Object tracking with Multiple instance learning and Gaussian mixture model

Na Li[a,b*], Xiangmo Zhao[a], Daxiang Li[b], Jing Wang[c], Bendu Bai[b]

[a] *School of Information Engineering, Chang'an University, Xi'an 710064, China*

[b] *School of Communication and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China*

[C] *School of Computing and Engineering University of Huddersfield, Huddersfield HD1 3DH, United Kingdom*

[*] *Corresponding author.*

E-mail address: lina114@xupt.edu.cn (Na Li)

Tel: +86 029 88166381, Fax: +86 029 88166381, Mobile:+86 18591406236

Address:School of Communication and Information Engineering, Xi'an University of Posts and Telecommunications, Chang'an West Street, Chang'an District, Xi'an , China. Postcode:710121.

Abstract: Recently, Multiple instance learning (MIL) technique has been introduced for object tracking applications, which has shown its good performance to handle drifting problem. While some instances in positive bags not only contain objects, but also contain the background, it is not reliable to simply assume that each feature of instances in positive bags obeys a single Gaussian distribution. In this paper, a tracker based on online multiple instance boosting has been developed, which employs Gaussian mixture model (GMM) and single Gaussian distribution respectively to model features of instances in both positive and negative bags. The differences between samples and the model are integrated into the process of updating the parameters for GMM. With the Haar-like features extracted from the bags, a set of weak classifiers are trained to construct a strong classifier, which is used to track the object location at a new frame. And the classifier can be updated online frame by frame. Experimental results have shown that our tracker is more stable

and efficient when dealing with the illumination, rotation, pose and appearance changes.

Keywords: Object tracking, Multiple instance learning, Gaussian mixture model

1. Introduction

Object tracking has become a research hot spot in computer vision area, and has been widely used for video surveillance, video retrieval and behavior analysis et al. [1]. During the past decades, numerous algorithms for object tracking [2-7] have been proposed. But many negative impacts coming from illumination changes, appearance modifications, shape variations, and partial or full occlusions often seriously affect the performance of the algorithms. It is still a challenging task to track objects in complex scenes.

In general, the tracking methods can be categorized into two classes: generative methods[8,9] and discriminative methods [3,5,10]. Generative methods search for the most similar regions as the object appearance at each frame, based on learning an appearance model for object representation. In [11], an appearance model was learned offline, which could not adapt significant appearance changes. In order to solve this problem, some adaptive appearance models were proposed [4,8]. However, those generative methods do not take background information into account, which may be useful to discriminate the objects from their background. Recently, sparse representation [12] has been introduced into visual tracking. Among these generative appearance models based on sparse representation, tracking problems are formulated to attempt to jointly estimate the target appearance by finding a sparse linear combination over a dictionary containing the target and trivial templates [13,14].

Discriminative methods define the tracking problems as binary classification tasks, which attempt to design a classifier to separate the objects from their surrounding background. These

methods are also named as tracking by detection *i.e.* treat tracking as detection problem [15]. In [5], support vector tracking was proposed by integrating an offline afore-trained Support Vector Machine classifier into an optical-flow-based tracker. To adapt to object appearance changes, the discriminative models are updated in an online manner. Avidan [3] proposed an ensemble of online weak classifiers, which could label each pixel as the object or the background. Grabner et al. [10] proposed a semi-supervised approach for training the classifier by only labeling the samples at the first frame. However, some useful information has been lost for object tracking by using the method. For tackling those problems, Viola et al. [16] discussed the inherent ambiguities of object detection that caused drift for traditional supervised learning methods, and suggested using Multiple instance learning (MIL) [17] for object detection. This method has been approved as a valid approach for many visual tracking applications [18-21,23].

In the papers mentioned above [18-20,23], there is a common assumption that each Haar-like feature of instances in positive and negative bags obey different single Gaussian distributions. However, as illustrated in Fig. 1, it is obvious that some instances in positive bags not only contain the object, but also contain the background. It is not reasonable to assume that each feature of instances in positive bags can be described as a Single Gaussian Model (SGM). In this paper, we propose a robust object tracking method with MIL and Gaussian mixture model (GMM). The main contributions of this paper are:

- A robust object tracking method based on MIL is proposed, which employs GMM to describe features of instances in positive bags.
- New parameter update rules for GMM is proposed, which fully consider the differences between samples and the model.

- To demonstrate the promising performance of our method, we have extensively compared our method with other state-of-the-art trackers.

The rest of this paper is organized as follows: In Section 2, representative related work is discussed; Section 3 highlights the new object tracking method based on MIL and GMM; in Section 4, the detailed experimental results are given and we conclude the work in Section 5.

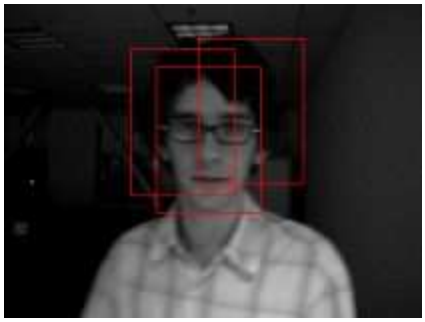


Fig. 1. Extracted instances in a positive bag

2. Related works

Recently, many researchers have concentrated on studying a class of tracking technique named as tracking by detection [5,15], which takes tracking as a detection task. It deploys machine learning algorithms to learn a discriminative classifier which separates objects from the background, and shows promising experimental results in real time. The adaptive tracking by detection methods [10,22] use samples extracted from the current frame for training an online classifier. Samples' coordinates of the next frame can be defined by the location information around the previous object. The new object location is then updated based on the maximum classification score from those samples.

The term MIL was proposed by Dietterich et al.[17] for investigating the problem of drug activity prediction. In the MIL problem, the training samples are regarded as bags containing many instances. Training labels are associated with bags rather than instances. A bag is labeled

positive if it contains at least one positive instance, otherwise it is labeled as a negative bag. The task is to learn some concept from the training set for correctly labeling unseen bags. MIL can handle ambiguities of the training data. During the last decade, numerous MIL algorithms have been proposed, such as axis parallel hyper-rectangles [17], Citation-kNN [24], Diverse Density (DD) [25], DD with Expectation Maximization (EM-DD) [26], Neural Network [27] and so on, which is widely used for drug activity prediction, image retrieval, intrusion detection and object tracking.

When using online MIL-based object tracking technique, an image can be represented as a set of image patches. A bag containing those image patches is labeled positive if at least one of its instances contains the object, while a bag is labeled negative if all of its instances only contain the background. In [18], a novel boosting-based algorithm for online MIL was proposed, which deployed MIL instead of traditional supervised learning to train a discriminative classifier, and it achieved superior results with real-time performance. Babenko et al. [19] proposed an improved online MIL tracking algorithm, fully considering scale tracking. And the results showed that on average, the tracker in the article was the most robust tracker with respect to partial occlusions and various appearance changes. Zhang et al. [20] proposed a novel weighted MIL (WMIL) tracker that integrated the sample importance into the learning procedure. The bag probability function combined weighted instance probability, and experimental results on challenging video sequences demonstrated the superior performance of the method in robustness, stability and efficiency to state-of-the-art methods in the literature. In [21], particle filter was applied to make best use of the learned classifier and help to generate a better representative set of training examples for the online MIL learning. The effectiveness of the proposed algorithm demonstrated in some

challenging environments for human tracking. Lu et al. [23] proposed a so-called co-training multiple instance learning algorithm, which labeled incoming data continuously, and then used the prediction from each classifier to enlarge the training set of the other. The discriminative classifier was implemented by using online MIL.

In this paper, a MIL tracker has been proposed that respectively employs GMM and SGM for modeling features of instances in both positive and negative bags. The differences between samples and the model are integrated into the process for renewing the GMM parameters. With the Haar-like features extracted from the bags, a set of weak classifiers are trained to construct a strong classifier, which is used to track the object location at a new frame. And the classifier can be updated online frame by frame.

3. Proposed algorithm

3.1 Algorithm overview

In MIL tracking, an instance is an image patch, while a bag is a set of image patches. A bag is positive if at least one of its instances contains the object, while a bag is negative if all of its instances only contain the background. The key problem of MIL-based tracking is how to train the classifier to separate the object from its surroundings, which can alleviate drift when illumination changes, appearance changes, rotation and partial or full occlusions occur.

The basic framework of our tracking method is illustrated in Fig. 2. The main steps are summarized as follows:

Step 1. Select the object for tracking with a rectangular bounding box, and initialize its parameters such as the radius of cropping positive and negative bags γ, ζ and β, the searching radius of the object s, the total number of weak classifiers M, the number of weak classifiers to

construct the strong classifier *K*, and the number of Gaussian distribution in GMM *C*.

Step 2. Extract positive and negative bags at the current frame. Crop two sets $X^{\gamma} = \{x : \| l(x) - l_t^* \| < \gamma\}$ and $X^{\zeta,\beta} = \{x : \zeta < \| l(x) - l_t^* \| < \beta\}$ around the object, and respectively tag them as positive and negative bags, where *x* is an image patch (i.e. an instance in a bag), *l(x)* is the location of image patch *x* represented by the *(x,y)* coordinates of the patch center, l_t^* is the most likely object location at the t^{th} frame, γ, ζ and β are scalar radius measured in pixels.

Step 3. Construct a strong classifier with online MIL. Firstly, extract Haar-like features for each instance in bags, and deploy GMM and SGM to model features of instances in positive and negative bags respectively (see Section 3.2). Secondly, train *M* weak classifiers and get the candidate weak classifier pool $\boldsymbol{\phi} = \{h_1, h_2, ..., h_M\}$. Each weak classifier is composed of a Haar-like feature f_k and the parameters of the feature distribution (see Section 3.3). Finally, the most discriminative *K* weak classifiers are chosen from the candidate pool $\boldsymbol{\phi}$ sequentially for composing the strong classifier $H_K(x) = \sum_{m=1}^{K} h_m(x)$ (see Section 3.4).

Step 4. Load the next frame, and track the object with the afore-trained strong classifier. Around the previous object location, crop a set of image patches $X^s = \{x : \| l(x) - l_t^* \| < s\}$, and compute their feature vectors, where s is a scalar radius for searching new location. And then the most likely object location can be renewed at the current frame denoted as l_{t+1}^* with the the afore-trained strong classifier H_K, where $l_{t+1}^* = l(\arg\max_{x \in X^s} p(y = 1 \mid x))$, and $p(y = 1 \mid x)$ represents the probability of the presence of the object in the image patch *x*.

Step 5. Repeat the steps 2 to 4 until the last frame.

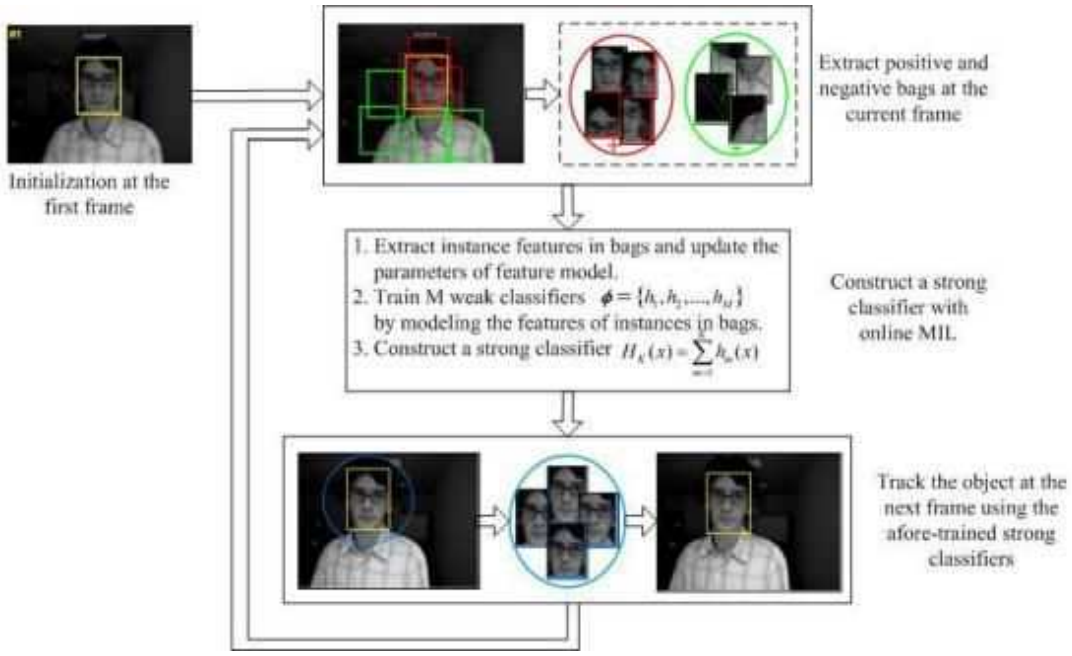


Fig. 2. The basic framework of our method

3.2 Modeling features of instances

3.2.1 Modeling features of instances in positive bags

Under the framework of MIL, although the size of the bags can vary, negative bags only consist of negative instances, whereas positive bags are comprised of both positive and negative instances. That is to say, some instances in positive bags not only contain object region, but also contain background region. So the features of instances in positive bags consist of not only background features but also object features. It is not appropriate to assume that each feature of instances in positive bags obeys a single Gaussian distribution. In this paper, GMM [28] has been introduced for feature modeling.

Each feature of instances in positive bags is modeled by a mixed Gaussian distributions. The probability of observing the current feature value f_t is

$$P(f_t) = \sum_{i=1}^{C} w_{i,t} \times \eta(f_t, \mu_{i,t}, \sum_{i,t}) \tag{1}$$

where C is the number of Gaussian distributions, usually varies from 3 to 5. $w_{i,t}$ is the weight of

the i^{th} Gaussian in the mixture model at time t. $\mu_{i,t}$ is the mean value of the i^{th} Gaussian in the mixture model at time t. $\sum_{i,t}$ is the covariance matrix of the i^{th} Gaussian in the mixture model at time t. And η is the probability density function of the i^{th} Gaussian in the mixture model at time t defined by

$$\eta(f_t, \mu_{i,t}, \sum_{i,t}) = \frac{1}{(2\pi)^{\frac{n}{2}} \left| \sum_{i,t} \right|^{\frac{1}{2}}} e^{-\frac{1}{2}(f_t - \mu_{i,t})^T \sum_{i,t}^{-1} (f_t - \mu_{i,t})} \tag{2}$$

And $w_{i,t}$ should satisfy the constraint

$$\sum_{i=1}^{C} w_{i,t} = 1 \tag{3}$$

In this paper, expectation maximum (EM) has been adopted to estimate the initial parameters of GMM model [29].

Each new feature f_{t+1} at time *(t+1)* has been checked whether it matches the existing C Gaussian distributions. If f_{t+1} can be covered by D times standard deviations of a Gaussian distribution in the mixture model, the f_{t+1} is defined as a matched feature of this distribution, where D is a scalar, usually set as 2.5. We can assume that the i^{th} Gaussian distribution in the mixture model is matched. So the parameters are updated based on the following rules

$$w_{i,t+1} = (1-\alpha) w_{i,t} + \alpha \tag{4}$$

$$\mu_{i,t+1} = (1-\rho) \mu_{i,t} + \rho f_{t+1} \tag{5}$$

$$\sigma_{i,t+1}^2 = (1-\rho) \sigma_{i,t}^2 + \rho (f_{t+1} - \mu_{i,t+1})^T (f_{t+1} - \mu_{i,t+1}) \tag{6}$$

Where α is the learning rate for weights, and ρ is the updating rate for mean and standard variance, which are defined as

$$\alpha = B \times (1\text{-}\tanh(\frac{|f_{t+1} - \mu_t|}{2\sigma_t})) \tag{7}$$

$$\rho = \alpha \eta(f_{t+1} \mid \mu_{i,t}, \sigma_{i,t}) \tag{8}$$

The μ and σ parameters for unmatched distributions remain the same during the detection, and the weights are adjusted as follows

$$w_{j,t+1} = (1-\alpha) w_{j,t} \tag{9}$$

If the current feature cannot match those *C* distributions, the distribution with the lowest value $\frac{w}{\sigma}$, which represents the least probability, will be replaced by a new distribution with the current value as its mean value, an initially high standard variance, and a low prior weight.

It is worth noting that, in the Eq. (7), the α is a very important parameter. How to choose α is critical to GMM. If we choose a larger α, the model will be updated more quickly and demonstrate the instability. Otherwise, the model will be updated more slowly, and cannot adapt the changes in the scenes. Therefore, we integrate the differences between feature values and the model into the learning procedure by using a decreasing function *y=(1-tanh(x))*. The α is smaller when facing bigger differences, and the model updates more slowly. Otherwise, the α is larger, and the model can update more quickly. In the experiments, we also introduce *B* to control the maximum learning rate for the weights. Meanwhile, to avoid α is too small to update the model, we use *T* instead of α when α is smaller than *T*.

3.2.2 Modeling features of instances in negative bags

It is assumed that each feature of instances in negative bags obeys a single Gaussian distribution, i.e. $p(\mathrm{f}_k(x_{ij})|y_i = 0) \sim N(\mu_0, {\sigma_0}^2)$

When the new data comes, the parameters of Gaussian distribution are updated as follows

$$\mu_0 = \alpha\mu_0 + \lambda\frac{1}{n}\sum_{j|y_i=0} f_k(x_{ij}) \tag{10}$$

$${\sigma_0}^2 = \alpha{\sigma_0}^2 + \lambda\frac{1}{n}\sum_{j|y_i=0}(f_k(x_{ij}) - \mu_0)^2 \tag{11}$$

Where $f_k(x_{ij})$ is the feature value of the j^{th} instance in the i^{th} negative bag, *n* is the total number of the instances in the i^{th} negative bag, μ_0 is the mean of the Gaussian distribution, σ_0 is the standard variance, and $0 < \lambda < 1$ is the learning rate. Large λ means the parameters can update quickly, otherwise, the parameters updates slowly.

3.3 Train weak classifiers

Each weak classifier h_m is composed by a Haar-like feature f_k and the parameters of the feature distribution. As mentioned above, it is assumed that the features of instances in positive

bags and those in negative bags obey GMM and SGM respectively. The weak classifier is defined as

$$h_k(x) = \log\left[\frac{p(y=1|\mathrm{f}_k(x))}{p(y=0|\mathrm{f}_k(x))}\right] \tag{12}$$

Let $p(y=1) = p(y=0)$ and use Bayes rule to compute Eq. (12). So we can get

$$h_k(x) = \log\left[\frac{p(\mathrm{f}_k(x)|y=1)}{p(\mathrm{f}_k(x)|y=0)}\right] \tag{13}$$

where $p(\mathrm{f}_k(x)|y=1)$ and $p(\mathrm{f}_k(x)|y=0)$ can be estimated by modeling the features of instances in bags as described in Section 3.2.

We can then obtain *M* weak classifiers by Eq. (13), which are candidate weak classifiers to construct a strong classifier.

3.4 Construct strong classifier

The MIL classifier H_K is a strong classifier built up of several weak classifiers h_k, which are related to Haar-like features f_k. *K* weak classifiers are chosen sequentially from the weak classifiers pool $\boldsymbol{\phi} = \{h_1, h_2, ..., h_M\}$ to optimize the following criterion

$$h_k = \arg\max_{h\in\boldsymbol{\phi}} L(H_{k-1} + h) \tag{14}$$

Where H_{k-1} is the strong classifier consisting of the first *(k-1)* weak classifiers, and $L = \sum_i (y_i \log p(y_i \mid X_i) + (1-y_i)\log(1-p(y_i \mid X_i)))$ is the bag log likelihood function. By maximizing the *L*, *K* weak classifiers are greedily selected from $\boldsymbol{\phi}$, and finally a strong classifier $H_K = \sum_{m=1}^{K} h_m$ can be constructed. When processing a new frame, the strong classifier is adopted to determinate the object location in a set of image patches X^s.

It is necessary to estimate the probability of a bag being positive $p(y_i \mid X_i)$ and weak classifier h_m (see Section 3.3) before constructing the strong classifier H_k.

In MIL problem, the training data is denoted by $\{(X_1, y_1), ..., (X_n, y_n)\}$, where $\boldsymbol{X_i} = \{x_{i1}, ..., x_{im}\}$ represents the i^{th} bag, x_{ij} represents the j^{th} instance in the i^{th} bag, and y_i

represents the label of the i^{th} bag. When $y_i = 1$, the i^{th} bag is labeled positive, otherwise, the bag is labeled negative. The bag label is defined as

$$y_i = \max_j (y_{ij}) \tag{15}$$

where y_{ij} is the label of instance x_{ij}, and $y_{ij} \in \{0,1\}$. In addition, the bag labels are known, and the instance labels are unknown in the training stage.

In order to estimate the probability of a bag being positive, Nosiy-OR (NOR) model [16] is adopted.

$$p(y_i \mid X_i) = 1 - \prod_j (1 - p(y_i \mid x_{ij})) \tag{16}$$

Estimating the probability of a bag being positive requires estimating the probability of instances in bags being positive. So the instance probability is defined as

$$p(y_i \mid x_{ij}) = p_{ij} = \sigma(H_K(x)) = \frac{1}{1 + e^{-H_K(x)}} \tag{17}$$

where $H_K(x)$ is the strong classifier mentioned above. The procedure of constructing $H_K(x)$ is described in Algorithm 1.

The procedure of constructing the strong classifier is summarized in Algorithm 1.

1: **Input:** data set $\{X_i, y_i\}_{i=1}^N$, where $\boldsymbol{X_i} = \{x_{i1}, ..., x_{im}\}$ and $y_i \in \{0,1\}$

2: **Output:** MIL classifier $H(x) = \sum_{k=1}^{K} h_k(x)$

Initialization:

3: Update M weak classifier $\{h_j(x)\}_{j=1}^M$ with data set $\{X_i, y_i\}_{i=1}^N$

4: Initialize MIL classifier $H_{i,j}(x) = 0$, for all i and j

Construction of *H(x)*:

5: for k=1 to K do

6: for m=1 to M do

7. Estimate the probability of the instance x_{ij} being positive using the joint of the strong classifier H_{ij} and the current weak classifier h_m: $p_{ij}^m = \sigma(H_{ij} + h_m(x_{ij}))$

8: Estimate the probability of the bag X_i being positive using the joint of the strong classifier H_{ij} and the current weak classifier h_m: $p_i^m = 1 - \prod_j (1 - p_{ij}^m)$

9: End for

10: Select the serial number of the weak classifier that makes the bag log likelihood function obtain the maximum value: $m^* = \arg\max_m L^m$

11: Select the weak classifier which makes the bag log likelihood function obtain the maximum value: $h_k(x) = h_{m^*}(x)$

12: Update the strong classifier: $H_{i,j}(x) = H_{i,j}(x) + h_k(x)$

13: End for

---

4. Experiments

We compare our proposed method with three latest trackers on three challenging video sequences: David indoor, Twinings and Cliffbar, which can be found at http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml. The three trackers are Online AdaBoost (OAB) tracker [22], MIL tracker [18], and WMIL tracker [20], whose parameters for best performance are empirically tuned. The experiments are carried out on a PC platform with Matlab 7.0.1 on a dual-core 2.93GHz CPU and 2G memory.

4.1 Parameter settings

We use a radius $\gamma = 4-6$ to crop the positive instances at each frame, which generates 45-190 positive instances. The inner and outer radius of the negative instances are set as $\xi = 1.5\gamma$ and $\beta = 2s$, where s is usually between 25-35, which randomly generates 42-100 negative instances. The radius for searching the new object location at the next frame is s. It has been noticed that the experimental results are sensitive to the different γ and s. A larger γ should be used only when object appearance changes very fast, and larger s should be applied if the objects moves very fast. We set s=25 for all test video sequences. The number of candidate weak classifiers in the pool is set as M=150-250, and the number of selected weak classifiers to construct the MIL classifier is set as K=15-50. In experiments, M=150 and K=15 are used for most

test video sequences. Only when the video sequence exhibits significant appearance changes, we should choose larger M and K. The number of Gaussian distributions in GMM is usually set as C=3-5, and C=3 is chosen for the consideration of computational complexity. The threshold of updating α is set as T=0.00001 in our experiments. The max learning rate of weights for GMM is set as B=0.0005-0.0015. And the learning rate for the features of instances in negative bags is set as λ=0.7-0.9, and we set λ=0.85 for all the test video sequences.

4.2 Comparison experiments

4.2.1 Qualitative analysis

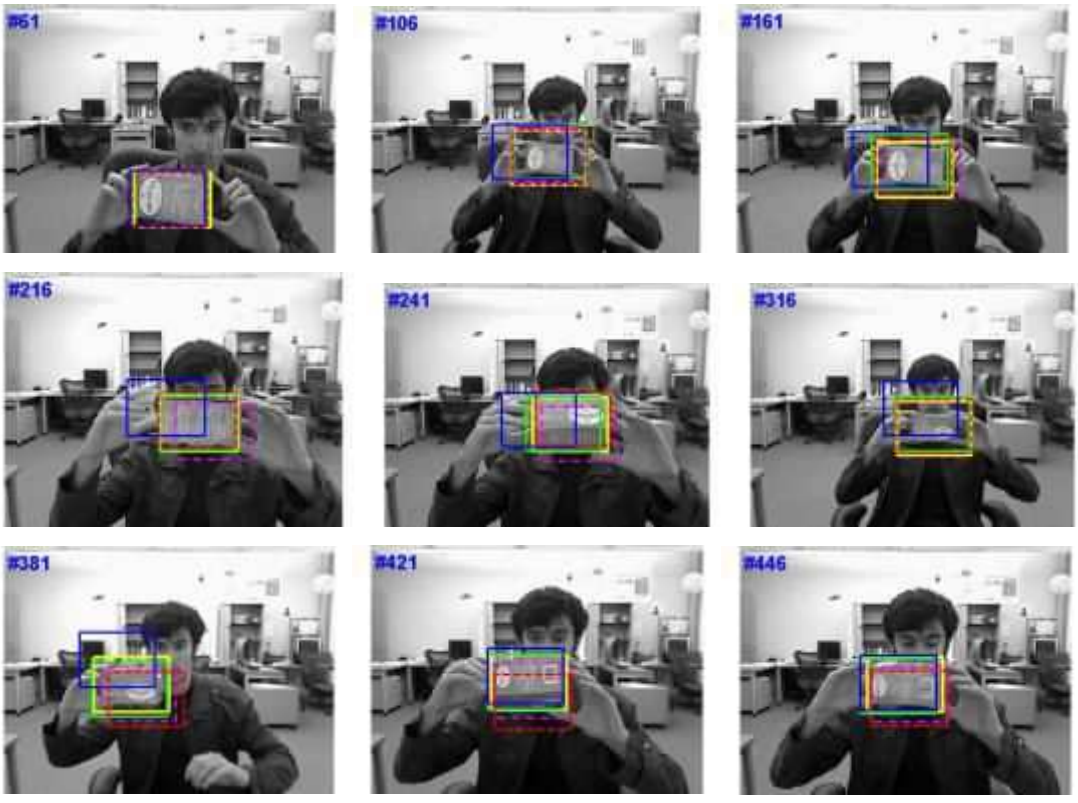
The video sequence "David indoor" presents challenging illumination, pose, appearance and scale changes. It consists of 462 frames with image size 320×240. As seen in Fig. 4(a), the OAB tracker cannot deal with these changes in the scene, and demonstrates serious drift as shown in frames #386, #436 and #461. Although MIL and WMIL yields more stable results than OAB, some of the results are imprecise when the changes are coming from pose and appearance, especially when David wears or removes glasses as seen in #311, #386 and #436. On the contrary, our proposed method achieves the best performance compared with the other three trackers.

The video sequence "Twinings" comprises massive rotation and appearance changes. It consists of 471 frames with image size 320×240. As illustrated in Fig. 4(b). When the object rotates, the OAB yields severe drift problem as shown in frames #216, #241, #316 and #381. MIL and WMIL are more stable than OAB. But when the object rotates as shown in frames #381, #421 and #446, MIL and WMIL cannot track the object effectively. On average, our proposed method can handle rotation and appearance changes well, yielding much more stable and accurate results than other compared trackers.

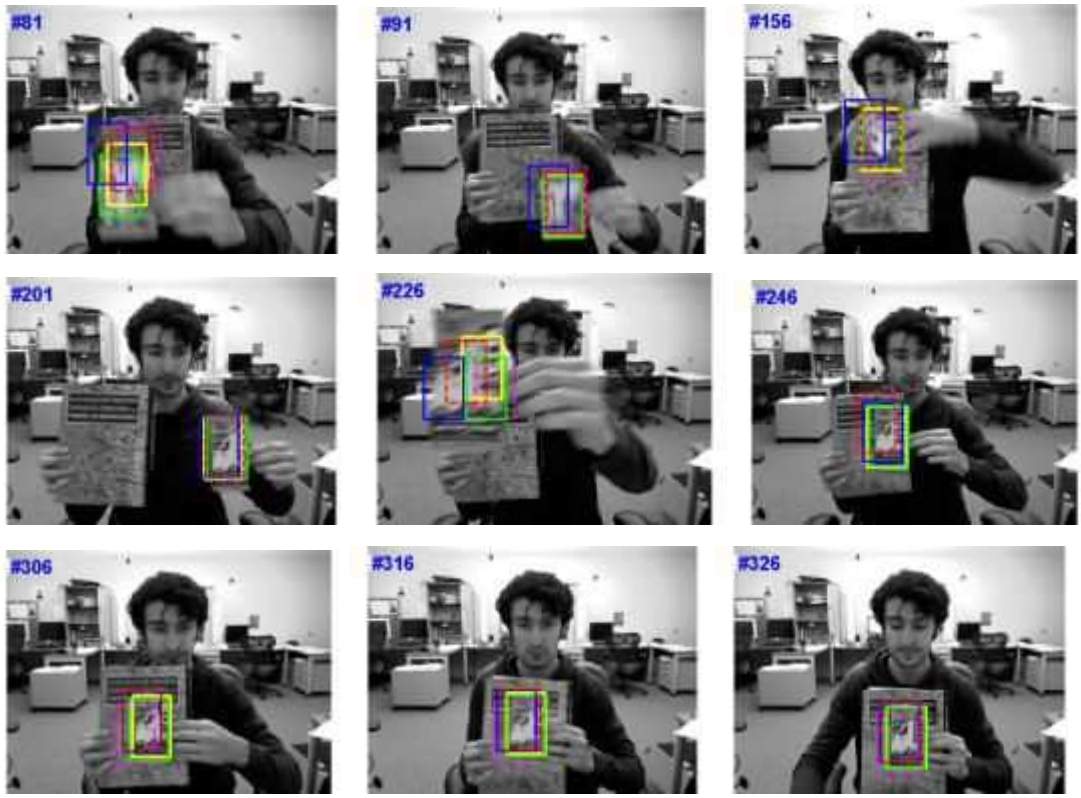
The video sequence "Cliffbar" exhibits dramatically appearance change, serious motion blur and similar texture between the object and the background. It consists of 327 frames with image size 320×240. As shown in Fig. 4(c), WMIL and our method can deal with slight blur as shown in frames #91 and #156. But when motion blur is severe as shown in frames #81 and #226, the four tracks yields imprecise results. It is worth noting that although the object is similar to the background, WMIL and our method can track it better than OAB and MIL.


#11
#96
#176
#211
#261
#311
#386
#436
#461

(a)


#61
#106
#161
#216
#241
#316
#381
#421
#446

(b)



(c)

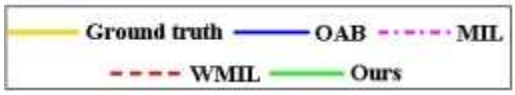


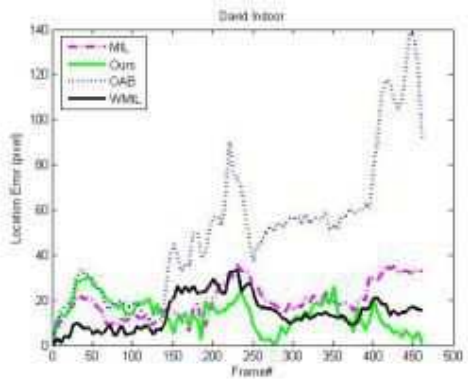
Fig. 4. Tracking results for test video sequences: (a) David indoor, (b) Twinings, and (c) Cliffbar.

4.2.2 Quantitative analysis

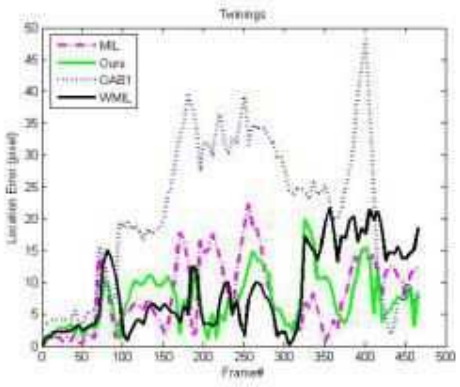
Two criteria are used to evaluate the performances of the developed trackers. The first one is the center location error, which is defined as the Euclidean distance between the detected object center and the ground truth center at each frame. Meanwhile, the maximum, mean and standard deviation of the center location error for each video sequence are also calculated. The second one is the failure rate (FR), which is defined as the number of failure frames divided by the total number of frames in one video sequence. And the failure frame is indicated when the intersection of the ground truth bounding box and the tracking bounding box is less than half of the union of the ground truth bounding box and the tracking bounding box.

The center location error plots are illustrated in Fig. 5. As showed in Fig. 5(a), in the first 150 frames of David indoor, WMIL works better. And in the rest of frames, our method performs more stable. In Fig. 5(b), before frame #350 of Twinings, MIL, WMIL and our method have similar precision. However, our method have the smallest error after frame #350. As seen in Fig. 5(c), MIL, WMIL and our method have similar precise. OAB performs worse than the others on Cliffbar, especially with those video clips containing certain motion blurs.

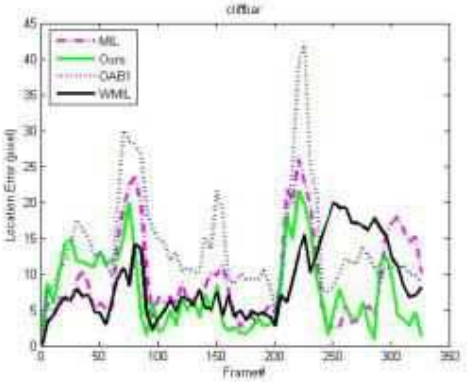
The maximum, mean and standard deviation of the center location errors are reported in Table1. Our method works best with the smallest maximum, mean and standard deviation of the center location error on David indoor and Twinings. For Cliffbar, our method has the smallest mean of the center location error, while WMIL has the smallest maximum and standard deviation of the center location error. Table 2 reports the FR of the four trackers. Our method has the smallest FR implying our method is more accurate and robust than the other three trackers.



(a)



(b)



(c)

Fig. 5. Location error plots for test video sequences: (a) David indoor, (b) Twinings, and (c) Cliffbar.

Table 1

Center location errors (in pixels) for test video sequences. Bold fonts indicate the best performance.

Method / Sequence	OAB			MIL			WMIL			Ours		
	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std	Max	Mean	Std
David indoor	138.96	51.03	30.83	35.63	19.63	8.18	33.44	14.44	7.74	**30.42**	**13.55**	**7.10**
Twinings	48.24	21.13	11.93	22.29	8.00	5.56	21.73	8.78	6.37	**19.80**	**7.26**	**4.24**
Cliffbar	41.85	14.00	7.43	25.99	9.19	6.30	**19.97**	8.64	**5.08**	21.52	**7.41**	5.31

Table 2

Failure rate (FR) (%) for test video sequences. Bold fonts indicate the best performance.

Sequence	OAB	MIL	WMIL	Ours
David indoor	74.19	27.96	7.53	**4.30**
Twinings	58.51	5.32	18.09	**3.19**
Cliffbar	42.42	24.24	24.24	**12.12**

5. Conclusion

In this paper, we propose a MIL tracker that respectively employs GMM and single Gaussian distribution to model features of instances in positive bags and those in negative bags. And the differences between samples and the model are integrated into the process of updating the

parameters for GMM. With these Haar-like features extracted from bags, a set of weak classifiers are trained to construct a strong classifier, which is used to track the object location at a new frame. And the classifier is updated frame by frame. Experimental results have shown that our tracker is more stable and efficient in dealing with the illumination, rotation, pose, and appearance changes. All of the test video sequences are gray-scale, and our method can be extended to solve object tracking problem in color data set by computing Haar-like feature over color channels. Moreover, it can be applied to other applications such as object detection.

Acknowledgments

This work was supported by National Science Foundation of China (No. 51278058), Natural Science Research Project of Education Department of Shaanxi Province (No. 12JK0731), and Young Teacher Research Foundation of Xi′an University of Posts and Telecommunications (No. ZL2013-04).

References

[1] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Computing Surveys 38(2006).

[2] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-based tracking using the integral histogram, in: IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 798–805.

[3] S. Avidan, Ensemble tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007) 261–271.

[4] D. Ross, J. Lim, R. Lin, M. Yang, Incremental learning for robust visual tracking, International

Journal of Computer Vision, 77 (2008) 125–141.

[5] S. Avidan, Support vector tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 1064–1072.

[6] R. Collins, Y. Liu, M. Leordeanu, Online selection of discriminative tracking features, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 1631–1643.

[7] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2003) 564–575.

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

[9] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-Based tracking using the integral histogram, Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1 (2006) 798-805.

[10] H. Grabner, C. Leistner, H. Bischof, Semi-supervised on-line boosting for robust tracking, in: European Conference on Computer Vision, (2008) 234–247.

[11] M. Black, A. Jepson, Eigentracking: robust matching and tracking of articulated objects using a view-based representation, European Conference on Computer Vision, (1996) 329–342.

[12] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, S. Yan, Sparse representation for computer vision and pattern recognition, Proc. IEEE 98 (6) (2010) 1031–1044.

[13] T. Bai, Y.F. Li, Robust visual tracking with structured sparse representation appearance model, Pattern Recognition, 45 (6) (2012) 2390–2404.

[14] C.J. Xie, J.Q. Tan, P. Chen, J. Zhang, L. He, Collaborative object tracking model with local sparse representation, Journal of visual communication and image representation, 25 (2014) 423–434.

[15] M. Andriluka, S. Roth, B. Schiele. People-tracking-by-detection and people-detection-by-Tracking,Proc, IEEE Conf. Computer Vision and Pattern Recognition, (2008).

[16] P. Viola, J. C. Platt, and C. Zhang, Multiple instance boosting for object detection, In NIPS, (2005) 1417–1426.

[17] T.G. Dietterich, R.H. Lathrop, and T. Lozano-Pérez T, Solving the multiple-instance problem with axis-parallel rectangles, Artificial Intelligence, 89(1-2) (1997) 31-71.

[18] B. Babenko, M. Yang, S. Belongie, Visual tracking with online multiple instance learning, CVPR, 2009, 983-990.

[19] B. Babenko, M. Yang, S. Belongie, Robust object tracking with online multiple instance learning, IEEE Transactions on Pattern Analysis and Machine Intelligence, 33,(2011) 1619-1632.

[20] K. Zhang, H. S. Real-time visual tracking via online weighted multiple instance learning. IEEE Transactions on Pattern Recognition, 2013, 46(1): 397-411.

[21] Z.Ni, S.Sunderrajan, A. Rahimi, B.S. Manjunath, Particle filter tracking with online multiple instance learning, 2010 International Conference on Pattern Recognition, (2010) 2616-2619.

[22] H. Grabner, M. Grabner, and H. Bischof, Real-time tracking via online boosting, Proc. Conf. British Machine Vision, (2006) 47-56.

[23] H. C. Lu, Q. H. Zhou, D. Wang, R. Xiang, A Co-training Framework for Visual Tracking with Multiple Instance Learning, Ninth IEEE International Conference on Automatic Face and Gesture Recognition , (2011) 539-544.

[24] J. Wang, J. D. Zucker, Solving the multiple-instance problem: a lazy learning approach,

Proceedings of the 17th International Conference on Machine Learning, (2000) 1119-1125.

[25] O. Maron, A. L. Ratan. Multiple-instance learning for natural scene classification, Proceedings of the 15th International Conference on Machine Learning, (1998) 341-349.

[26] Q. Zhang, S. Goldman, Em-dd: An improved multiple instance learning technique, Advances in Neural Information Processing Systems, (2002) 1073-1080.

[27] M. L. Zhang, Z. H. Zhou, A multi-instance regression algorithm based on neural network, Journal of Software, 14(7) (2003) 1238-1242.

[28] C. Stauffer, W. E. L. Grimson. Adaptive background mixture models for real-time tracking, IEEE Computer Society, (2) (1999) 246-252.

[29] A. Dempster, N. Laird, D. Rubin. Maximum likelihood from incomplete data via the EM algorithm, Royal Statistical Society, 39 (Series B) (1977) 1-38.