Robust model adaptation for tracking with online weighted color and shape feature

Jingjing Xiao and Mourad Oussalah
School of Electronics, Electrical and Computer Engineering, University of Birmingham, U.K.
e-mail: shine636363@sina.com, m.oussalah@bham.ac.uk

Abstract— Histogram-based particle filters have emerged as an appealing method for the target tracking. As colour and shape features are widely used to represent the target, we propose in this paper a novel method to combine these two features by assigning an adaptive weighting factor to each feature, in a particle filtering framework. In other words, the feature with higher likelihood and the property of high saliency will contribute more than other features to estimate the posterior density function of the target state in tracking. To cope with the target appearance changes, our tracker extracts the contextual information from the background to alleviate model drifting problem. The contextual information is therefore used for reference model update. We tested our proposed algorithm on some publicly available datasets, and the results from these video sequences have shown that the proposed tracker can tackle several open problems in tracking including heavy illumination changes, dramatic self-deformation and background clutter.

Keywords—Tracking, Background learning, online weighted features

I. INTRODUCTION

Thousands of deployed surveillance systems foster an increasing demand of robust automatic tracking systems [1]. This motivates the extensive passion among the researchers to enhance the effectiveness of automatic event recognition and target tracking from video sequences [2]. In this respect, two main streams of research can be distinguished. The first one relies on the use of visual features of the object as a main representation model, while the second espouses spatiotemporal prediction (search mechanism) like approach [3]. Many different cues can be utilized for target representation, i.e. color, shape, texture, silhouette, motion, etc. [4]. It is believed that the effective mathematical models for a good tracker should detect and combine target information from multiple features [5]. For spatiotemporal prediction, several approaches have been put forward (i.e. CamShift [6], Kalman Filter [7], Particle Filter [8] etc.). As Particle Filters have distinguished properties in dealing with multi-modal visual tracking problem of general non-linear and non-Gaussian systems without any assumption about the dynamics and shape of the conditional density, it has been used extensively in recent years. Nevertheless, the recourse to a single feature model tracking has often shown its limitation, which motivates the integration of more than one feature in the tracker model [9]. In this course, Kim et al. [10] recently proposed a non-rigid object tracker that combines shape and background information to enhance the tracking performance, which can account for object’s sudden movement or change of features, and whose background generation method performs well only in case of stationary camera. Erdem et al [11] utilized color, motion, infrared brightness cue for tracking. They combined multi-modal data by associating each particle with a specific cue and accordingly with a specific proposal function. A representative method was proposed by Collins et al [12] who utilized contextual information surrounding the target to assist tracking. The color feature was tuned to discriminate between object and (surrounding) background pixels by a “center-surround” approach. In the same spirit, Zhao et al [13] adaptively combined color feature and gradient feature for tracking by learning surrounding contextual information, as color histogram is robust to the shape deformation while histogram of gradient can cope with illumination changes.

In this paper, we first solve the problem of features fusion in the framework of particle filtering where two distinct features related to color and edges were employed, while employing contextual information to ensure robustness. Roughly speaking, the application of previously aforementioned works to extract background information for discriminative feature selection to Particle Filter framework is not straightforward. Indeed, if one extracts the surrounding contextual information for each particle, the method becomes very time consuming, and therefore, non-efficient. In our approach, to select the discriminative feature in Particle Filter framework, we use the standard deviation from particles’ observation as this has been found to better reflect the salient property of the features. This has therefore been employed to balance the contribution of the different features.

On the other hand, as the appearance of target may undergo in real scenarios self-deformation, viewpoint change and illumination variation, the appearance model also needs to be updated accordingly. For this purpose, Nummiaro et al. [8] suggested a linear combination between previous and current estimations of template model at every frame. Li and Chua [14] put forward a more sophisticated version of updating scheme that utilizes a decision of minimum error over the whole particle distribution. Work in [15] proposed a Rao-Blackwellised Particle Filter (RBPF) based tracking algorithm to handle the uncertainties caused by illumination changes and short-time occlusion. Especially, a joint image characteristic-space tracking scheme was introduced which updated the appearance model simultaneously to the object location estimation. Nevertheless all the above updating stages are solely based on the information extracted from the estimation where any inaccurate estimation is likely to bring serious drifting problem. Inspired by the work in [12] [13], we extracted the background information in the same spirit. Unlike the previously mentioned work, our approach does not use the background information to determine the weight associated to the various features (color, edges) but rather employed for updating the (target) reference models. In summary two main contributions have been put forward in this paper. First, adaptive fusion architecture has been proposed to combine the outputs of the color-based particle filter and edge-based particle filter outcomes. Second, contextual information is extracted for model update so that if the change of the appearance is originated from the background, the update should not be important.

The reminder of this paper is organized as follows. First, the basic framework of Particle Filter is introduced in Sec. II. Then, we propose our adaptive features fusion method in Sec. III. Contextual information based updating mechanism is designed in Sec. IV. Sec. V gives the details about the experiment while highlighting some of the results. A further analysis and conclusion is drawn in the Sec. VI.
II. PARTICLE FILTER FRAMEWORK

Particle Filter is acknowledged for its ability to solve the online estimation problems and is widely used in tracking problems [16]. The key idea of this method is to use a set of random samples (particles) to represent the posterior density function of the target, with associated weights. The overall estimation of the state $X$ of object from its particles is performed using the expectation operator over the set of particles:

$$X = \sum_{i=1}^{N} s^{(i)} \omega^{(i)}$$

where $s^{(i)}$ represents some hypothetical state of the object of interest, referred to $i$th particle, with associated weight $\omega^{(i)}$ such that $\sum_{i=1}^{N} \omega^{(i)} = 1$. In our algorithm, the target is modelled by a rectangular region where $s^{(i)}$ includes the parameters for describing the position, scale and velocity of the corresponding kernel. Namely,

$$s^{(i)} = [x\ y\ \hat{x}\ \hat{y}\ h\ w]^T$$

Where $x$, $y$, $\hat{x}$, $\hat{y}$ are the x-y coordinates and their associated derivatives of reference point of the target, $h$ and $w$ correspond to the height and width of the rectangular region centered at the above reference point, which constitutes the kernel (bounding box).

In the Particle Filter framework, there are three main steps: prediction, update and re-sampling.

In the prediction stage, the algorithm propagates the particles according to some prediction model; namely:

$$s_k = A s_{k-1} + v_k$$

Where $v_k \sim \mathcal{N}(0, R)$ is a zero mean Gaussian noise with variance-covariance matrix $R$. For the sake of simplification, we utilize a motion $A$ corresponding to a constant velocity, which yields a linear trajectory.

In the update stage, the measurement constituted of row image is used to modify the prior density by the principle of importance sampling, where the weight will be allocated to each particle according to the similarity between the reference target model and the candidate target model. In the same spirit as [8], we also utilize Bhattacharyya distance $D_B$ [17] to evaluate the similarity between the reference model and the observation. Then, this distance is calculated for each particle $s_k^{(i)}$, and used to compute the corresponding weights $\omega^{(i)}$ which are used for updating the state of the target as in Eq.1. In this step, we utilize both color (color histogram) and shape (histogram of gradient) features as visual models. As one feature might drop into a poor performance as compared to other, an adaptive fusion method is designed to dynamically assign weights to different features. Once an estimation of the bounding box of the underlying tracker is determined, we utilize contextual information to update the reference models as will be detailed in later sections. On the other hand, a resampling stage is performed to discard particles with small weights that cause a degeneracy problem. This step also helps in reducing the computational complexity caused by less-weighted particles [18]. In our algorithm, we employ a widely used systematic resampling method [19] to perform this task.

III. ONLINE WEIGHTED FEATURES

Color histogram is acknowledged for its effectiveness and robustness to the scale variation, rotation and small deformations, and widely used for target representation. However, this feature is subject to background clutter and illumination changes [13]. For this purpose, the histogram of oriented gradient [20], which provides distribution of gradient intensity with respect to various orientations, is suggested as a complementary feature. These two features were used to evaluate the corresponding weights of particles in the update stage.

A. Color-based histogram

In our color model, let us denote the pixel locations by $\{u'_i\}_{i=1}^M$ in the corresponding kernel (note that $u'_i$ is a 2-D dimensional) where $M$ stands for the total number of pixels in kernel. Consider the function $h: \mathbb{R}^2 \rightarrow \{1,2,...,M\}$

$$u'_i \mapsto h(u'_i)$$

which associates to a pixel $u'_i$ the index $h(u'_i)$ of its bins in the quantized color histogram from RGB color space, according to its grey level value.

The probability of the (color) bins $k$ ($k = 1$ to $m$, $m$ equals to 512 in our tracker) in the target model (RGB color space) is then computed in the same spirit as in [8].

$$H_{color}(k) = \left\{ \frac{1}{M} \sum_{i \in I} \delta[h(u'_i) - k] \right\}_{k=1..m}$$

(3)

Where $\delta$ is the Dirac delta function while $R$ represents the region of the kernel. Notice that $H_{color}$ is normalized as:

$$\sum_{k=1}^{m} H_{color}(k) = 1.$$

B. Histogram of gradient

The histogram of gradient (HOG) proposed by [20] is employed in our work. The essence consists of describing local object shape within an image by the distribution of gradient intensities with respect to edge directions.

The overall implementation is highlighted in Fig.1. Especially, the bounding box region is divided into small connected regions (cell) with the following three (sub) tasks:

- **Gradient computation** where a 1-D centered, point discrete derivative mask in both the horizontal and the vertical directions, is employed. Specifically, this method requires filtering the color or intensity data of the image with the following filter kernels:

$$[-1,0,1]$$ and $$[-1,0,1]^T$$

For each pixel, the norm and orientation is computed by:

$$\text{norm}(x,y) = \sqrt{px^2(x,y) + py^2(x,y)}$$

$$\text{orient}(x,y) = \arctan(py(x,y)/px(x,y))$$

Where $px(x,y)$ and $py(x,y)$ represent the horizontal and vertical gradient values, respectively.

- **Orientation binning** where each pixel within the cell casts a weighted vote for an orientation-based histogram channel (9 bins histogram per cell in our tracker) according to gradient values. Especially, the histogram of bin $\theta$ in the cell can be computed by:

$$h_{cell}(\theta) = \sum_{j=1}^{N} \text{norm}(x_p, y_p) \delta[\text{orient}^j(x_p, y_p) - \theta]$$

(6)
Where δ is the Dirac delta function, and \( \text{orient}'(x_p,y_p) \) is quantized orientation, computed from \( \text{orient}(x,y) \). \( N_p \) is the number of pixels in each cell. We represent the set of sums of magnitude in gradient θ for each cell as \( N \)-orientation histogram by \( H_{\text{cell}} = \{ h_{\text{cell}}(1), h_{\text{cell}}(2), \ldots, h_{\text{cell}}(N) \} \).

- **Descriptor blocks.** R-HOG blocks are generally square grids, represented by three parameters: the number of cells per block (9 cells), the number of pixels per cell (decided by the scale of the kernel of the tracker), and the number of channels per cell histogram (9 bins). Like [21], the histogram channels are calculated over rectangular cells (R-HOG) by the computation of unsigned gradient. In our HOG feature, the nine histograms (cells’ histogram \( \{ H_{\text{cell}}^{i} \}_{i=1..9} \) with nine bins (\( N = 9 \)) were then concatenated to make a 81-dimensional feature vector \( H_{\text{hog}} \). The cells have half overlap of their area. In other words, each cell is computed more than once to form the final histogram. To cope with the illumination and contrast changes, the gradient values of each cell are locally normalized, according to the gradient L2-norm:

\[
 h_{\text{hog}}'(n) = \frac{h_{\text{hog}}(n)}{\sqrt{\sum_{k=1}^{9} h_{\text{hog}}(k)^2} + \epsilon} \quad (7)
\]

There are \( q \times q \) cells (\( q = 3 \) in our tracker) in each block region, and regulation parameter \( \epsilon = 0.01 \). After normalization, the histogram of the whole bounding box \( H_{\text{hog}} \) becomes \( H_{\text{hog}} = \{ h_{\text{hog}}(1), h_{\text{hog}}(2), \ldots, h_{\text{hog}}(B \times N) \} \). Here, \( B \) is the number of cell regions (\( q \times q \) cells) that are contained in the block region.

C. **Online weight features.**

After the features extraction, each particle has two feature vectors (color, shape). Then, these observed feature vectors are compared to their corresponding feature models using Bhattacharya distance [17] yielding \( D_{\text{color}} \) and \( D_{\text{hog}} \) for color and shape feature, respectively. Finally, the Bhattacharya distances are used to compute the weights associated to the underlying particles; namely, for the \( i \)th particle, a Gaussian with standard deviation \( \sigma_{\text{color}} \) (or \( \sigma_{\text{hog}} \)), which is part of design parameter, and mean related to such distance is employed:

\[
 \omega_{\text{color}}^{(i)} \propto N(D_{\text{color}}, 0, \sigma_{\text{color}}^2) = \frac{1}{\sqrt{2\pi\sigma_{\text{color}}}} \exp\left\{ \frac{D_{\text{color}}^2}{2\sigma_{\text{color}}^2} \right\} \quad (8)
\]

And

\[
 \omega_{\text{hog}}^{(i)} \propto N(D_{\text{hog}}, 0, \sigma_{\text{hog}}^2) = \frac{1}{\sqrt{2\pi\sigma_{\text{hog}}}} \exp\left\{ \frac{D_{\text{hog}}^2}{2\sigma_{\text{hog}}^2} \right\} \quad (9)
\]

Notice that both \( \omega_{\text{color}}^{(i)} \) and \( \omega_{\text{hog}}^{(i)} \) fulfill the normalization condition; namely, \( \sum_{i=1}^{N} \omega_{\text{color}}^{(i)} = 1 \), \( \sum_{i=1}^{N} \omega_{\text{hog}}^{(i)} = 1 \).

A linear combination of the weights associated to color and shape features to output the overall weight as:

\[
 \omega^{(i)} = \mu_d \omega_{\text{color}}^{(i)} + (1 - \mu_d) \omega_{\text{hog}}^{(i)} \quad (10)
\]

Where the weighting factor \( \mu_d \) takes values in unit interval.

Noticeably, some features perform differently in various conditions; namely, the target may endure a dramatic color change, i.e. illumination changes where shape may provide good hints. On the other hand, target self-deformation / angle’s view change may degrade the shape feature solely results, suggesting color outperformance. Besides, since the illumination and self-deformation can occur at any time during the video clip, it is important to have a non-static weighting factor \( \mu_d \).

To tune the weighting factor \( \mu_d \) during tracking, we should design a performance metric that quantifies the performance of each feature. Intuitively, a good feature should have a high likelihood value as translated by a larger Bhattacharyya distance. Additionally, the weight of the particles belonging to the object should be quite discriminative, e.g., being larger than the particles distributed among background. At a given frame, a properly performing particle filter exhibits particles which are distributed around the target region but also in background region. This would suggest the (normalized) standard deviation computed from the particle weight values, namely, \( \omega_{\text{color}} \) (or \( \omega_{\text{hog}} \)) for color (HOG) features, would provide useful information regarding the quality of the global estimation. Typically, if the particle weight is too small, then it may indicate that this coincides with background. Therefore, a large discrepancy among particle weights, translated into a larger standard deviation would indicate a good discrimination between foreground and background region. For instance, if \( \sigma_{\text{color}} \) is quite large compared to \( \sigma_{\text{hog}} \), it would mean the color feature is more discriminative than HOG feature. Otherwise, the feature may indicate some clutter as the weights of particles would be close to each other. Therefore, we utilize four parameters to jointly decide the value of the weighting factor \( \mu_d \); namely, the standard deviations \( \sigma_{\text{color}} \) and \( \sigma_{\text{hog}} \) and \( D_{\text{color}} \) and \( D_{\text{hog}} \), which correspond to the maximum value of the particles’ Bhattacharya distance, in case of color and HOG features, respectively. In agreement with expression (10), one suggests \( \mu_d \), which quantifies the contribution of one feature with respect to another one as:

\[
 \mu_d = \frac{\sigma_{\text{color}} D_{\text{color}}}{\sigma_{\text{color}} D_{\text{color}} + \sigma_{\text{hog}} D_{\text{hog}}} \quad (11)
\]

IV ROBUST MODEL ADAPTATION

It is crucial for the tracker to update the visual model adaptively with appearance changes. However, the updating mechanisms which are solely based on the information extracted from the estimation are likely to result in serious drifting problems when inaccurate estimation happens. Naturally, the best way to avoid this drifting problem for appearance adaption is to extract contextual information from the background. Inspired by the work in [12] [13], even though the surrounding information of estimation is not straightforwardly suitable for particle filter like approach, it can be useful for updating the reference model. The methodology adopted in this paper is continuously impact the target update model by the extent of the change occurring in color (shape) histogram (s) of target kernel with respect to that of background region. The method is conducted in two steps: background extraction and weight determination.
A. background information extraction

We re-learn the contextual information by enlarging the bounding box as shown in Fig.3. A local background region is defined by delimiting a border strip that surrounds the foreground region.

![Background and foreground regions](image)

Figure. 3 Foreground and background regions

More specifically, we enlarge the bounding box of the estimation by a (multiplicative) scaling factor \( \tau \) at each edge, so that the resolution of the expanded kernel region:

\[
A_{f+b} = \tau^2 A_f = \tau^2 H_x^x H_y^y
\]

Where \( H_x^x \) and \( H_y^y \) are the lengths of the bounding box in \( x \) and \( y \) axis, respectively, while \( A_f \) is the resolution of the kernel associated to the foreground. Then, similarly to expression (3) we can get the appearance model (histogram) \( \hat{h}_{f+b} \) associated to the enlarged bounding box, which contains both foreground and background information. Then, the histogram of background can be approximated by (after normalization by \( A_{f+b} \) - \( A_f \)):

\[
h_b = \frac{A_{f+b} \hat{h}_{f+b} - A_f \hat{h}_f}{A_{f+b} - A_f}
\]

B. Online update model

After identifying the appearance model of the background according to (13), the similarities between background and current foreground appearance model (histogram) can be evaluated at each bin \( u \) as:

\[
c^u = \begin{cases} 
1 - e^{-\lambda_c \frac{\hat{h}_f(u)}{\hat{h}_b(u)}} & \hat{h}_b(u) \neq 0 \\
1 & \hat{h}_b(u) = 0
\end{cases}
\]

Where \( \lambda_c \) is some regulation parameter (which is set here to 0.01). \( \hat{h}_f(u) \) and \( \hat{h}_b(u) \) evaluate the histogram of the current foreground and background pixels at \( u \), respectively. The similarity measure \( c^u \), which ranges in unit interval, is then used to adaptively update the target reference model:

\[
\hat{h}_{\text{ref}}(u) = (1 - c^u) h_{\text{ref}}(u) + c^u h_f(u)
\]

Where \( h_{\text{ref}} \) represents the reference appearance model. After such change, a normalization stage is also carried out to bring the outcomes of (15) in unit interval. The overall scheme of model update is summarized in Fig.4.

![Model update diagram](image)

Figure 4 Diagram of model update

V. EXPERIMENT AND RESULTS

In order to quantify the performance of our tracking system, we used the challenging video sequences Gymnastics and David from the publically available dataset [22]. The sequence involves heavy illumination changes and environment clutter. Two evaluation metrics were employed for evaluation purpose: accuracy and robustness. The former is defined as the overlap between ground truth \( GT \) and tracker truth \( TT \) (estimated bounding box region):

\[
A = \frac{\text{GT} \cap \text{TT}}{\text{GT} \cup \text{TT}}
\]

The robustness is defined as the number of times that the tracker loses the target with regards to the accuracy, i.e., ratio of number of frame whose accuracy is below some threshold value over the total number of frames. Therefore, the trade-off curve of robustness with respect to accuracy can be drawn to show the overall performance of the tracker.

A. Sequences of gymnastics

In this sequence the object undergoes a dramatic self-deformation in clutter environment. As shown in Fig.5, after an initial run, the athlete (object to be tracked) stands almost immobile (frames 5-95) so that no strict preference between the two features can be noticed, yielding a weighting factor close to 0.5, reflecting this indifference in judgment. When the athlete begins to run, the shape feature becomes problematic while color feature provides better discrimination power, which should also be reflected in weighting factor choice (frames 95-110). While the athlete keeps running (frame 110-140), the background influence becomes important (clutter), as red background is almost similar to athlete suite, but the shape of the athlete remains almost unchanged. This yields a weighting factor that favors the shape feature. As the athlete did air tumbling from 165-190, the shape feature becomes extremely weak for some distinguished frames, which, in turn, yields peak values for weighting factor as can be seen in Fig. 5.

![Variation of weighting factor in Gymnastic sequences](image)

Figure 5 Variation of weighting factor in Gymnastic sequences

In Fig. 6, we compared our proposed adaptive combined-feature tracker with static weighting factor tracker (\( \mu_s = 0.5 \)) as well as with individual color / shape feature only tracker using robustness/accuracy performance metric.

![Robustness-accuracy curves of Gymnastics sequence](image)

Figure 6 Robustness-accuracy curves of Gymnastics sequence

From Fig.6, one notices that the dynamically combined feature tracker outperforms both any of individual single feature trackers and combined tracker with static weighting factor. It also shows...
that color feature in Gymnastics sequences outperforms the shape feature (HOG).

To demonstrate the efficiency of our updating method, we compare our proposed tracker to the one without updating stage, and to the updating scheme proposed in [15] in Fig.7 which exhibits large deformations of athlete shape.

\[ \text{Figure 7. Tracking experiment of appearance adaption (Self-deformation): first raw: no updating; second raw: updating using method of [15]; third raw: our method (frame: 1, 90, 150, 180)} \]

The plot shows the robustness of our approach to track the object even in its high deformations, while updating using [15] may result in even worse result than when updating is dropped off.

### B. Sequences of David

Typically, David’s sequence clip undergoes heavy illumination changes, which are likely to challenge any color feature-based reasoning. Nevertheless, unlike Gymnastic sequences where the clutter around the object of interest is important, the color of the face is quite distinguished from that of the background. Therefore, it is hard to make a clear judgment about a good feature even from an expert viewpoint. This is also reflected in result pointed out in Fig.8 where the weighting factor ranges quite close to 0.5 in most of the time.

\[ \text{Figure 8 Variation of weighting factor in David sequences} \]

Similarly to Gymnastic sequence, performances with respect to robustness/accuracy metric are exhibited in Fig.9. The plot shows again that adaptive feature combined tracker outperforms single feature trackers as well as static combined feature tracker. It also shows that the shape feature is generally better than the color feature.

\[ \text{Figure 9 Robustness-accuracy curves of David sequence} \]

Quantification of the importance of the model update is highlighted in Fig.10, where the importance of appearance model update is clearly stressed.

\[ \text{Figure 10. Tracking experiment of appearance adaption (Illumination change): first raw: no updating; second raw: [15] updating; third raw: our method (frame: 1, 50, 100, 150)} \]

Indeed the tracking improvements when using our contextual information based updating are made very clear. However, there is still a risk that the model drifts to the background when large inaccurate estimation occurs.

Finally, we also compared our tracker with another well distinguished state-of-the-art LGTracker [23] using other publicly available video sequences [22]. As the failure is proportionate to the defined overlap ratio, Table 1 summarizes the average accuracy of each video across all features for each video sequence (a higher average accuracy means a better tracking performance). We can see that the average accuracy of our tracker is systematically better than LGTracker in all employed videos.

<table>
<thead>
<tr>
<th>Name</th>
<th>Our tracker</th>
<th>LGT [23]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bolt</td>
<td>0.5649</td>
<td>0.414</td>
</tr>
<tr>
<td>Cup</td>
<td>0.6992</td>
<td>0.628</td>
</tr>
<tr>
<td>Face</td>
<td>0.6288</td>
<td>0.598</td>
</tr>
<tr>
<td>Jump</td>
<td>0.6982</td>
<td>0.588</td>
</tr>
</tbody>
</table>

VI. CONCLUSION

In this paper, we first propose a feature fusion method that can adaptively choose discriminative feature while tracking. Then, the model of the target is updated online by learning contextual information. The experiment carried out using publicly available dataset has shown the proposed tracker systematically outperforming single feature based tracker even under severe object or scene changes, i.e., heavy illumination changes, large self-deformation and background clutter.

ACKNOWLEDGMENT

Xiao is supported by the China Scholarship Council and school scholarship in the University of Birmingham. National Science Foundation of PR China under grant number 61171136 is also acknowledged for their financial support.

REFERENCES

surveillance system." 2nd International Conference in Information Technology, pp. 31-34, 2010.