This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Multiple Feature Fusion via Weighted Entropy for Visual Tracking

Lin Ma, Jiwen Lu, Jianjiang Feng and Jie Zhou
Department of Automation, Tsinghua University, China

Abstract

It is desirable to combine multiple feature descriptors to improve the visual tracking performance because different features can provide complementary information to describe objects of interest. However, how to effectively fuse multiple features remains a challenging problem in visual tracking, especially in a data-driven manner. In this paper, we propose a new data-adaptive visual tracking approach by using multiple feature fusion via weighted entropy. Unlike existing visual trackers which simply concatenate multiple feature vectors together for object representation, we employ the weighted entropy to evaluate the dissimilarity between the object state and the background state, and seek the optimal feature combination by minimizing the weighted entropy, so that more complementary information can be exploited for object representation. Experimental results demonstrate the effectiveness of our approach in tackling various challenges for visual object tracking.

1. Introduction

Object tracking is a fundamental researching topic in computer vision, and robust object tracking provides a further step to high-level visual analysis and understanding. In object tracking, the appearance model is an important factor for object representation, and a variety of feature descriptors with effective appearance models have been proposed in the literature [33]. Due to the computational convenience, single feature descriptor has been widely used in appearance based visual tracking models [4, 16, 28, 37]. However, single feature is usually not powerful enough to describe objects of interests and it is desirable to combine multiple feature descriptors to improve the visual tracking performance because different features can provide complementary information [24, 41].

Over the past few years, some researchers have proposed several multiple feature fusion based visual tracking methods, where different features were concatenated directly for object representation [22]. Kwon *et al.* [19] decomposed the observation model into multiple basic observation models, where each model was associated one feature. Then, they selected the optimal features to capture the appearance variations by using sparse principal component analysis. Grabner *et al.* [9] adopted multiple features and selected the best one according to the background information. However, both of them are not not effective to discriminate the object from background, especially when the background varies drastically.

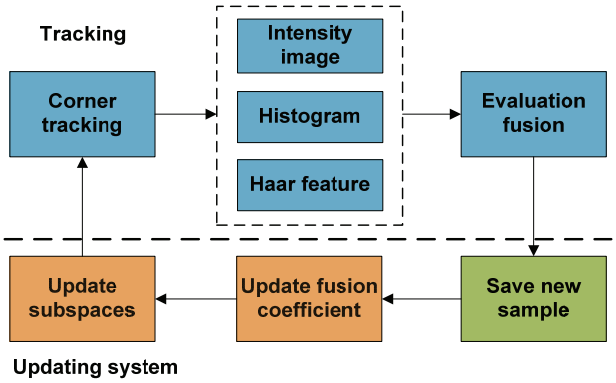

Figure 1. **Flowchart of our proposed approach.** Our approach contains two main parts: 1) tracking with complementary feature evaluation and 2) system updating. We first perform corner tracking and obtain an initial state, and then sample a set of candidate states around the initial state. Each state is first evaluated with three different features (intensity image, color histogram and Haar feature), and then they are evaluated and fused based on the weighted entropy. In the state updating procedure, we update the fusion configure by using saved samples with the weighted entropy, and also update the subspaces for different features separately.

To make the tracking model discriminative and robust to the drastic background variation, we propose a new data-adaptive visual tracking approach by using multiple feature fusion via weighted entropy, where both the generative and discriminative information are exploited in our approach. Figure 1 shows the flowchart of our approach. Since entropy can represent the disorderness of signals, it is much easier to separate signals by minimizing their entropy [10, 36]. To better highlight different contributions of different features in object representation, we employ the weighted entropy [11, 12, 18] to represent the difference or disorderness of the candidate states, and the most dis-

criminative feature fusion strategy can be obtained by minimizing the weighted entropy. Specifically, the states which are close to the optimal state evaluation are assigned larger weights, and the optimal state is well discriminated from similar candidate states.

2. Related Work

Appearance Models in Object Tracking: To obtain robust tracking performance, many appearance models have been proposed [7, 8, 14, 15, 17, 23, 27, 30]. These appearance models can be mainly classified into two categories: generative models and discriminative models. Generative methods learn an appearance model to represent the object and seek the optimal candidate which has the highest similarity with the templates, where the histogram representation [1, 4], subspace representation[22, 28, 32], and sparse representation [3, 16, 31, 34] are wisely used to learn the model. Histogram tackles the pose variation robustly, but it is easily influenced by camouflage background due to the lacking of spatial information [4, 5]. PCA (Principal Component Analysis) subspace[21, 28] preserves the spatial information and can well represent the distributions of objects. However, when computing the PCA subspace the information of all object samples is used, which makes the model ineffective to represent the local sample distribution. Sparse models [25, 26, 38] represent the object with only a few samples, which is robust to occlusion. But, the model fails to utilize the information provided by other samples, while other samples are useful to make the model robust to sample clutters. Generative models only consider the object appearances, and usually suffer from the large drifting problem. Discriminative models can address this shortcoming by representing the contrast between the object appearance and background, and many such models have been proposed in recent years [2, 20, 35, 37]. However, when the background varies drastically, the discriminative information is also inaccurate to robustly track the objects. Therefore, some researchers combine the generative and discriminative models together to represent the appearance of objects [39]. To better tackle the occlusion and pose variations, part-based models are utilized in the two kinds of models, i.e. generative model [1, 16] or discriminative model [29, 35]. In our work, we utilize the generative model to represent the object appearance, and discriminatively compute the optimal multiple feature fusion strategy by weighted entropy. Hence, our approach is also a hybrid appearance model for visual tracking.

Multiple Feature Fusion Based Visual Tracking: Many existing object tracking methods utilize single feature descriptor to represent the appearance of objects. However, each type of feature suffers from some limitations to handle the variations of objects. Therefore, it is desirable to combine multiple feature descriptors to improve the visual tracking performance because multiple features can provide complementary information. There have been some attempts on using multiple features for object tracking [24, 41]. For example, Li *et al.* [21] computed the covariance matrix from multiple features and combine them in the Riemann manifold. They concatenated multiple features to represent the object appearance and used no complementary information between features, while we obtained a better fusion result by using the complementary information. Kwon *et al.* [19] decomposed object tracking into multiple components, and combined multiple features for robust tracking, where, sparse principal component analysis was utilized to select the most important features to capture the appearance variations. No discriminative information was used in their method. However, by enlarging the difference between object sample evaluation and background sample evaluation, we separated the object from background more effectively. Grabner *et al.* [9] combined a set of weak classifiers to discriminate the object from the background, where multiple features were adopted in boosting to select the most discriminative information for the separation of objects from the background. In contrast, we use a generative model and exploit discriminative information in this model, so that it is more robust to background variations.

3. Feature Evaluation by Weighted Entropy

Let X_t denote the object state at frame t. The object state represents the position, height and width scales of object. At each frame, we first perform corner tracking and obtain an initial state. Then we sample a set of candidate states around the initial state according to a Gaussian distribution. Each candidate state is evaluated with the proposed feature evaluation method. The candidate state with best evaluation is selected as the optimal object state. By using multiple features, we can exploit complementary information to represent the object appearance. In this work, we utilize three widely used features: intensity image (raw pixel), color histogram feature, and Haar feature to represent the object appearance[1]. We first evaluate the candidate states with each feature separately, and then use the weighted entropy to obtain the best combination of these features.

To perform corner tracking, K base sub images (the representative image within the object bounding box) are firstly retained, where B_i is the ith subimage, $i = 1, 2, \cdots, K$. At frame t, the base subimage with the smallest Euclidean distance to object appearance at frame $t-1$ is selected as the current base subimage $\widetilde{B}$. We detect Harris corners on $\widetilde{B}$, and compute the translation of each corner separately. For corner P_i, we search the point P_i^c with the most similar tex-

[1]While three features are used in our approach, more other features can be easily incorporated into our tracking approach because our method is a general tracking framework and it is independent to the usage of individual features.

ture within a neighborhood, and then obtain the translation of P_i. Let E_T be the l_1 norm distance between two textures. Then the texture similarity is computed as $\exp(-E_T)$. The object translation is defined as the average of the translations of the corners where the weight of each corner is proportional to the texture similarity. The object size keeps invariant during the corner tracking. With the corner translation, we obtain the coarse object state. Then, we refine the state with feature evaluation.

3.1. Discriminative Feature Evaluation

We employ the weighted entropy to discriminatively combine multiple features for visual tracking. Let N be the number of candidate states in each frame, L^{inten}, L^{hist} and L^{haar} (defined with Eq. (10)) be the candidate state evaluation for the intensity image, color histogram, Haar feature at the tth frame, respectively. To uniform the evaluation of different features, we normalize the evaluation of each feature. We first remove the one which yields the smallest evaluation first, and then normalize the summation of the evaluation values of the remaining features to one. Let $\widetilde{L}_i^{inten}$, $\widetilde{L}_i^{hist}$ and $\widetilde{L}_i^{haar}$ be the normalized evaluation for the ith state, and the combined evaluation of the ith candidate state is defined as follows:

$$p_i = \alpha_0 \widetilde{L}_i^{inten} + \alpha_1 \widetilde{L}_i^{hist} + (1-\alpha_0-\alpha_1)\widetilde{L}_i^{haar}, \quad (1)$$

where $0 \le \alpha_0 \le 1, 0 \le \alpha_1 \le 1$ (also $0 \le \alpha_0+\alpha_1 \le 1$) are the coefficients of $\widetilde{L}^{inten}$ and $\widetilde{L}^{hist}$ respectively, which will be learned for multiple feature fusion. Having obtained α_0 and α_1, these features can be effectively combined.

Generally, the higher importance of the feature, the larger weight is assigned. A good multiple feature fusion strategy should make the combined evaluation discriminative. We employ the weighted entropy to measure the discriminative power of the fused state evaluation. Let $\theta = [\alpha_0, \alpha_1]^T$, the weighted entropy of the combined state evaluation is defined as

$$H(\theta) = -C(\theta) \sum\nolimits_i w_i p_i \lg p_i, \quad (2)$$

$$s.t. \quad \begin{array}{c} 0 \le \alpha_0 \le 1 \\ 0 \le \alpha_1 \le 1 \\ 0 \le \alpha_0+\alpha_1 \le 1 \end{array}$$

where $w_i = g(p_i)$ is the weight of the ith candidate state and $g(p_i)$ is a function of p_i. Here, $C(\theta) = 1/\sum_i w_i$ is a normalization term, and p_i is a function of θ. According to this fused state evaluation, we separate the optimal state from the most similar candidate states. Let $A_i = [\widetilde{L}_i^{inten} - \widetilde{L}_i^{haar}, \widetilde{L}_i^{hist} - \widetilde{L}_i^{haar}]^T$, and $g(p_i) = p_i^{\beta}$, β be a constant, (2) can be rewritten as

$$\begin{aligned} H(\theta) &= -C(\theta) \sum_i p_i{}^{1+\beta} \lg p_i \\ &= -C(\theta) \sum_i (\alpha_0 \widetilde{L}_i^{int} + \alpha_1 \widetilde{L}_i^{hist} + (1-\alpha_0-\alpha_1) \\ &\quad \widetilde{L}_i^{haar})^{1+\beta} \lg(\alpha_0 \widetilde{L}_i^{int} + \alpha_1 \widetilde{L}_i^{hist} + (1-\alpha_0 \\ &\quad -\alpha_1)\widetilde{L}_i^{haar}) \\ &= -C(\theta) \sum_i (A_i{}^T\theta + \widetilde{L}_i^{haar})^{1+\beta} \lg(A_i{}^T\theta + \widetilde{L}_i^{haar}). \end{aligned} \quad (3)$$

$$s.t. \quad \begin{array}{c} 0 \le \alpha_0 \le 1 \\ 0 \le \alpha_1 \le 1 \\ 0 \le \alpha_0+\alpha_1 \le 1 \end{array}$$

Smaller weighted entropy represents that the state evaluation is highly different from others, so that such a state evaluation is more discriminative. Thus, we obtain the best fusion coefficient by minimizing (3).

3.2. Optimization

We design an iterative method to minimize the objective function in (3). Specifically, we utilize the gradient descent method to seek the optimal fusion coefficient θ. As there are some constraints on θ, we need to carefully check this requirement of θ in each step. Since $\beta \neq 1$, $C(\theta)$ is a function of θ, we define $\beta = 1$ and obtain $C(\theta) = 1/\sum_i p_i = 1$ to reduce the computational complexity. To minimize the objective function in (3), we need to determine the searching direction. The optimal searching direction is determined by the gradient direction. Thus, we need to compute the gradient direction of the objective function in (3). According to (3), we obtain

$$\begin{aligned} \frac{\partial H(\theta)}{\alpha_0} = -\sum_i \left(A_i{}^T\theta + \widetilde{L}_i^{haar}\right)(\widetilde{L}_i^{inten} - \widetilde{L}_i^{haar})(1+ \\ 2\lg(A_i{}^T\theta + \widetilde{L}_i^{haar})), \end{aligned} \quad (4)$$

$$\begin{aligned} \frac{\partial H(\theta)}{\alpha_1} = -\sum_i \left(A_i{}^T\theta + \widetilde{L}_i^{haar}\right)(\widetilde{L}_i^{hist} - \widetilde{L}_i^{haar})(1+ \\ 2\lg(A_i{}^T\theta + \widetilde{L}_i^{haar})). \end{aligned} \quad (5)$$

Let $\Delta\theta = [\frac{\partial H(\theta)}{\alpha_0}, \frac{\partial H(\theta)}{\alpha_1}]^T$ and then the optimal searching direction is determined by $\Delta\theta$. Let a be the step length, then the new fusion coefficient at step k+1 is $\theta_{k+1} = \theta_k - a\Delta\theta$. We use the gradient descent to solve a step length a^0, where the optimal step length also needs to fulfill the positive constraint of θ in (3).

Figure 2 shows the conditions of the searching direction. When the current θ is on the boundary and the searching direction goes outward, we find a new searching direction (Figure 2(c)). Let the norm of the boundary is q ($||q||_2^2 = 1$), the new searching direction is set as:

$$\Delta\theta^{new} = \Delta\theta - (\Delta\theta)^T qq. \quad (6)$$

Based on (6), we make the new fusion coefficient stay within the acceptable range.

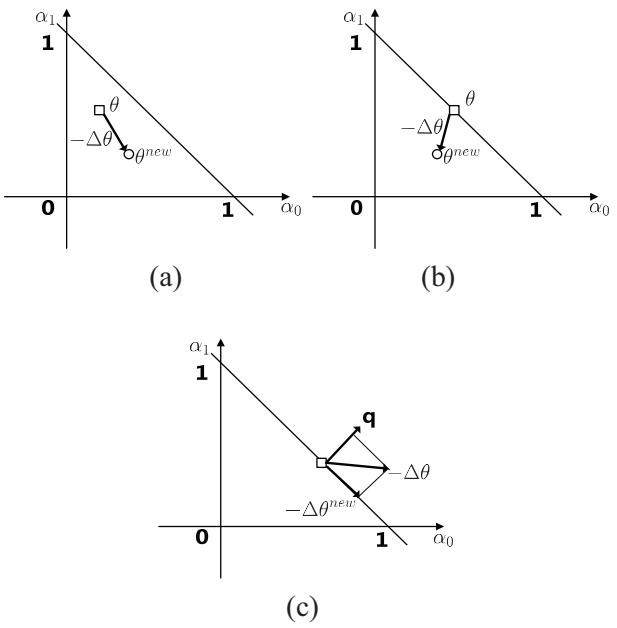

Figure 2. **Confining the searching direction of solving (3).** (a) When the original value is within the accepted range. (b) When the original value is on the boundary, and the direction points inward. (c) When the original value is on the boundary, and the direction points outward. The searching direction in (a) and (b) need not be changed. In (c), the direction is projected to an acceptable searching direction.

Let $[\Delta\alpha_0, \Delta\alpha_1]^T = \Delta\theta^{new}$. According to the constrains of the coefficients in (3), we obtain

$$\min\left(\frac{\alpha_0+\alpha_1}{\Delta\alpha_0+\Delta\alpha_1}, \frac{\alpha_0+\alpha_1-1}{\Delta\alpha_0+\Delta\alpha_1}\right) \leq a \leq \max\left(\frac{\alpha_0+\alpha_1}{\Delta\alpha_0+\Delta\alpha_1}, \frac{\alpha_0+\alpha_1-1}{\Delta\alpha_0+\Delta\alpha_1}\right) \quad (7)$$

$$\min\left(\frac{\alpha_1}{\Delta\alpha_1}, \frac{\alpha_1-1}{\Delta\alpha_1}\right) \leq a \leq \max\left(\frac{\alpha_1}{\Delta\alpha_1}, \frac{\alpha_1-1}{\Delta\alpha_1}\right). \quad (8)$$

For a concise presentation, we define $[\alpha_0, \alpha_1]^T = \theta_k$. According to the constrains in (3), we obtain $a^1 \leq a \leq a^2$. We define the final optimal step length a^* as $a^* = \min(a^0, a^2)$. If $a^0 \leq a^1$, the process ends. The process for minimizing the equation (3) is summarized as Algorithm 1. When an object sample is stored, we use the state evaluations at this frame to form Eq. (3) to obtain the optimal fusion coefficient. To make the θ retaining historical information, we set the new θ as

$$\theta^{new} = s\theta^{pre} + (1-s)\theta, \quad (9)$$

where $s \in (0, 1)$ is a scale to balance the importance of historical information and the new information.

Algorithm 1: Minimizing the weighted entropy objective function

Input: $\widetilde{L}_i^{inten}, \widetilde{L}_i^{hist}, \widetilde{L}_i^{haar}, i = 1, \cdots, N$.
Output: θ.

Compute $\Delta\theta = [\frac{\partial H(\theta)}{\alpha_0}, \frac{\partial H(\theta)}{\alpha_1}]^T$ with (4) and (5).
if *θ_k is on the boundary and the searching direction is outward* **then**
  Compute $\Delta\theta^{new}$ with (6).
end
Obtain a^0 with gradient descent.
Obtain a^1 and a^2 fulfilling the constraints in (3).
Obtain $a^* = \min(a^0, a^2)$.
if $a^0 \leq a^1$ **then**
  End.
end
Define $\theta_{k+1} = \theta_k - a^*\Delta\theta$.
if *achieving maximum iteration number (4 in this paper)* **then**
  Terminate.
end
else
  Back to Line 1.
end

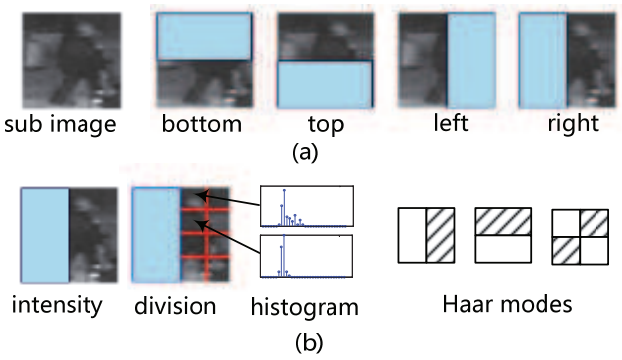

Figure 3. **The features.** (a) A subimage and corresponding four kinds of part sub images. (b) The three kinds of features, intensity image, color histogram and Haar feature. For Haar feature, we divide the part subimage into several patches, and for each patch we obtain three values by computing the convolution with the Haar modes (white areas are 1, the other areas are -1). The histogram feature is formed by concatenating the histograms of the patches.

4. Multiple Feature Fusion for Visual Tracking

In this section, we show how to evaluate the state for each individual feature and how to combine them for object tracking.

4.1. Forming Multiple Features

We utilize three features, i.e. intensity image, color histogram and Haar feature, to represent the object appearance (as shows in Figure 3). These features have complemen-

Algorithm 2: Our object tracking approach

Perform corner tracking and obtain an initial state.
Sample a set of candidate states, X_t^i, $i = 1, ..., N$ around the initial state.
for $i = 1, ..., N$ **do**
    Evaluate X_t^i according to intensity image.
    Evaluate X_t^i according to histogram.
    Evaluate X_t^i according to Haar feature.
    Obtain the fused evaluation with (1).
end
Obtain the optimal object state.
Update fusion coefficients whenever a sample is saved.
Update subspaces every M saved samples.
Update base sub images every M saved samples.

tary advantages. The intensity image [28] retains the original information of the object appearance. The histogram is robust to local distortion. Haar feature can represent the differences between neighbor object areas [2, 37], which is also robust to local variations.

To tackle occlusion problem, we divide the object subimage into four parts of sub images (Figure 3(a)), and perform state evaluation for each part of subimage. The evaluation of each feature is defined as the sum of the four part subimage's evaluations. When extracting Haar features, we divide the part subimage into a set of small patches, and obtain three values by computing the convolution with the three kinds of Haar modes (Figure 3(b)) for each patch. Then the Haar feature is represented by concatenating the values of the patches. The intensity image and color histogram features are also unfolded to vectors before evaluation.

4.2. Evaluating States with Incremental PCA

We use Principal Component Analysis (PCA) to evaluate each feature. Let U be the PCA subspace, F_t be the object feature in the tth frame, $\overline{F}_t$ be the mean of the historical object feature. Given the object observation O_t in the tth frame, the likelihood of state X_t is defined as

$$p(O_t|X_t) = \exp\left(-c\left\|(F_t - \overline{F}_t) - UU^T(F_t - \overline{F}_t)\right\|_2^2\right), \tag{10}$$

where c is a constant. To adapt to the object appearance variations, we update the subspace for each kind of feature every M saved frames. The subspace is obtained by performing singular value decomposition (SVD) on a matrix formed of previous information and new samples.

4.3. Updating Base Sub Images

To make the model adaptive to object appearance variation, we update the base sub images with new saved object samples. Let $\overline{E}_t$ be the average value of the objective function in (1) in terms of all frames from 1 to t. If the evaluation E_t of the optimal sample I_t is larger than a constant scale of $\overline{E}_t$, the object appearance normally has no large variation and then we consider I_t is not polluted (e.g. occluded), and save I_t. Otherwise, we drop I_t. To obtain better tracking accuracy, we update the base sub images by using every M saved object sub images rather then the polluted samples to update the subspace. Specifically, we form a set of $K + M$ candidate sub images with previous K base sub images and the new M saved sub images. Then we select K candidate sub images to form the new base sub images. For each candidate subimage, we compute the sum of the Euclidean distances between the current candidate subimage and other candidate sub images. The sub images corresponding to the largest K sums are selected. In this way, we make the base sub images better represent various object appearance forms. If the two sub images both are previous base sub images in the procedure of computing the distances between candidate sub images, the distance between them is multiplied by a constant s_1 ($s_1 = 1.5$ in this paper). When there are large background clutters, increasing s_1 increases the probability of selecting the old sub images, and then the model is able to avoid drifting due to incorrect new saved samples. In another part, when the object experiences pose variations, by decreasing s_1 the model is more likely to select new object samples to represent new object appearance forms. The process of our tracking system is summarized as Algorithm 2.

5. Experiments

In this section, we show the experimental results of our method. This section contains four parts. Firstly, we give the implementation details of our method. Secondly, we validate our multiple feature fusion based method with four methods which use single feature for state evaluation. Thirdly, we test the discriminating ability of the weighted entropy by comparing with the entropy based fusion method. Finally, we compare our method with nine state-of-the-art methods. Both quantitative and qualitative analysis are presented to show the effectiveness of our method.

5.1. Implementation Details

We test our approach on the 51 benchmark videos [33] which involves various challenges. The experiments are performed using C++ on a computer with 2.5G HZ CPU, 8G RAM. In each experiment, we warp the object subimage specified by the state to a normalized 32×32 subimage, and sample 200 particles per frame. For each candidate state, we divide the corresponding object subimage into four parts, i.e. left, right, top and bottom. Each part is further divided into 8 patches from which we form three kinds of features, i.e. intensity, color histogram and Haar feature. The bin

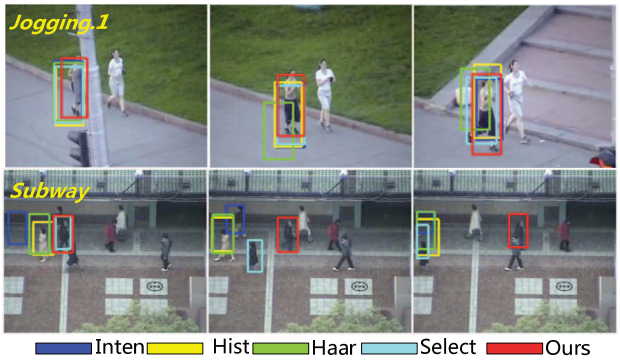

Figure 4. Some frames of the comparing methods and ours on two videos. The objects suffer from deformation or background clutter. With the weighted entropy based fusion strategy, our method obtains more discriminative state evaluation and performs more robustly than other four comparing methods.

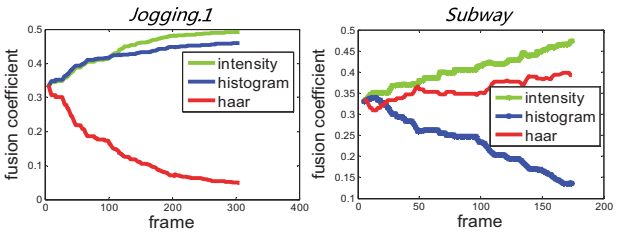

Figure 5. Fusion coefficients of each feature on two video sequences. Through weighted entropy, we obtain the optimal fusion coefficients for most discriminative state evaluation.

number of color histogram is set to be 32. The evaluations of the three kinds of features are fused to select the optimal state. We update the fusion coefficient whenever a new subimage is saved. We set $\theta = [0.33, 0.33]^T$ initially when updating the fusion coefficients, and set $s = 0.98$. When computing the reconstruction error, we set $c = 10^{-6}$ (the pixel value is not warped to $[0, 1]$) for intensity feature and Haar feature, and set $c = 1$ for the histogram feature. The subspaces are updated every 5 saved sub images. The run time of our method is around 0.4 sec/frame. We propose an efficient method to solve the objective function. The time cost of the system depends on feature choosing and feature evaluation method. Choosing faster feature evaluation method improves the processing speed. We compare our method with nine state-of-the-art methods: IVT [28], VTD [19], Frag [1], SLAM [21], ALSA [16], CT [37], Struck [13], SCM [40] and ColorT [6]. For the comparing methods, we utilize the source codes provided by the authors. We utilize the pixel center error and PASC overlap evaluation to evaluate the tracking performance. The precision plotted is plot over [0,50] with interval 10, and the success plot is plotted over [0,1] with interval 0.2.

5.2. Multiple Features vs. Single Feature

We compare our approach with single feature based trackers on ten video sequences. The ten video sequences

Table 1. AUC value of two criteria of *Intensity*, *Histogram*, *Haar*, *Select* and *Ours* on ten videos. *Select* represents the method selecting the most discriminative single feature according to weighted entropy to perform state evaluation.

	Intensity	Histogram	Haar	Select	Ours
Center	0.612	0.505	0.402	0.567	**0.823**
PASC	0.539	0.450	0.381	0.509	**0.698**

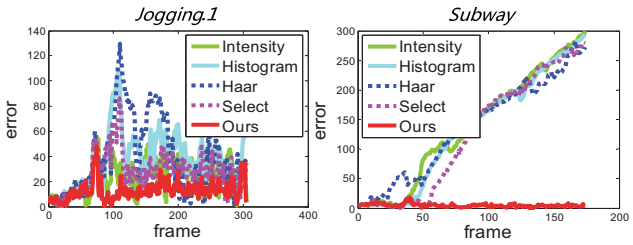

Figure 6. Pixel center error of our method and four single feature based methods at each frame on two video sequences. Our method tracks the objects more robustly than other four methods on the two videos.

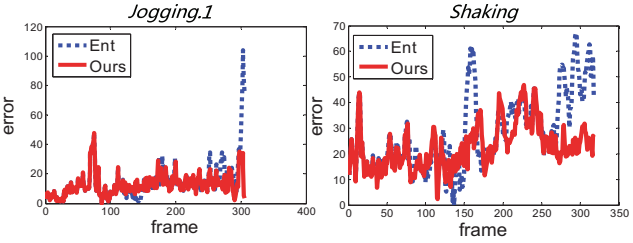

Figure 7. Pixel center error of Ent and our method at each frame on two video sequences. Our method represents the object states more accurately than Ent on the two videos.

Table 2. AUC values of two criteria of *Ent* and *Ours* on ten videos.

	Ent	Ours
Center	0.614	**0.823**
PASC	0.535	**0.698**

are *Basketball*, *Car4*, *Couple*, *FaceOcc1*, *FaceOcc2*, *Jogging.1*, *Jogging.2*, *Shaking*, *Singer2*, *Subway*. As shown in Table 1, our method discriminates the object more robustly than single feature based methods. Figures 4 and 5 show sample frames and the fusion coefficients for two video sequences, respectively. In *Jogging.1*, there is large object deformation. The haar features of local patches are disturbed, while the histogram feature tolerates with local distortion and is more useful to discriminate from background. With the weighted entropy, the histogram feature is given larger weight. In *Subway*, the background is similar to the object. The histogram feature is disturbed by the background and is given smaller weight. We see that our method is able to enlarge the weight of the feature which is more discriminative from background. Figure 6 shows the pixel center error of different methods on *Jogging.1* and *Subway*.

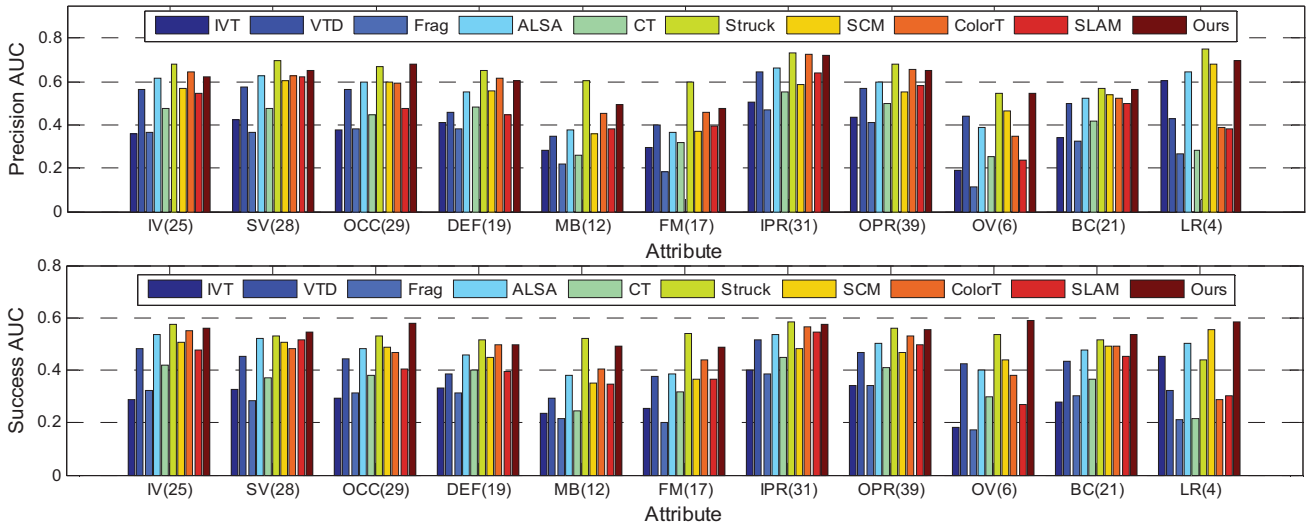

Figure 8. Precision AUC and success AUC of the 11 attributes on the 51 benchmark videos. Our method ranks in the top three on all the attributes according to the two criteria, and ranks the first on five attributes (SV, OCC, OV, BC and LR) according to success AUC. The number after each attribute is the corresponding video number.

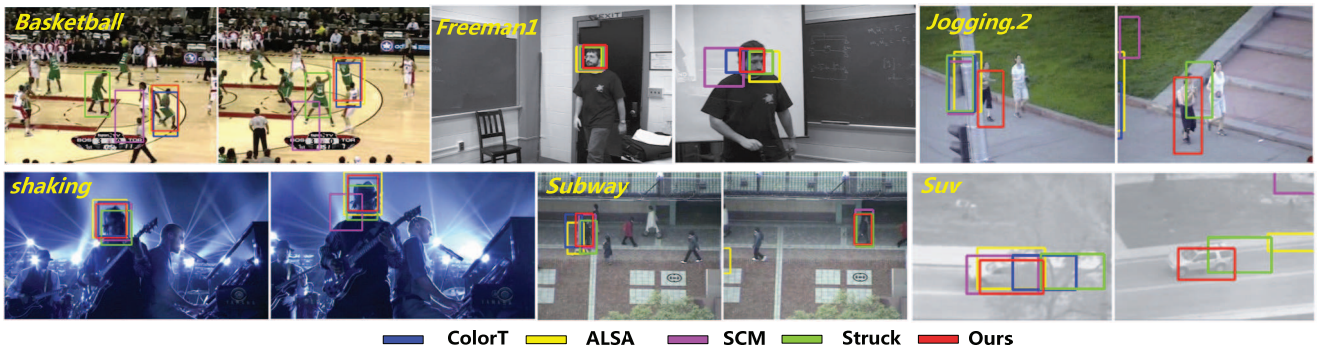

Figure 10. Sample frames of our method and four other methods, ColorT [6], ALSA [16], SCM [40] and Struck [13], on six videos. Various challenges [33], such as drastic deformation and severe occlusion, exist in these videos. Our method tackles these challenges effectively in comparison to the state-of-the-arts.

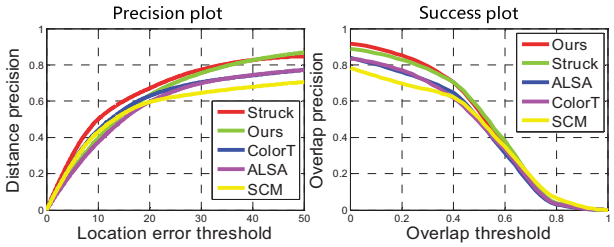

Figure 9. Precision and success plot of our method and four comparing methods, ColorT [6], ALSA [16], SCM [40] and Struck [13], on the 51 benchmark videos. The five methods are the top five according to Table 3. Our method ranks the second according to precision AUC, and ranks the first according to success AUC.

5.3. Weighted vs. Non-weighted Entropy

In this section, we compare our weighted entropy with entropy based fusion method (*Ent*) on ten video sequences (the same as those in Sec 5.2). By setting the weights of the candidate states according to the state evaluations, the states near the optimal state are more different from each other using weighted entropy. As shown in Table 2, our method obtains more accurate tracking performances than the entropy method. Figure 7 shows the pixel center error of *Ent* and our method at each frame on two video sequences.

5.4. Comparison with State-of-the-arts

Quantitative Analysis. We compare our method with nine state-of-the-art methods on the 51 benchmark video sequences. The AUC values of our method and comparing methods according to two kinds of criteria are shown in Table 3. From the table, we see that our method achieves best AUC values on overlap criterion, and obtains second best AUC value on the pixel center error criterion. We also show the performance of each comparing method on the 11

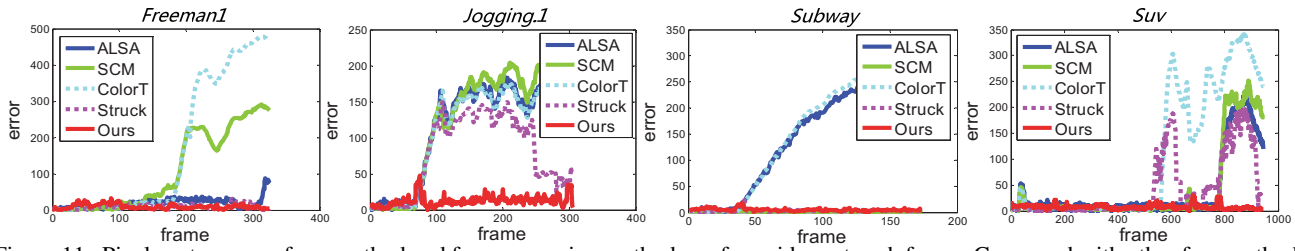

Figure 11. Pixel center error of our method and four comparing methods on four videos at each frame. Compared with other four methods, our method seldom loses tracks on the four videos.

Table 3. AUC value of pixel center error and PASC overlap of the comparing methods and our method. Red: best. Blue: second best.

	IVT	VTD	Frag	ALSA	CT	Struck	SCM	ColorT	SLAM	Ours
Center	0.458	0.577	0.440	0.636	0.492	0.723	0.610	0.657	0.586	0.694
PASC	0.357	0.459	0.354	0.517	0.400	0.567	0.503	0.517	0.486	0.571

attributes in Figure 8. The video attributes can be found in [33]. We see that on all attributes and both the two criteria, our method ranks among top three. Under the overlap criterion, our approach achieves the best on 5 attributes, and the second best on other 6 attributes. By utilizing the complementary information between various features, our method obtains better performance than SLAM [21] and VTD [19] which also uses multiple cues. Figure 9 shows the precision plot and success plot on the 51 benchmark videos. We see that our approach achieves very competitive performance with the state-of-the-art trackers.

Qualitative Analysis. Figure 10 shows some sample frames of our method and the other four comparing methods on six videos. Figure 11 shows the pixel center error of different methods on four videos. We divide the object subimage into four parts and perform PCA on each part subimage. When occlusion occurs, e.g. *Suv*, the not occluded parts still have large evaluation values, and when combining the evaluations of different parts together the object state also have large evaluation values. Thus the object state can be selected out robustly when occlusion occurs. Moreover, by introducing discrimination into the appearance model through weighed entropy, our method discriminates the object from background more effectively. On the contrary, other trackers, such as Struck [13] and ColorT [6], are influenced by severe occlusion, e.g. *Suv*, and fail to obtain robust performances.

When large pose variations occur, e.g. *Jogging.1*, many tackers are influenced and drift away. In contrast, the histogram feature tolerates with local distortion and is given large weight by weighted entropy as Figure 5 shows. And then, our method is able to tolerate with deformation.

Our method also tackles illumination variation effectively, e.g. *Basketball* and *Shaking*, by utilizing PCA appearance model. When the object experiences illumination variation, the object samples are distributed in linear subspace approximately. PCA is able to represent the linear subspace and thus our method is robust to illumination variation. When the object experience background clutter, our method finds the most discriminative feature evaluation through weighted entropy, e.g. *Subway*. And then our method can discriminate the object from background effectively. However, when the background clutter is severe, e.g. *Ironman*, our method is disturbed largely and fails to obtain robust tracking performance any more.

6. Conclusion and Future Work

In this paper, we have proposed a new multiple feature fusion based state evaluation method with weighted entropy. Through weighted entropy, we are able to utilize the complementary information provided by multiple cues, and make the candidate states around the optimal state more different from each other. Three different features and PCA subspace model are used in this paper, however other features and feature evaluation methods can be easily adopted in this strategy. In the future, we will continue the researches in the complementary information between various features and forming more discriminative features. Moreover, how to employ our feature fusion method to other vision applications such as image classification and action recognition is also interesting future topic.

Acknowledgement

This work is supported by the National Natural Science Foundation of China under Grants 61225008, 61572271, 61527808, 61373074 and 61373090, the National Basic Research Program of China under Grant 2014CB349304, the Ministry of Education of China under Grant 20120002110033, and the Tsinghua University Initiative Scientific Research Program.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. *CVPR*, 2006. 2, 6

[2] B. Babenko, M. Yang, and S. Belongie. Visual tracking with online multiple instance learning. *CVPR*, 2009. 2, 5

[3] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust l1 tracker using accelerated proximal gradient approach. *CVPR*, 2012. 2

[4] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Tech. J.*, 2, 1998. 1, 2

[5] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *PAMI*, 25(4):564–575, 2003. 2

[6] M. Danelljan, F. S. Khan, M. Felsberg, and J. Weijer. Adaptive color attributes for real-time visual tracking. *CVPR*, 2014. 6, 7, 8

[7] T. B. Dinh, N. Vo, and G. Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. *CVPR*, 2011. 2

[8] J. Gao, H. Ling, W. Hu, and J. Xing. Transfer learning based visual tracking with gaussian processes regression. *ECCV*, 2014. 2

[9] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via online boosting. *BMVC*, 2006. 1, 2

[10] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. *NIPS*, 2004. 1

[11] B. Guan, B. Bhanu, N. Thakoor, P. Talbot, and S. Lin. Automatic cell region detection by k-means with weighted entropy. *ISBI*, 2013. 1

[12] S. Guiasu. Weighted entropy. *Reports on Math Phys*, 2:165–l79, 1971. 1

[13] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured output tracking with kernels. *ICCV*, 2011. 6, 7, 8

[14] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 24(5):1–14, 2015. 2

[15] P. M. J. Henriques, R. Caseiro and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. *ECCV*, 2012. 2

[16] X. Jia, H. Lu, and M. H. Yang. Visual tracking via adaptive structural local sparse appearance model. *CVPR*, 2012. 1, 2, 6, 7

[17] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *PAMI*, 34(7):1409–1422, 2012. 2

[18] J. Khan and S. Bhuiyan. Weighted entropy for segmentation evaluation. *Optics and Laser Technology*, 57:236–242, 2014. 1

[19] J. Kwon and K. M. Lee. Visual tracking decomposition. *CVPR*, 2010. 1, 2, 6, 8

[20] H. Li, Y. Li, and F. Porikli. Deeptrack: Learning discriminative feature representations by convolutional neural networks for visual tracking. *BMVC*, 2014. 2

[21] X. Li, W. Hu, Z. Zhang, and X. Zhang. Robust visual tracking based on an effective appearance model. *ECCV*, 2008. 2, 6, 8

[22] X. Li, W. Hu, Z. Zhang, X. Zhang, and G. Luo. Visual tracking via incremental log-euclidean riemannian subspace learning. *CVPR*, 2008. 1, 2

[23] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Robust tracking using local sparse appearance model and k-selection. *CVPR*, 2011. 2

[24] X. Lu, H. Lei, and Z. Hao. Automatic camshift tracking algorithm based on multi-feature. *J. of Computer Application*, 30(3):650–652, 2010. 1, 2

[25] X. Mei and H. Ling. Robust visual tracking using l1 minimization. *ICCV*, 2009. 2

[26] X. Mei and H. Ling. Robust visual tracking and vehicle classification via sparse representation. *TPAMI*, 33(11):2259–2272, 2011. 2

[27] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan. Locally orderless tracking. *CVPR*, 2012. 2

[28] D. A. Ross, J. Lim, R. Lin, and M. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1):125–141, 2008. 1, 2, 5, 6

[29] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. *CVPR*, 2012. 2

[30] H. Z. Wang, D. Suter, K. Schindler, and C. H. Shen. Adaptive object tracking based on an effective appearance filter. *PAMI*, 29(9):1661–1667, 2007. 2

[31] N. Wang, J. Wang, and D. Yeung. Online robust non-negative dictionary learning for visual tracking. *ICCV*, 2013. 2

[32] T. Wang, I. Gu, and P. Shi. Object tracking using incremental 2d-pca learning and ml estimation. *ICASSP*, 2007. 2

[33] Y. Wu, J. Lim, and M. Yang. Online object tracking: A benchmark. *CVPR*, 2013. 1, 5, 7, 8

[34] Y. Wu, B. Ma, M. Yang, Y. Jia, and J. Zhang. Metric learning based structural appearance model for robust visual tracking. *TCSVT*, 24(5):865–877, 2014. 2

[35] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. Hengel. Part-based visual tracking with online latent structural learning. *CVPR*, 2013. 2

[36] J. Zhang, S. Ma, , and S. Sclaroff. Meem: Robust tracking via multiple experts using entropy minimization. *ECCV*, 2014. 1

[37] K. Zhang, L. Zhang, and M. Yang. Real-time compressive tracking. *ECCV*, 2012. 1, 2, 5, 6

[38] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via multi-task sparse learning. *CVPR*, 2012. 2

[39] X. Zhang, W. Hu, S. Maybank, and X. Li. Graph based discriminative learning for robust and efficient object tracking. *ICCV*, 2007. 2

[40] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. *CVPR*, 2012. 6, 7

[41] Z. Zhou, D. Wu, X. Peng, Z. Zhu, and K. Luo. Object tracking based on camshift with multi-feature fusion. *J. of Software*, 9(1):147–153, 2014. 1, 2