (8).

3.3 Likelihood Integration

In recent years, several methods for multiple feature integration have been proposed. One of the popular approaches used to obtain feature fusion is linear combination, possibly because it yields a simple weight formula and achieves good results. Thus, we propose a simple combination based on the linear combination of the computed likelihoods for each feature. In our approach, we use the properties of both color histogram feature and GAP feature mentioned above. Under the assumption that they are independent, in the state $x_t^{(m)}$, the overall likelihood is the sum of the likelihoods of the separate features, in our case color and GAP, as shown in the following equation:

$$\rho(x_t^{(m)}) = \alpha \cdot \rho_{\text{GAP}}(x_t^{(m)}) + (1 - \alpha) \cdot \rho_{\text{color}}(x_t^{(m)}), \quad (10)$$

where $\alpha$ is a parameter to adjust the proportion between two features. Then we use a Gaussian filter with variance $\sigma$, so that samples with high weights can be chosen several times, while others with relatively low weights can not be chosen at all. The weights after filtering are denoted as

$$w_t^{(m)} = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{\rho_t^{(m^2)}}{2\sigma^2}}. \quad (11)$$
The higher the score of $w_i^{(m)}$, the better the match between the candidate particle and tracking target.

4. Experiments

In this section, we present the results of experiments on real video sequences and examine the robustness of the proposed algorithm to several challenging conditions of environment in the situations where other algorithms may fail. Five challenging sequences are used in both qualitative and quantitative analyses for evaluating our tracking algorithm, namely the CAVIAR sequence [17], the OTCBVS-BENCH sequence [18], the PETS sequence [19], the i-LIDS hard sequence [20], and the Karl-Wilhelm sequence [21]. They are taken from both indoor and outdoor scenes and vary with respect to viewpoint, illumination condition, and type of occlusion, demonstrating the robustness of our approach. The proposed method has been implemented in MATLAB and tested on a 3.2 GHz PC with 6 GB memory. The number of particle samples processed in the experiments is 100. The main parameters in all the experiments are set to be the same. In the likelihood integration, the value $\sigma$ is set as 0.5. The setting of other parameters in the particle filter framework, such as $\sigma$ in the Gaussian filter and $A$ and $W$ in the motion model, can be found in Ref. [22].

4.1 Result Visualization

To give a vivid view of how our hybrid-feature-based likelihood expresses better than the color-based likelihood, Fig. 3 shows the visualization results in one frame of the OTCBVS-BENCH sequence. To make it easier for us to observe the comparison results, the locations of particles with both hybrid-feature-based likelihood and color-based likelihood are considered to be the same. As a result, the weights of particles have a direct effect on the performance of tracking. The red circles in Fig. 3 denote high weights at the center point of the particles (the value of likelihood is larger than 0.65), while green ones denote low weights. As expected, the particle weights of hybrid-feature-based likelihood have larger differentiation around the true target state. In the color-based case, a group of particles with high weights are found on the wall by mistake, resulting in an incorrect orientation of particles at the next sampling step. Once particles appear by mistake, the mistake will increase with time. It is exactly the characteristic of the GAP feature making the tracking results are not affected by the confusing background. As shown in Fig. 3 (b), particles on the wall, whose color is similar to that of the target, have low weights. Such distribution of weights can assure correct of particle orientation.

4.2 Quantitative Comparison

We selected three sequences for quantitative comparison. The groundtruth of the CAVIAR sequence is available in [17]. The OTCBVS-BENCH and PETS2001 Database 3 sequences do not have a groundtruth in description of target regions. To obtain the true target locations of the OTCBVS-BENCH and PETS sequences, we manually inspected 60 sample images from the OTCBVS-BENCH sequence and 100 samples from the PETS sequence and selected an ellipse region containing the target for both sequences. Three particle-filter-based tracking algorithms are compared with the proposed method. One is the traditional standard color-based method [6]. The other one is a combination of color feature and the Mixture of Gaussian (MoG) method, which is one of the commonly used background subtraction techniques [23]. The third tracking algorithm is proposed by Khan et al. using MCMC-based particle filter [24]. Because the software package used by this method is available on the website [25], we directly used the results obtained by the software package for the comparison experiments. The trajectory is not shown in the result in cases where the tracker thinks that the target disappears from the scene.

In the CAVIAR sequence, two men walk, split, and meet with the third man in an entrance lobby under strong partial illumination. The crossing of the two men and the large changes in body size and shape are also challenges in this sequence with images 288 × 384 pixels in size. The top row in Fig. 4 shows the groundtruth of this dataset. The second row shows the tracking results of the standard color-based particle filter. The third row shows the tracking results with particle filter using color and MoG features. The fourth row shows the results obtained by using the method of Khan et al. The result in which no tracking trajectory is shown denotes the loss of targets in the current frame. The bottom row shows the tracking result using the proposed method. It shows that our tracker can follow the target consistently, despite the complex environment. In Frame #23, the two men intersect and one is occluded by the other. The color-based method, the combination of the MoG and color-based method, and the proposed method can handle this situation.
Fig. 4 Experimental results in CAVIAR sequence.

well. However, Khan’s method does not show any trajectory of the target object. This is because Khan’s method cannot efficiently deal with overlapped objects. In this frame, the target man is obscured by another man; hence, the tracker thinks that the target has already disappeared. Then, one man goes into a very bright area from a dark area and stops near another man for a while. Because of the sudden change in illumination, the color-based particle filter fails in tracking in Frame #95. Since MoG algorithm is weak in the sudden illumination changes, the method using a combination of MoG and color-based particle filter also fails. The results of Khan’s method show that after separation of the two men, the tracker believes that the target object has reap- peared and therefore continues to track it. However, the tracker incorrectly tracks the second man instead of the target man. Our method can still continue tracking the target correctly since it is robust to situations of illumination variation. In Frame #200, Frame #259, and Frame #309, the walking man goes through large scale and shape changes and the partial changes in illumination caused by the shadows of window ledges. Nevertheless, the tracking results obtained by the proposed method were good, smooth, and satisfied the groundtruth.

Figure 5 shows the OTCBVS-BENCH sequence, which is captured in an outdoor environment with occlusions by branches and large changes in body shape. The image size is 240 × 320 pixels. Groundtruth is shown in the first row. In the results obtained by the method that combines color and MoG features and the color-based method alone, after Frame #114, the object is totally lost as it gradually disappears behind the branches. The results of Khan’s method show that the object can be tracked successfully in Frame #24, Frame #90, Frame #114, and Frame #144. However, the method fails to track the object in Frame #162. However, in our tracking result, since the person is not completely occluded and because the proposed method is more
robust, the object could be tracked successfully.

Figure 6 shows some sample images from the PETS sequence with the presence of confusing background colors. The size of the image is 240 × 320. At the beginning of the sequence, the man wearing dark clothes walks on the light-colored ground shown in Frame #007. Then, he turns right and goes past the parked car that has a dark color similar to that of the person. This process is shown in Frame #127 and Frame #196, in which it is very difficult to distinguish the person from the car. Under this condition, all of the other three methods fail, but the proposed method succeeds. In Frame #235 and Frame #298, the person moves away from the parked car so that the shape of his body can be seen clearly. Since the time for which the person is indistinguishable is too long, the color-based method does not track the target successfully. The accuracy of the MoG algorithm is not as high as that of the GAP feature when discriminating objects from confusing backgrounds. Therefore, the combined MoG and color-based particle filter method fails in tracking objects. Khan’s method is also inadequate because it does not deal well with confusing backgrounds.

In the experiments, we use the accuracy rate (list in Table 1) to reflect the tracking performance. Successful tracking is indicated by a position error that is smaller than 10 (in pixels). In all sequences, the accuracy rate of our proposed method is much higher than that of color-based method, MoG and color based method and Khan’s method. Since the MoG algorithm cannot properly deal with environmental fluctuations, incorrect background subtraction results would mislead the combined feature, resulting in tracking results that are less accurate than those of the traditional color-based method. The tracking results of the position error curve are also shown in Fig. 7. In Khan’s method, for the results in which the object is lost, we use the length of the diagonal line of the input image as a substitute for the largest position error. It can be seen that the position error
of the proposed method is smaller than that of the other three methods tracking throughout almost the entire tracking process.

We also included four additional experiments. Two of them are traffic scenes with rapidly moving cars, shown in Fig. 8(a). The different viewing angles and scale changes make the experiment even more difficult. In the i-LIDS hard sequence, the camera vibrates slightly in the beginning frames and moving cars are under severely changing illumination conditions. The Karl-Wilhelm sequence has heavy fog, which causes low visibility. In both cases, the target cars can be tracked accurately by our proposed method. Figure 8(b) shows two other experiments from databases PETS2006 and PETS2009 having intersected objects. In the PETS2006 sequence, two medium-sized men intersect with each other. The PETS2009 sequence shows another example involving much smaller objects. The target object intersects three other objects in Frame #15, Frame #41 and

Table 1  Quantitative comparison (the successful tracking is that the position error is smaller than 10 (in pixels)).

|                | CAVIAR Mean Pos. Err. | Accur. |
|----------------|-----------------------|--------|
| Color-based method  | 43.98                | 20.65% |
| MoG+color-based method | 84.41               | 15%    |
| Khan’s method    | 346.19               | 9.3%   |
| Proposed method  | 7.80                 | 71.94% |

|                | OTCBVS-BENCH Mean Pos. Err. | Accur. |
|----------------|----------------------------|--------|
| Color-based method  | 17.15                  | 47.46% |
| MoG+color-based method | 33.26                | 47.46% |
| Khan’s method    | 83.15                  | 79.66% |
| Proposed method  | 2.10                   | 100%   |

|                | PETS2001 Mean Pos. Err. | Accur. |
|----------------|-------------------------|--------|
| Color-based method  | 17.00                   | 40%    |
| MoG+color-based method | 22.38                 | 42%    |
| Khan’s method    | 348.31                 | 13%    |
| Proposed method  | 4.76                   | 97%    |

Fig. 6  Experimental results in PETS sequence.
Fig. 7 Position error curves (in Khan’s method, for the results in which the object is lost, we use the length of the diagonal line of the input image as a substitute for the largest position error): (a) Results in CAVIOR sequence; (b) Results in OTCBVS-BENCH sequence; (c) Results in PETS sequence.

Fig. 8 Other experimental results: (a) traffic scenes, (b) scenes with intersected objects.
Frame #51 respectively. In case of Frame #15 and Frame #51, the intersected objects and the target object are dark in color, while in the case of Frame #41, the intersected object has a contrasting light color. For each case, in addition to the intersection frame, we also show the frames before and after intersection in the same interval. The experimental results show that the proposed method can track the intersected object smoothly. This is because we use the hybrid feature, and even similar objects have differences (in either the GAP feature or the color feature) that may not be detected by the human eye. Although the detected shape is sometimes not identical to that of the target object, the error rate is within the acceptable range. However, in some special cases where the objects are almost the same, our proposed method may fail to track the objects.

4.3 Parameter Discussion

The significant parameter in the proposed method is the integrating factor $\alpha$. The parameter $\alpha$ is utilized to adjust the proportion between two features ranging from 0 to 1. According to our experiments, $\alpha$ is insensitive and it achieves a good performance for a large range of changes. However, for different databases, $\alpha$ is varied to obtain an optimal value. A large $\alpha$ shows a higher reliability of the GAP feature, while a small $\alpha$ shows a higher reliability of the color feature.

We tested the performance using different values of $\alpha$ in the CAVIAR sequence (shown in Fig. 4), and the results are shown in Fig. 9. To simplify the comparison of results, we calculated the mean and variance of the position errors in Table 2 and Fig. 10. In Fig. 9, it is shown that for a large range of changes (from 0.3 to 0.95), the position error curves under different values of $\alpha$ are similar and difficult to distinguish, and the differences of mean position errors recorded in Table 2 are small (smaller than 2). Figure 10 shows that the mean position error curve with different values of $\alpha$ is almost a horizontal line, and the lowest point in the curve is 0.9. This is because in this particular sequence, the GAP feature is more reliable than the color feature. However, in other sequences, such as situations involving large differences between the object and the background or rich color information, the lowest point will appear with small values of $\alpha$. Because of the insensitivity of $\alpha$, although the position of the lowest point appears differently at different sequences, other positions that are far from the lowest point can achieve similar performances. Thus, the range of $\alpha$ is large. For simplicity, in this paper, we set $\alpha = 0.5$ in all the examples.

5. Conclusions

In this paper, we proposed an efficient and robust particle-filter-based object tracking algorithm. The observation model in the particle filtering framework is built with two different types of features: the color histogram feature, which has high ability to accurately identify the detected object, and the GAP feature, which has high sensitivity in discriminating between the background and the objects. In addition, improved robustness is achieved by a simple and efficient integration algorithm that allows for compensation of multiple features for object tracking. Experimental results have demonstrated that the proposed tracking algorithm consistently provides more precise tracking compared to the other methods.

Acknowledgement

We would like to thank Prof. T. TANAKA for his helpful suggestions and assistance in our lab. We would also like to thank Dr. K. IWATA and Mr. R. OZAKI in National Institute of Advanced Industrial Science and Technology (AIST) for their helpful discussion and contributions in the development of this work.

References

[1] D. Comaniciu, V. Ramesh, and P. Meer, “Real-time tracking of non-rigid objects using mean shift,” IEEE International Conference
on Computer Vision and Pattern Recognition (CVPR), pp.142–149, 2000.

[2] T. Broida and R. Chellappa, “Estimation of object motion parameters from noisy images,” IEEE Trans. Pattern Anal. Mach. Intell., vol.8, no.1, pp.90–99, 1986.

[3] M. Isard and A. Blake, “Condensation: conditional density propagation for visual tracking,” Int. J. Comput. Vis., vol.29, no.1, pp.5–28, 1998.

[4] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp.886–893, 2005.

[5] F. Pérez, C. Hue, J. Vermaak, and M. Gangnet, “Color-based probabilistic tracking,” European Conference on Computer Vision (ECCV), pp.661–675, 2002.

[6] K. Nummari, E. Koller-Meier, and L. Van-Gool, “An adaptive color-based particle filter,” Image Vis. Comput., vol.21, pp.99–110, 2003.

[7] J. Wang, X. Chen, and W. Gao, “Online selecting discriminative tracking features using particle filter,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp.1037–1042, 2005.

[8] P. Pérez, J. Vermaak, and A. Blake, “Data fusion for visual tracking with particles,” Proc. IEEE, vol.92, no.3, pp.495–513, 2004.

[9] J. Wang and Y. Yagi, “Integrating color and shape-texture features for adaptive real-time object tracking,” IEEE Trans. Image Process., vol.17, no.2, pp.235–240, 2008.

[10] X. Zhao, Y. Satoh, H. Takauiji, S. Kaneko, K. Iwata, and R. Ozaki, “Robust moving object detection based on a statistical model,” The Sixteenth Korea-Japan Joint Workshop on Frontiers of Computer Vision, pp.283–288, 2010.

[11] X. Zhao, Y. Satoh, H. Takauiji, S. Kaneko, K. Iwata, and R. Ozaki, “Object detection based on a robust and accurate statistical multi-point-pair model,” Pattern Recognit., vol.44, no.6, pp.1296–1311, 2011.

[12] Y. Wu and T. Huang, “Color tracking by transductive learning,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp.222–227, 2000.

[13] M. Spengler and B. Schiele, “Towards robust multi-cue integration for visual tracking,” Machine Vision and Applications, vol.14, pp.50–58, 2003.

[14] A. Lipton, H. Fujiyoshi, and R. Patil, “Moving target classification and tracking from real-time video,” DARPA Image Understanding Workshop, pp.129–136, 1998.

[15] S. Khan and M. Shah, “Object based segmentation of video using color, motion and spatial information,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp.746–751, 2001.

[16] A.D. Bimbo and F. Dini, “Particle filter-based visual tracking with a first order dynamic model and uncertainty adaptation,” Computer Vision and Image Understanding, vol.115, pp.771–786, 2011.

[17] http://homepages.inf.ed.ac.uk/rbf/CAVIAR3D/

[18] J. Davis and V. Sharma, “Background-subtraction using contour-based fusion of thermal and visible imagery,” Computer Vision and Image Understanding, vol.106, no.2-3, pp.162–182, 2007.

[19] ftp://ftp.cs.rdg.ac.uk/pub/PETS2001/DATASET3/

[20] http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html

[21] http://pub.cs.rdg.ac.uk/avss2007_d.html

[22] J. Czyza, B. Ristic, and B. Macq, “A particle filter for joint detection and tracking of color objects,” Image Vis. Comput., vol.25, no.8, pp.1271–1281, 2007.

[23] C. Stauffer and W.E.L. Grimson, “Adaptive background mixture models for real-time tracking,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.246–252, 1999.

[24] Z. Khan, T. Balch, and F. Dellaert, “Mcmc-based particle filtering for tracking a variable number of interacting targets,” IEEE Trans. Pattern Anal. Mach. Intell., vol.27, no.11, pp.1805–1918, 2005.

[25] http://www.kinetrack.org/software.html