Novel Observation Model for Probabilistic Object Tracking

Dawei Liang[1], Qingming Huang[2,3], Hongxun Yao[1], Shuqiang Jiang[3], Rongrong Ji[1], Wen Gao[4,1]
[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
[2]Graduate University of Chinese Academy of Sciences, Beijing, China
[3]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[4]National Engineering Laboratory for Video Technology, Peking University, Beijing, China
{dwliang|qmhuang|sqjiang|wgao}@jdl.ac.cn {yhx|rrji}@vilab.hit.edu.cn

Abstract

Treating visual object tracking as foreground and background classification problem has attracted much attention in the past decade. Most methods adopt mean shift or brute force search to perform object tracking on the generated probability map, which is obtained from the classification results; however, performing probabilistic object tracking on the probability map is almost unexplored. This paper proposes a novel observation model which is suitable to perform this task. The observation model considers both region and boundary cues on the probability map, and can be computed very efficiently by using the integral image data structure. Extensive experiments are carried out on several challenging image sequences, which include abrupt motion change, background clutter, partial occlusion, and significant appearance change. Quantitative experiments are further performed with several related trackers on a public benchmark dataset. The experimental results demonstrate the effectiveness of the proposed approach.

1. Introduction

Object tracking has been a hot research topic over the past decades, since it is a core component in many computer vision applications, such as video surveillance [1], human computer interaction [2], driver assistance systems [3], robotics [4], etc. Till now many researchers have devoted their effort to object tracking with a large body of literature produced, and a recent survey on this topic can be found in [5]. Although lots of tracking algorithms have been developed, object tracking is still a challenging topic. When the object's appearance varies little and the object's motion changes smoothly, simple algorithm such as template matching may work well. However, in real applications these conditions can hardly be satisfied. The object's appearance often varies significantly especially in outdoor environment, and the existence of background clutter also increases the difficulty of object tracking.

To tackle the above mentioned problems, treating visual object tracking as online foreground and background classification has attracted much attention. These methods treat appearance modeling as foreground and background classification, and mainly adopt mean shift or brute force search to perform object tracking on the generated probability map, which is obtained from the classification results; however, performing probabilistic object tracking on the probability map is almost unexplored. This paper proposes a novel observation model which is suitable to perform probabilistic object tracking on the probability map. The observation model considers both region and boundary cues, and can be computed very efficiently by using the integral image data structure. Extensive experiments are performed against mean shift method which runs on the probability map, and probabilistic object tracking methods with other styles of observation model. The experimental results demonstrate the effectiveness of the proposed approach.

2. Related works

To deal with appearance change of the tracked object, an effective way is to online model the object's appearance variation, e.g., [6], [7], [8]. By virtue of online appearance modeling, this kind of methods can capture appearance change of the tracked object due to variations in lighting, 3D pose, etc. However, without taking background information into account, these methods tend to fail in the case of cluttered background.

Collins *et al.* [9] propose to treat tracking as a binary classification problem between the tracked object and its surrounding background. They online select discriminative features from a set of linear combinations of RGB values by evaluating their variance ratios, and employ mean shift [10] to locate the object. Treating tracking as a binary classification problem opens a new area for object tracking; therefore, many classical pattern classification methods can be adapted to object tracking, such as AdaBoost [11], online boosting [12], and linear discriminant analysis [13], to just name a few. Avidan [11] combines several weak classifiers which are trained online into a strong classifier using AdaBoost, and employs the strong classifier to label

978-1-4244-6983-3/10/$26.00 ©2010 IEEE

pixels in the next frame. Hence, a probability map is constructed, and the local mode is found by mean shift as the object's location. Grabner and Bischof [12] adapt online boosting to select discriminative local tracking features, and simply identify the location with maximum confidence value as the object's location. Nguyen and Smeulders [13] treat tracking as a texture discrimination problem, where Gabor filter is used to extract texture representations of the object and its surrounding background. Then, they develop a differential version of linear discriminant analysis to locate the object.

Up to this point, we have introduced object representation methods either considering the object itself alone or combining the object and its surrounding background together. Besides object representation, another problem related to object tracking is object localization, which can be roughly categorized into two classes: deterministic methods (e.g., mean shift [10]) and probabilistic methods (e.g., particle filter [14]). Recently, probabilistic methods have attracted growing attention, since they are capable of maintaining multiple hypotheses of the object's state, and demonstrate good performance in the case of partial occlusion, abrupt motion change, and background clutter. As a representative of probabilistic methods, particle filter [14], which is also known as CONDENSATION [15] in computer vision community, is extensively investigated in the area of object tracking. In essence, particle filter is a sequential Monte Carlo approach, which uses a set of weighted samples (also called particles) to represent and propagate the posterior probability density function over time. Particle filter is first applied to track object contour in the seminal work of Isard and Blake [15], and then extended to color-based tracking [16], [17], and appearance-based tracking [7], [18].

Due to partial success of online feature selection and particle filter in object tracking, it is natural to expect better performance by combining both methods together. There are some related works, e.g., [19], [20], [21], [22]. Chen *et al.* [19] use particle filter to infer both discriminative color features and the object's state. Twenty particles are employed to infer discriminative color features, i.e., twenty color features are selected, with each feature producing a likelihood image. Then, a compound likelihood image is constructed as the weighted sum of all the twenty likelihood images for tracking, which is computationally inefficient especially for real time applications. Wang *et al.* [20] select a subset of discriminative Haar-like features using AdaBoost. Then, they embed the AdaBoost classifier into particle filter framework to construct the observation likelihood model. The classifier is also updated online by adding new features and discarding old ones. Since the features used in their method are local features, the tracker may not work well for object with low texture. Li *et al.* [21] present a cascade particle filter tracker with discriminative observers of different lifespans, which demonstrates good performance in low frame rate video. Their tracker requires a detector trained offline using local Haar-like features, and mainly focuses on face tracking. Han and Davis [22] derive a set of likelihood images from several different color spaces, from which the most discriminative likelihood image is selected by principal component analysis, and then embedded into color-based particle filter tracker to evaluate the feasibility of each particle. In their method the reference color histogram of the tracked object is fixed during tracking; therefore, it also bears the risk of tracking failure when the object's appearance changes significantly.

3. Probabilistic object tracking

Object tracking can be formulated within Bayesian filtering framework. Given all the observations (e.g., images) $Z_t = \{\mathbf{z}_1, \ldots, \mathbf{z}_t\}$ up to time t, we want to determine the posterior probability of the hidden state $\mathbf{x}_t$ (e.g., the object's location, size, etc.). Denoting $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ as system dynamic model and $p(\mathbf{z}_t \mid \mathbf{x}_t)$ as observation likelihood model, the posterior probability density function of the hidden state $\mathbf{x}_t$ can be determined by recursive Bayesian filtering [15]

$$p(\mathbf{x}_t \mid Z_t) \propto p(\mathbf{z}_t \mid \mathbf{x}_t) \int_{\mathbf{x}_{t-1}} p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \mid Z_{t-1}) d\mathbf{x}_{t-1}. \quad (1)$$

When either system dynamic model or observation likelihood model is nonlinear/non-Gaussian, no analytic solution can be obtained. In this case, particle filter, also known as sequential Monte Carlo, provides an effective approach to implement recursive Bayesian filtering by Monte Carlo simulations. The key idea is to use a set of random samples (particles) with associated weights $\{\mathbf{x}_t^i, w_t^i\}_{i=1}^M$ to represent and propagate the posterior probability density function, and to compute estimates using these samples and weights. More details about particle filter can be found in [14].

3.1. System dynamic model

System dynamic model determines how particles evolve over time. In this paper, the hidden state is chosen to be $\mathbf{x}_t = (x_t, y_t, a_t, b_t, x_{t-1}, y_{t-1}, a_{t-1}, b_{t-1})^T$, where $(x_t, y_t)^T$ is the object's centroid coordinates, a_t is the scale along x-axis, and b_t is the scale along y-axis. In some particular cases, e.g., 3D people tracking, system dynamic model can be learnt from representative sequences where correct tracks can be obtained in a certain way. However, in more general conditions, system dynamic model is very hard to be obtained. In this paper, autoregressive model is adopted and defined as

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{v}_t,\ \mathbf{v}_t \sim \mathcal{N}(0, \mathbf{\Sigma}), \qquad (2)$$

where $\mathcal{N}(0,\mathbf{\Sigma})$ is a multivariate Gaussian function with mean vector $\mathbf{0}$ and covariance Σ, and Σ is a diagonal matrix which is set as $\Sigma = diag(\sigma_x{}^2, \sigma_y{}^2, \sigma_a{}^2, \sigma_b{}^2, 0,0,0,0)$. Constant velocity models are selected for x_t and y_t, and random walk models are selected for a_t and b_t, with standard deviations $\sigma_x = 1$ pixel/frame, $\sigma_y = 1$ pixel/frame, $\sigma_a = 0.1$ /frame, and $\sigma_b = 0.1$ /frame, respectively.

3.2. Observation likelihood model

We use the method in [9] to obtain a classifier in the previous frame with some modifications. Instead of variance ratio, we adopt Bayes error rate [23] to evaluate each feature's discriminability. Moreover, as opposed to treating each selected feature independently, we combine the selected features together by constructing ensemble of Bayesian classifiers [24]. Applying the classifier to all pixels within a local window in the current frame, we can construct a probability map $W_t(\mathbf{u})$, each pixel value of which demonstrates to what degree the pixel belongs to the tracked object. Then, we construct observation model on the probability map. The advantages of this treatment are in twofold. On the one hand, in general the object has high contrast with its surrounding background in the probability map, which facilitates successful object tracking; on the other hand, the observation likelihood model can be computed very efficiently on the probability map.

Traditionally, for color-based probabilistic tracking, observation likelihood model is constructed by extracting color histogram of each particle, which is computationally inefficient especially for objects with large size. In this paper, we propose to construct observation likelihood model directly on the probability map, the advantages of which are fully demonstrated against traditional color-based probabilistic tracking as shown in section 4. When constructing observation likelihood model, both region and boundary cues are considered. Before going deep into details, let us first introduce some notations. The initial width and height of the object's bounding box are denoted as w_{ref} and h_{ref}, respectively. Given an occurrence of the state $\mathbf{x}_t$, the corresponding region $R(\mathbf{x}_t)$ in which observation will be gathered is defined by its top-left corner coordinates $(x_L, y_T)^T = (x_t - w_t/2, y_t - h_t/2)^T$, and its bottom-right corner coordinates $(x_R, y_B)^T = (x_t + w_t/2, y_t + h_t/2)^T$, where $w_t = a_t w_{ref}$ and $h_t = b_t h_{ref}$.

As for region cue, we consider the average pixel value within the region $R(\mathbf{x}_t)$ on the probability map. The observation likelihood model for region cue is thus defined

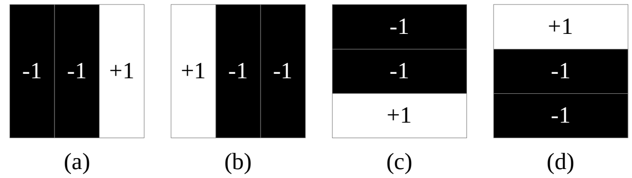

Figure 1. Correlation templates are used to compute the observation likelihoods of the left (a), right (b), top (c), and bottom (d) boundaries of the object's bounding box.

as

$$p_{region}(\mathbf{z}_t \mid \mathbf{x}_t) \propto \exp(-\lambda(1 - \frac{1}{w_t h_t}\sum_{\mathbf{u}\in R(\mathbf{x}_t)} W_t(\mathbf{u}))^2), \qquad (3)$$

where the parameter λ is actually the reciprocal of the variance of the Gaussian function. We found that the tracking algorithm can work well when the parameter λ takes value in a certain range. Lacking a proper way to estimate satisfactorily the parameter λ, we empirically set it to be the same value $\lambda = 5$ in all experiments reported in this paper. From (3), we can see that the larger the average pixel value within the region, the more likely the region corresponds to the object's region. However, this measure is insensitive to the object's scale change; therefore, boundary cue needs to be considered.

As for boundary cue, based on the observation that pixel values on the probability map change sharply across the object's boundaries, we introduce four kinds of correlation templates C_L and C_R with the size of $3\times h_t$, C_T and C_B with the size of $w_t \times 3$ for the left, right, top, and bottom boundaries, respectively. The templates are shown in Figure 1. We measure the correlation values of these templates on the four boundaries of the region, respectively. The correlation operation is defined as

$$C_I \otimes W_t(x,y) = \sum_{i=-w'/2}^{w'/2} \sum_{j=-h'/2}^{h'/2} C_I(i,j) W_t(x+i, y+j), \qquad (4)$$

where $\otimes$ denotes the correlation operator, $I \in \{L,R,T,B\}$, w' and h' are the width and height of the correlation template, respectively. Then, the observation likelihood models for the four boundaries are defined as follows

$$p^L_{boundary}(\mathbf{z}_t \mid \mathbf{x}_t) \propto \exp(-\lambda(1 - g(\frac{1}{h_t} C_L \otimes W_t(x_L, y_t)))^2), \qquad (5)$$

$$p^R_{boundary}(\mathbf{z}_t \mid \mathbf{x}_t) \propto \exp(-\lambda(1 - g(\frac{1}{h_t} C_R \otimes W_t(x_R, y_t)))^2), \qquad (6)$$

$$p^T_{boundary}(\mathbf{z}_t \mid \mathbf{x}_t) \propto \exp(-\lambda(1 - g(\frac{1}{w_t} C_T \otimes W_t(x_t, y_T)))^2), \qquad (7)$$

sequence	number of frames	frame size
egtest01	1821	640x480
egtest02	1301	640x480
egtest03	2571	640x480
egtest04	1833	640x480
egtest05	1764	640x480
redteam	1918	352x240

Table 1. Details of the benchmark data set.

$$p^{B}_{boundary}(\mathbf{z}_t \mid \mathbf{x}_t) \propto \exp(-\lambda(1-g(\frac{1}{w_t}C_B \otimes W_t(x_t, y_B)))^2), \quad (8)$$

where $g(\bullet)$ is a normalization function, which normalizes the independent variable into the interval $[0,1]$, and is defined as

$$g(x) = \frac{x+2}{3}. \quad (9)$$

From (5)-(8), we can see that the larger the correlation values on the region's boundaries, the more likely the region corresponds to the object's region. Assuming that likelihoods of the four boundaries are independent of each other, the observation likelihood model for boundary cue is defined as

$$p_{boundary}(\mathbf{z}_t \mid \mathbf{x}_t) \propto \prod_{I \in \{L,R,T,B\}} p^{I}_{boundary}(\mathbf{z}_t \mid \mathbf{x}_t). \quad (10)$$

Furthermore, assuming that the likelihood of region cue is independent to that of boundary cue, the final observation likelihood model is thus defined as

$$p(\mathbf{z}_t \mid \mathbf{x}_t) \propto p_{region}(\mathbf{z}_t \mid \mathbf{x}_t) p_{boundary}(\mathbf{z}_t \mid \mathbf{x}_t). \quad (11)$$

The sum of pixel values within a region can be computed very quickly using the integral image data structure [25]. Pixel value on the integral image stores the sum of pixel values on the probability map above and to the left of that pixel's location. After one pass scan of the probability map, the integral image is constructed, and then only several additions are needed to compute the sum and correlation values. Therefore, more particles can be employed to obtain a proper representation of the posterior probability density function, without significantly decreasing the system's efficiency.

4. Experimental results

In this section, extensive experiments are carried out and reported to evaluate the proposed approach. We qualitatively evaluate the proposed tracker on several challenging image sequences, followed by quantitative comparison with several related trackers on the benchmark data set [26], which consists of several image sequences of moving vehicles captured from airborne sensor platforms. This data set has several typical challenges, e.g., cameras' zooming in and out, frequent and even abrupt motion change of sensor platforms, appearance changes of vehicles and surrounding background, partial and complete occlusions by trees, occlusions among vehicles, vehicles' moving in and out of trees' shadow, etc. Moreover, ground truth is provided every ten frames for each sequence, which includes accurate mask of the tracked vehicle. Details of the benchmark data set are given in Table 1.

4.1. Qualitative evaluation of the proposed tracker

In this subsection, we test the proposed tracker on several challenging image sequences, and make comparison with two related trackers. One is the tracker of Collins *et al.* [9], which is a representative of trackers performing mean shift tracking on the probability map, is implemented in the open source tracking test bed [26], and is denoted as variance ratio tracker in this paper. Variance ratio tracker selects the top 3 features with each feature producing a weight map, in which one mean shift process is employed to locate the object. Finally, the median value among the convergence locations of these mean shift processes is selected as the object's location. The other is color-based probabilistic tracker [16], which is a representative of probabilistic trackers, and is one of the pioneering works introducing color information into probabilistic tracking. Color information is modeled by histograms in HSV color space to alleviate the influence of illumination change. We implement color-based probabilistic tracker strictly following the description of the original algorithm, more details of which can be found in [16].

In the following experiments, images are down-sampled by a factor of two to eliminate noises and for computational efficiency, the number of particles is set to be $M = 400$ for the proposed tracker and color-based probabilistic tracker, and feature selection is performed every 10 frames for the proposed tracker and variance ratio tracker. It is still an open problem about how many particles should be used in particle filter. In general, the more the number of particles is, the better the system's state can be represented. However, too many particles will degrade the system's efficiency, which is an important factor for real time tracking. In the literature, the number of particles is generally set empirically [7], [15], [16], [17], [18], [19], [20], [22]. In this paper, the number of particles is also set empirically. And we found that setting $M = 400$ is a good trade-off between efficiency and the system's performance. All trackers are initialized with the same condition. And the experimental results are provided in Figures 2-5, where sub-figures (a), (b), and (c) show the results of variance ratio tracker, color-based probabilistic tracker, and the proposed tracker, respectively. For the convenience of observation, frame number is shown under each frame, and the successfully tracked regions are cropped, enlarged, and pasted on the top-left corner of each frame for Figures 2-4.

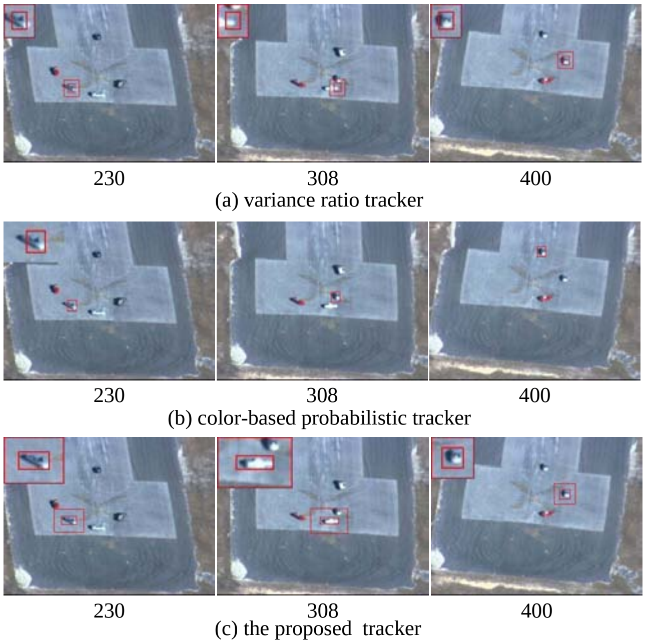

Figure 2. Tracking results of different trackers on *egtest01* sequence.

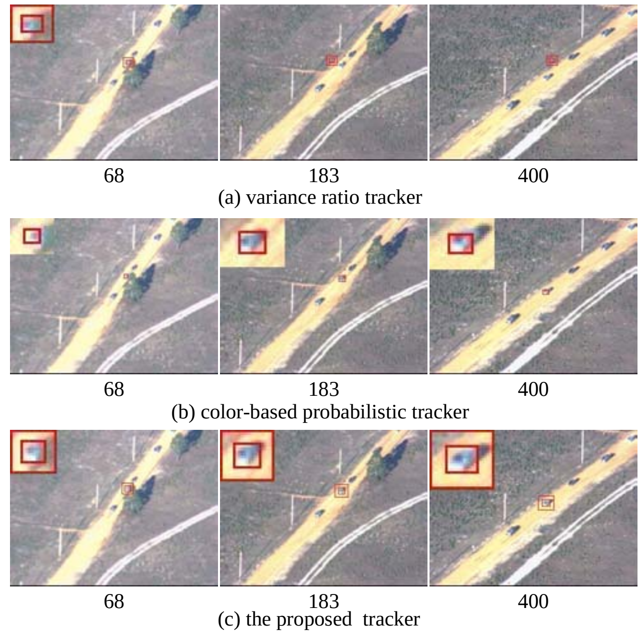

Figure 3. Tracking results of different trackers on *egtest04* sequence.

In Figure 2, the tracked vehicle loops around on a runway with its 3D pose and size changing remarkably. Moreover, around frame 308 the reflection of sunlight from the vehicle makes its appearance change drastically. Color-based probabilistic tracker drifts away from the vehicle in frame 308, locks on another vehicle nearby which more resembles to the color reference model of the tracker, and never recovers. This tracking failure is mainly because color-based probabilistic tracker has no built-in mechanism of appearance adaptation. When the appearance of the vehicle changes drastically, the tracker tends to drift. With online appearance adaptation, variance ratio tracker and the proposed tracker both successfully follow the vehicle even when its appearance changes drastically in frame 308. Variance ratio tracker emphasizes the front part of the vehicle which more resembles to the color reference model of the tracker than the center highlight part of the vehicle, whereas the proposed tracker more accurately obtains the location and size of the vehicle, which demonstrates the success of embedding the proposed observation model into particle filter.

In Figure 3, the tracked vehicle runs on a dirt road with short-term partial occlusion by trees. Sensor recording drops some frames, which is manifested in the sequence as duplicated frames followed by a sudden discontinuity. This results in an abrupt motion change. Variance ratio tracker drifts away from the vehicle as shown in frame 183 due to abrupt motion change, whereas color-based probabilistic tracker and the proposed tracker both successfully follow the vehicle, due to their capability of maintaining multiple

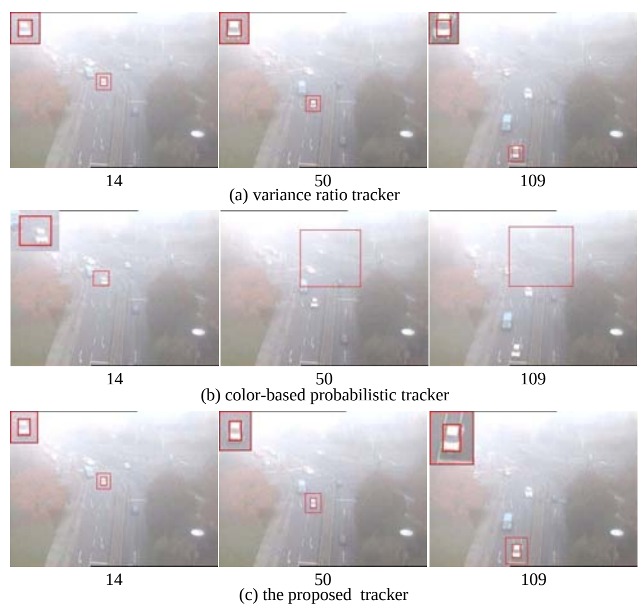

Figure 4. Tracking results of different trackers on *dtneu_nebel* sequence. (The sequence was downloaded from http://i21www.ira.uka.de/image_sequences/)

hypotheses of the vehicle's state. Compared with color-based probabilistic tracker, the proposed tracker follows the vehicle more accurately both in location and size as shown in frames 68 and 400.

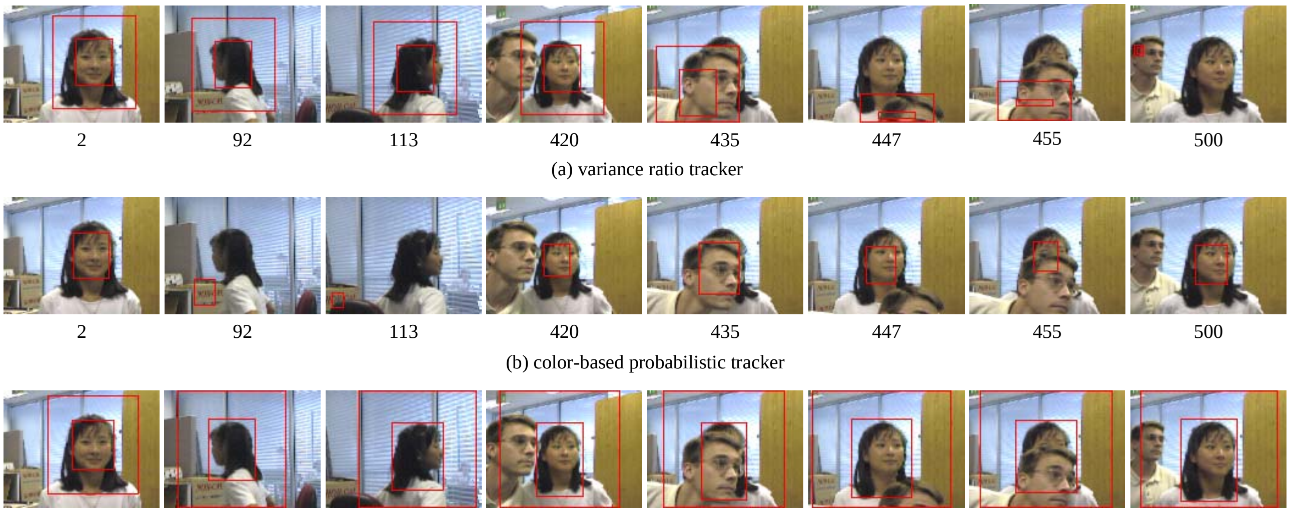

Figure 5. Tracking results of different trackers on *seq_mb* sequence. (The sequence was downloaded from http://www.ces.clemson.edu/~stb/research/headtracker/seq/)

Figure 4 shows a white vehicle being tracked in a traffic scene with heavy fog. White-like color resulting from the fog spreads over the scene, which poses great challenge to the trackers. Without taking background information into account, the window of color-based probabilistic tracker grows larger and larger, drifts away from the white vehicle, and finally locks on the white-like background. Through online selecting features that best discriminate the vehicle from white-like background, variance ratio tracker and the proposed tracker both successfully follow the vehicle until it disappears from the scene. Moreover, the proposed tracker captures the vehicle's size variation more accurately.

Figure 5 shows a more challenging 500-frame long sequence, in which a woman's head is tracked by the trackers. The challenges in this sequence include 360 degree out-of-plane rotation of the head, partial and complete occlusion of the woman's head by another man's head, etc. From Figure 5 we can see that the proposed tracker succeeds in the whole sequence, whereas variance ratio tracker and color-based probabilistic tracker both fail. Variance ratio tracker also succeeds when the woman turns her head around; however, it drifts on the occluding man's head and never recovers, when the man's head occludes the woman's head. When the woman turns her head around, color-based probabilistic tracker drifts away from the woman's head locking on the background region, which more resembles to the face's color than that of the hair. When the woman turns her head back, color-based probabilistic tracker occasionally captures the woman's face, and successfully tracks it through the following frames. The proposed tracker successfully tracks the woman's head through the whole sequence, which fully demonstrates the merits of embedding the proposed observation model into particle filter.

4.2. Quantitative comparison with the related trackers

Quantitative comparison with the related trackers is very important to show the effectiveness of the proposed tracker. Besides variance ratio tracker [9] and color-based probabilistic tracker [16], we also make comparison with Wang *et al*.'s tracker [20] (which is one of the related works that combine online feature selection with particle filter) on the benchmark data set with ground truth [26]. Two kinds of measure are employed to perform quantitative evaluation. One is the percentage of frames successfully tracked by a tracker, which is an important criterion to evaluate a tracker's performance. Furthermore, we provide another measure called tracking accuracy, which evaluates in each frame to what degree the region provided by a tracker coincides with the one provided by the ground truth. To achieve this aim, tracking accuracy measure [13] is adopted and is defined as follows

$$accuracy = \frac{2\,|R_{gt} \cap R_{tk}|}{|R_{gt}| + |R_{tk}|}, \tag{12}$$

where R_{gt} is the ground truth region of the tracked object, R_{tk} is the region provided by a tracker, $R_{gt} \cap R_{tk}$ denotes the intersection region between the ground truth region and the region provided by a tracker, and $|\bullet|$ denotes the area of the region. There exists an easily accessible interpretation of (12). Denoting precision as $P = |R_{gt} \cap R_{tk}| / |R_{tk}|$ and

sequence	variance ratio tracker		color-based probabilistic tracker		Wang *et al.*'s tracker		the proposed tracker	
	frames tracked (%)	average accuracy (%)	frames tracked (%)	average accuracy (%)	frames tracked (%)	average accuracy (%)	frames tracked (%)	average accuracy (%)
egtest01	**100.00**	68.32	16.48	65.03	14.84	72.53	**100.00**	**76.78**
egtest02	71.54	56.67	21.54	**65.24**	38.46	53.30	**100.00**	60.81
egtest03	15.95	77.16	**16.34**	65.08	11.67	**85.84**	**16.34**	77.39
egtest04	9.84	**84.61**	**40.44**	59.71	2.73	83.52	**40.44**	81.40
egtest05	**13.64**	82.01	**13.64**	71.15	**13.64**	**83.87**	**13.64**	80.56
redteam	98.43	65.60	**100.00**	51.90	48.69	69.10	**100.00**	**82.09**

Table 2. Quantitative comparison with the related trackers.

recall as $R = |R_{gt} \cap R_{tk}| / |R_{gt}|$, and then substituting these quantities into (12), we can obtain

$$accuracy = \frac{2PR}{P+R}, \quad (13)$$

which is just the definition of F_1 measure. F_1 measure is the harmonic mean of precision and recall, and is extensively used in pattern recognition community to evaluate a classifier's performance.

All the six sequences, details of which are provided in Table 1, are employed to quantitatively evaluate the performance of different trackers. The experimental results are shown in Table 2. The left column under each tracker's item shows the percentage of frames successfully tracked by the tracker, and the right column shows the average accuracy on the successfully tracked frames. A track is considered to be lost, if the ground truth region and the region provided by the tracker have no intersection. And the first such occurrence terminates the evaluation. For the purpose of easy comparison, we mark the best results in boldface font for each sequence as shown in Table 2.

Regarding the percentage of frames successfully tracked by a tracker, from Table 2 we can see that the proposed tracker achieves the best performance in all the six sequences, out of which three sequences are tracked completely. Whereas only one sequence is tracked completely for variance ratio tracker and color-based probabilistic tracker, respectively, and no sequence is tracked completely for Wang *et al.*'s tracker. The failure of Wang *et al.*'s tracker is mainly because the tracked objects do not have enough discriminative local features. Sequence egtest02 fully demonstrates the merits of embedding the proposed observation model into particle filter. In this sequence the tracked vehicle changes its appearance significantly, the airborne sensor platform shakes sharply, and the tracked vehicle passes several vehicles nearby with similar appearance. The other three trackers all fail in this sequence, whereas the proposed tracker succeeds. The other three sequences egtest03, egtest04, and egtest05 have much challenge. As for sequences egtest04 and egtest05, the tracked vehicles are completely occluded by trees for a period of time. Since no tracker has the mechanism of occlusion handling, they all fail in this case. For sequence egtest03 the road shows very similar color to the tracked vehicle when it turns its direction and runs across the road, and color features alone can hardly deal with this case.

As for average accuracy, from Table 2 we can see that the proposed tracker achieves the best performance in two sequences, and for the other four sequences the proposed tracker falls slightly behind the best ones. Whereas for variance ratio tracker and color-based probabilistic tracker the best performance is achieved in only one sequence, respectively, and for Wang *et al.*'s tracker the best performance is achieved in two sequences. Note that the average accuracy is computed for the successfully tracked frames; therefore, the percentage of frames successfully tracked by the tracker also needs to be considered when evaluating the tracker's performance. Sequence redteam fully demonstrates the effectiveness of the proposed tracker. In this sequence the camera zooms in and out twice, which results in significant scale change of the tracked vehicle. The proposed tracker not only completely tracks the vehicle in the whole sequence, but also achieves the highest average accuracy. From Table 2 we can also see that the average accuracy of the proposed tracker is about 10-30 percent higher than that of color-based probabilistic tracker, which further demonstrates the feasibility of the proposed observation model. The only exception occurs in egtest02 sequence, in which other vehicles with similar color passing the tracked vehicle make the tracker's window unstable, and hence decrease the overall average accuracy. However, color-based probabilistic tracker only tracks 21.54 percent of the sequence, whereas the proposed tracker tracks the whole sequence completely.

As for computational cost, on a PC with Pentium IV 3.2GHz CPU and 512M RAM, a non-optimized implementation of the proposed tracker in C runs comfortably in real time at 30-40 fps (frames per second) for 320x240 frame size, whereas color-based probabilistic tracker runs at 8-18 fps depending on the size of the tracked object. Most of the computational cost of color-based probabilistic tracker is spent in computing color histograms of 400 particles. For Wang *et al.*'s tracker, the running speed is 10-15 fps. The running time of these trackers includes the time cost by the main tracking algorithm, the time cost by reading image files, and the time cost by displaying images.

5. Conclusions

In this paper, we propose a novel observation model for probabilistic object tracking. The observation model is constructed on the probability map, and considers both region and boundary cues. Moreover, with the aid of the integral image data structure, the observation model can be computed very efficiently, and the proposed tracker runs comfortably in real time without any special optimization. Extensive experiments are carried out and reported on several challenging image sequences, followed by quantitative comparison with several related trackers on a public benchmark dataset. The experimental results demonstrate the effectiveness of the proposed approach.

Acknowledgements

This work was supported in part by National Basic Research Program of China (973 Program): 2009CB320906, and in part by National Natural Science Foundation of China: 60773136 and 60775024.

References

[1] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):747-757, 2000.

[2] G. R. Bradski. Real time face and object tracking as a component of a perceptual user interface. In *IEEE Workshop on Applications of Computer Vision*, pp. 214-219, 1998.

[3] S. Avidan. Support vector tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(8):1064-1072, 2004.

[4] N. P. Papanikolopoulos, P. K. Khosla, and T. Kanade. Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision. *IEEE Trans. on Robotics and Automation*, 9(1):14-35, 1993.

[5] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4): Article 13, 2006.

[6] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1296-1311, 2003.

[7] D. Ross, J. Lim, and M.-H. Yang. Adaptive probabilistic visual tracking with incremental subspace update. In *European Conference on Computer Vision*, pp. 470-482, 2004.

[8] B. Han and L. Davis. On-line density-based appearance modeling for object tracking. In *IEEE International Conference on Computer Vision*, pp. 1492-1499, 2005.

[9] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(10):1631-1643, 2005.

[10] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. on Information Theory*, 21(1):32-40, 1975.

[11] S. Avidan. Ensemble tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(2):261-271, 2007.

[12] H. Grabner and H. Bischof. On-line boosting and vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 260-267, 2006.

[13] H. T. Nguyen and A. W. M. Smeulders. Robust tracking using foreground-background texture discrimination. *International Journal of Computer Vision*, 69(3):277-293, 2006.

[14] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174-188, 2002.

[15] M. Isard and A. Blake. CONDENSATION: Conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5-28, 1998.

[16] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *European Conference on Computer Vision*, Copenhagen, Denmark, pp. 661-675, 2002.

[17] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 21(1):99-110, 2003.

[18] S. K. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Transactions on Image Processing*, 13(11):1491-1506, 2004.

[19] H.-T. Chen, T.-L. Liu, and C.-S. Fuh. Probabilistic tracking with adaptive feature selection. In *International Conference on Pattern Recognition*, pp. 736-739, 2004.

[20] J. Wang, X. Chen, and W. Gao. Online selecting discriminative tracking features using particle filter. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1037-1042, 2005.

[21] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade. Tracking in low frame rate video: A cascade particle filter with discriminative observers of different lifespans. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[22] B. Han and L. Davis. Robust observations for object tracking. In *IEEE International Conference on Image Processing*, pp. 442-445, 2005.

[23] D. Liang, Q. Huang, W. Gao, and H. Yao. Online selection of discriminative features using Bayes error rate for visual tracking. In *Pacific-Rim Conference on Multimedia*, 2006.

[24] D. Liang, Q. Huang, S. Jiang, H. Yao, and W. Gao. Mean-shift blob tracking with adaptive feature selection and scale adaptation. In *IEEE International Conference on Image Processing*, 2007.

[25] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 511-518, 2001.

[26] R. T. Collins, X. Zhou, and S. K. Teh. An open source tracking testbed and evaluation web site. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2005.