

## Research Article

# Patch Based Multiple Instance Learning Algorithm for Object Tracking

Zhenjie Wang,<sup>1</sup> Lijia Wang,<sup>1</sup> and Hua Zhang<sup>2</sup>

<sup>1</sup>Department of Information Engineering and Automation, Hebei College of Industry and Technology, Shijiazhuang, China

<sup>2</sup>Faculty of Electrical & Electronics Engineering, Shijiazhuang University of Applied Technology, Shijiazhuang, China

Correspondence should be addressed to Lijia Wang; wanglijia1981@hotmail.com

Received 17 September 2016; Revised 19 December 2016; Accepted 9 January 2017; Published 22 February 2017

Academic Editor: Silvia Conforto

Copyright © 2017 Zhenjie Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To deal with the problems of illumination changes or pose variations and serious partial occlusion, patch based multiple instance learning (P-MIL) algorithm is proposed. The algorithm divides an object into many blocks. Then, the online MIL algorithm is applied on each block for obtaining strong classifier. The algorithm takes account of both the average classification score and classification scores of all the blocks for detecting the object. In particular, compared with the whole object based MIL algorithm, the P-MIL algorithm detects the object according to the unoccluded patches when partial occlusion occurs. After detecting the object, the learning rates for updating weak classifiers' parameters are adaptively tuned. The classifier updating strategy avoids overupdating and underupdating the parameters. Finally, the proposed method is compared with other state-of-the-art algorithms on several classical videos. The experiment results illustrate that the proposed method performs well especially in case of illumination changes or pose variations and partial occlusion. Moreover, the algorithm realizes real-time object tracking.

## 1. Introduce

Object tracking is an important topic in computer vision, and it has attracted more and more attention [1]. Recently, researchers have studied the issue and proposed many excellent algorithms. However, the object tracking is still a challenging task because there are noise, pose variations, illumination changes, occlusion, and so on [2].

An efficient appearance model plays an important role in object tracking system. Studies categorize the appearance models into two classes: generative appearance model and discriminative appearance model [3]. Generative appearance algorithms [4–7] take the tracking problem as searching for the region with the maximum score. The methods represent an object in a particular feature space such as edge, color, and HOG. By using the appearance model, the algorithms search for interesting regions with the minimal error in the successive frames. It is demonstrated that the algorithms have achieved favorite tracking results. However, the generative algorithms take into account only the feature of object but ignore the background information which are helpful for recognizing object from background [1].

The discriminative methods [8–12] formulate the tracking problem as learning a discriminative classifier which is updated over time to separate object from background. The classifier updates its parameters considering the information from both the object and background to handle the issues of appearance changes as tracking evolves. Grabner et al. [13] presented an online boosting feature selection algorithm for object tracking. However, only one positive sample (the tracking result) is used for classifier updating. Once the tracking result drifts away from the ground truth, the positive sample is inaccurately cropped [14]. Therefore, the interference from background will be introduced into the classifier, which leads to tracking failure. To handle this problem, Grabner et al. [15] presented a semisupervised object tracking algorithm which labeled the positive samples in the first frame. Then, Babenko et al. proposed multiple instance learning (MIL) algorithm [8] which puts samples into bags. The MIL algorithm represents samples in labeled bags. The bag is labeled as positive or negative according to the contained instances. A bag is positive if at least an instance is positive, while a bag is negative when all the instances are negative [8]. The MIL algorithm learns the information from

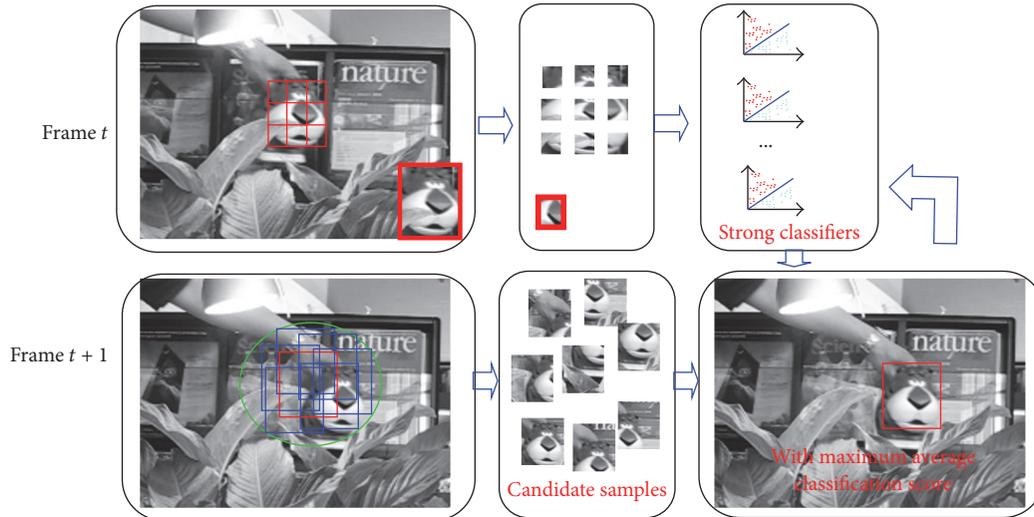


FIGURE 1: The flow chart of patch based MIL algorithm.

the instances in the positive and negative bags to obtain a discriminative classifier. Then, the classifier recognizes the object from background. The MIL algorithm overcomes the problem of ambiguity in object tracking.

The online MIL algorithm often fails to track an object when there is pose or illumination variations or serious occlusion. Moreover, it often suffers from computational cost. Researchers proposed some improved MIL algorithms to deal with these problems. Zhang and Song proposed a weighted multiple instance learning (WMIL) algorithm for saving the computational time [16]. The algorithm selects the most powerful weak classifiers from a classifier pool by maximizing the inner product between weak classifiers and the maximum likelihood probability, which avoids computing the bag and instance probability  $M$  time before a powerful classifier is selected. Xu et al. [14] selected the powerful weak classifiers by using the Fisher criterion to save the computational time. Zhou et al. [17] overcame the problem of tracking drift by computing the bag probability based on the instance importance. Wang and Zhang [18] proposed an adaptive update strategy to adjust the learning rate to deal with the problems of occlusion, pose, and illumination variations.

Therefore, a patch based multiple instance learning (P-MIL) algorithm is proposed to further improve the tracking performance and overcome the problems mentioned above. The algorithm divides an object into many blocks. Then, the multiple instance learning algorithm (MIL) is applied on each block. For each block, the positive and negative bags are composed by the instances cropped surrounding the object's position in previous frame. By training the cropped instances, weak classifiers are learned. Then, strong classifiers are generated for the blocks. The average classification score for the whole object is calculated to detect the object. Moreover, based on the average classification score and the classification scores of all the blocks, the tracking cases (occlusion, pose variations, and illumination changes) are detected and the learning rate is adjusted accordingly. Furthermore, the inner product method [16] used in WMIL algorithm is utilized to

select the powerful weak classifiers for saving the computing time. Finally, we compared the P-MIL algorithm with the other MIL related algorithms in several classical videos.

The paper is organized as follows. In Section 2, we present the patch based MIL algorithm. The experiment results are illustrated in Section 3. We make a summary in Section 4.

## 2. Patch Based MIL Algorithm

We proposed a patch based MIL algorithm (P-MIL) to deal with the problems mentioned above. Compared with the method representing an object in the whole, patches based method is robust to partial occlusion [19]. The P-MIL method divides an object into many blocks. Then, the online MIL algorithm is applied on each block. By training the instances in the positive and negative bags, the strong classifiers for all the blocks are obtained. In the tracking process, the obtained strong classifiers are used and the average classification scores are computed to determine the object location. Furthermore, the P-MIL algorithm detects the tracking situations including occlusion, pose changes, and illumination variations according to the average classification score and the classification scores of all the blocks. To deal with the problems mentioned above, the learning rate is adaptively tuned to update the parameters of weak classifiers. The flow chart of the proposed P-MIL algorithm is shown in Figure 1.

**2.1. Patch Based MIL Algorithm.** To realize real-time object tracking, the object is divided into 9 blocks. For block  $o_i$ , the instances from the block and background are cropped to compose positive and negative bags, respectively. The bag is labeled "1" as it is positive, while the negative bag is assigned "0." The positive bag is obtained by cropping instances surrounding it:  $X_i^{\alpha} = \{x : \|l(x) - l_i^t\| < r_i\}$ ,  $i = 1, \dots, 9$ .  $l_i^t$  is the position of block  $o_i$ ;  $r_i$  is the radius of the circle. The negative bag is obtained by cropping the instances from an annulus region around the block:  $X_i^{\beta} = \{x : r_i <$

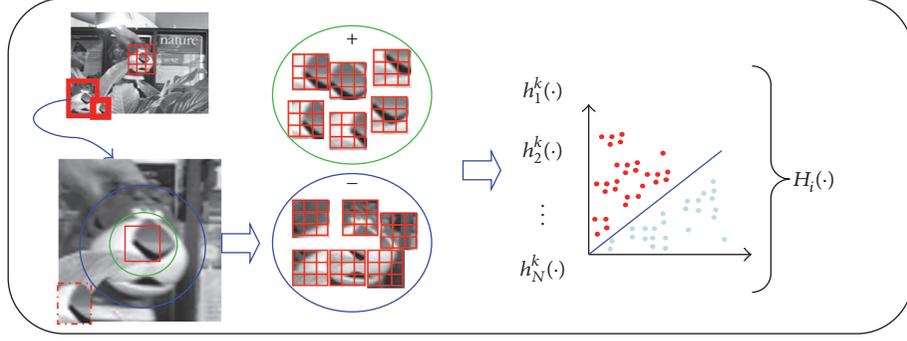


FIGURE 2: The process of applying the MIL on each block.

$\|l(x) - l_i^t\| < \beta_i\}$ ,  $i = 1, \dots, 9$ . By training the instances in the positive and negative bags, we learn the weak classifiers and obtain the classifier pool:  $\phi_i = \{h_i^1, \dots, h_i^k, \dots, h_i^N\}$ , where  $h_i^k$ ,  $i = 1, \dots, 9$ ,  $k = 1, \dots, N$ , is the  $k$ th weak classifier for the  $i$ th block.

The instances in the positive and negative bags are represented by the Haar-like features [8]. It is assumed that the features of all the instances satisfy the Gaussian distribution, which means  $p(v_k(x) | y = 1) \sim N(\mu_k^1, \sigma_k^1)$ ,  $p(v_k(x) | y = 0) \sim N(\mu_k^0, \sigma_k^0)$ , and  $p(y = 1) = p(y = 0)$ . Then, the  $k$ th weak classifier with parameters  $(\mu_k^1, \sigma_k^1, \mu_k^0, \sigma_k^0)$  is obtained by training the  $k$ th feature of all the instances in the positive and negative bags:

$$h_i^k(x) = \log \left( \frac{p(y = 1 | v_k(x))}{p(y = 0 | v_k(x))} \right). \quad (1)$$

Then, the strong classifier  $H_i$  is generated by selecting  $K$  ( $\ll N$ ) powerful weak classifiers from the weak classifier pool  $\phi_i$ .

$$H_i(x) = \sum_{k=1}^K h_i^k(x). \quad (2)$$

To save computing time, we use the inner product strategy [8] to select the powerful weak classifiers. The strategy computes the inner product between the weak classifier and log-likelihood function, which avoids computing the bag probability and instance probability  $N$  times before selecting a weak classifier. The process of applying the MIL on each block is shown in Figure 2.

After applying the MIL algorithm on each block, strong classifiers  $H_i$  ( $i = 1, \dots, 9$ ) are obtained and used for object tracking. In the successive frames, candidate samples  $X^s = \{x : \|l(x) - l_{t-1}^*\| < s\}$  are extracted from the neighbor of the object location in the previous frame. For each sample, we also divide it into many blocks. Then, the classification scores of all blocks for a sample are calculated by using the corresponding strong classifiers:  $S_i^j$ ,  $i = 1, \dots, 9$ ,  $j = 1, \dots, N_s$ , where  $N_s$  is the number of the candidate samples. The candidate sample's average classification score is obtained by averaging the classification scores of all the blocks.

$$S^j = \frac{1}{9} \sum_{i=1}^9 S_i^j. \quad (3)$$

The candidate sample with the maximum average classification score  $S_{\max}^j$  is considered as the tracking result.

**2.2. The Illumination, Pose, and Occlusion Problems.** The P-MIL algorithm considers the average classification score and the classification scores of all the blocks to handle the problems of illumination and pose changes and occlusion. Normally, the sample which is considered as the tracking result has the maximum average classification score and the score is greater than a given threshold. However, when the problems mentioned above happen, the maximum average classification score and the classification scores for all the blocks decrease seriously. We also see that the classification scores for all the blocks of the tracking result decrease simultaneously in case of pose and illumination variations, while the classification scores for some blocks decrease seriously in case of occlusion. Therefore, we present a method to detect the tracking situations by analyzing the classification scores for all the blocks.

We have obtained the maximum average classification score ( $S_{\max}^j$ ) and the classification scores for all the blocks ( $S_i^j$ ) of the detected candidate sample. Then, we set two thresholds  $th_1$  and  $th_2$  ( $th_1 < th_2$ ) as the lower threshold and higher threshold, respectively. Finally, the tracking situations are detected by analyzing the relationship between the classification score and two thresholds  $th_1$  and  $th_2$  ( $th_1 < th_2$ ):

- (1) The object is successfully tracked when the maximum average classification score is greater than the higher threshold:  $S_{\max}^j > th_2$ . In such a case, it is considered that there are not illumination changes, pose variations, and occlusion.
- (2) The object is occluded by the other things when the maximum average classification score is between the thresholds  $th_1$  and  $th_2$  ( $th_1 < S_{\max}^j < th_2$ ) and the classification scores for the unoccluded blocks are greater than the threshold  $th_2$  ( $S_i^j > th_2 \forall i$ ). In such a case, the classification scores for the occluded blocks are smaller than the threshold  $th_2$  ( $S_i^j < th_2 \forall i$ ) (some classification scores may be smaller than the threshold  $th_1$  ( $S_i^j < th_1 \forall i$ )).
- (3) The issues of illumination variation or pose change happen when the maximum average classification score is between the thresholds  $th_1$  and  $th_2$  ( $th_1 <$

$S_{\max}^j < th_2$ ) and the classification scores for all of the blocks are also between the two thresholds  $th_1$  and  $th_2$  ( $th_1 < S_i^j < th_2$ ).

- (4) The algorithm fails to track the object when the maximum classification score and the classification scores for all the blocks are smaller than the threshold  $th_1$  ( $S_{\max}^j < th_1$ ,  $S_i^j < th_1$ ).
- (5) The algorithm fails to track the object when the maximum classification score and the classification scores for all the blocks are smaller than the threshold  $th_2$  for many successive frames ( $S_{\max}^j < th_2$ ,  $S_i^j < th_2$ ).

**2.3. Classifier Update Strategy.** After tracking an object, the parameters of weak classifiers are updated to deal with the issues of illumination and pose variations and occlusion. The MIL, WMIL, and significance-MIL algorithms update their parameters with constant learning rate [5].

$$\begin{aligned}\mu_k^1 &= \lambda \mu_k^1 + (1 - \lambda) \mu^1, \\ \sigma_k^1 &= \sqrt{\lambda (\sigma_k^1)^2 + (1 - \lambda) (\sigma^1)^2 + \lambda (1 - \lambda) (\mu_k^1 - \mu^1)^2}, \\ \mu_k^0 &= \lambda \mu_k^0 + (1 - \lambda) \mu^0, \\ \sigma_k^0 &= \sqrt{\lambda (\sigma_k^0)^2 + (1 - \lambda) (\sigma^0)^2 + \lambda (1 - \lambda) (\mu_k^0 - \mu^0)^2},\end{aligned}\quad (4)$$

where  $0 < \lambda < 1$  is the learning rate.  $\mu^1 = (1/n) \sum_{i|y_i=1} v_k(x_i)$  and  $\mu^0 = (1/n) \sum_{i|y_i=0} v_k(x_i)$  are the mean values for the instances in the positive and negative bags, respectively.  $\sigma^1 = \sqrt{(1/n) \sum_{i|y_i=1} (v_k(x_i) - \mu^1)^2}$  and  $\sigma^0 = \sqrt{(1/n) \sum_{i|y_i=0} (v_k(x_i) - \mu^0)^2}$  are the variance values for the instances in the positive and negative bags, respectively.

These methods can resolve the problems mentioned above to some extent. However, it is difficult to handle the serious issues. With a large learning rate, the information of the occluding thing will be introduced into classifiers after updating. Thus, the small learning rate is helpful for the occlusion case. However, with a small learning rate, the classifier will suffer from underupdating when there is illumination or pose changes. Therefore, we present an adaptive updating algorithm to handle these problems. We have detected the tracking cases (e.g., normal tracking, illumination or pose variations, and partial occlusion) in Section 2.2. Then, different learning rates in different tracking cases are tuned as follows:

$$\lambda = \begin{cases} 0.85 & th_1 < S_{\max}^j < th_2, \\ & th_1 < S_i^j < th_2 \quad \forall i; \\ 0.5 & S_{\max}^j > th_2; \\ 0.25 & th_1 < S_{\max}^j < th_2, \\ & S_i^j > th_2 \text{ for some } i. \end{cases}\quad (5)$$

The learning rate is 0.5 in the normal case. Then, the classifier takes account of both the tracking result and the

model to update its parameters. The object's appearance changes much when there is illumination or pose variations. Thus, the learning rate is set as 0.85 for updating the classifiers' parameters mainly depending on the tracking result. The classification scores of some blocks are smaller than the threshold  $th_2$  (even  $th_1$ ) when there is partial occlusion. The learning rate is set as 0.25 for updating the classifiers' parameters of the unoccluded blocks. For the occluded blocks, the classifiers' parameters are not updated. Therefore, the method avoids introducing information from the occluding thing. The adaptive update strategy avoids overupdating and underupdating the classifiers' parameters and resolves the issues of illumination and pose variations and occlusion.

### 3. Experimental Results

In this section, the proposed P-MIL is compared with the MIL, WMIL, and significance-MIL algorithms on several videos including "David indoor" [20], "Face occluded" [20, 21], "Tiger" [20, 21], and "Dollar" [18]. There is pose and illumination changes in the "David indoor" video. In the "Face occluded" video, the face is often occluded by a book or a hat. There are also pose variations. In the "Tiger" video, the object moves fast and is often occluded by other things. In the "Dollar" video, the object is confused by the similar features in the background. The performance of the above algorithms is evaluated in terms of tracking results, failure rate (FR), center location error, and the average computing time. All of the algorithms are implemented in the Matlab language.

**3.1. Parameter Setting.** For the online MIL boosting tracker [8], the radius for cropping instances in the positive bag and negative bag is set as  $r = 4$  and  $\beta = 50$ , respectively. Thus, about 45 instances are cropped for the positive bag, while about 45 instances are for the negative bag. The MIL algorithm learns 250 weak classifiers. About 50 classifiers are selected to generate the strong classifier. For getting the final tracking result, we extract 1000 candidate samples in a circle with the radius  $s = 35$  and centering at the previous object location. For the WMIL tracker [16], the instances in the positive bag are cropped in a circle with radius  $r = 4$  and the instances in the negative bag are cropped in an annulus with radius  $a = 2r$  and  $\beta = 1.5s$ .  $s = 25$  is the searching radius for extracting the candidate samples. The number of weak classifiers in classifier pool is 150. We select 15 powerful weak classifiers to generate a strong classifier for object tracking. For the significance-MIL [17], we set  $r = 4$  and  $\beta = 50$  to crop instances for the positive and negative bags, respectively. The number of the weak classifiers is 150, while that of the selected weak classifiers is 15. The learning rates for the MIL, WMIL, and significance-MIL algorithms are set to be constant: 0.85. For the P-MIL algorithm, the object is divided into 9 blocks. For each block, the number of the weak classifiers is 20, while that of the selected powerful weak classifiers is 3. To obtain the best performance, two thresholds are set as 0.8 and 0.3 as the higher and lower threshold, respectively.



FIGURE 3: The tracking results by using the MIL, WMIL, significance-MIL, and P-MIL algorithms.

**3.2. Tracking Location.** The tracking results are denoted by rectangle boxes in the images which are shown in Figure 3. The results in the first line are for the “David indoor” video. There are illumination variations (e.g., frames 8 and 83) and pose changes (e.g., frame 134). The WMIL and significance-MIL algorithms drift away in frames 281 and 371. The tracking results for the “Face occluded 2” video are shown in line 2. The “face” is frequently occluded by a book (e.g., frames 279, 495, and 708) or a hat (e.g., frames 495 and 708). Using the MIL, WMIL, and significance-MIL algorithms, the interference from background (the book and hat) is introduced into the classifier and leads to tracking drift. The “tiger” moves fast and is often occluded by leaves in the “tiger” video. The tracking results are illustrated in the third line. The MIL algorithm loses the “tiger” in frame 81 because it suffers from a heavy computation load and cannot track the fast object. The WMIL and significance-MIL methods drift away from the ground truth when occlusion occurs (in frames 241 and 321). The last line details the tracking results of the “Dollar” video. The object “Dollar” is affected by the other similar “Dollar,” which results in the fact that the MIL, WMIL, and significance-MIL algorithms drift away in frame 112. Overall, the tracking results in Figure 3 demonstrate that the P-MIL outperforms the other algorithms when there are pose and illumination variations and partial occlusion.

**3.3. Failure Rate.** We use failure rate (FR) to evaluate the performance of the MIL, WMIL, significance-MIL, and P-MIL algorithms. The tracking box with the overlap region less than 50% is considered as failure. The tracking results are shown in Table 1. The object is often affected by the issues of pose and

TABLE 1: The FR (%) by using the MIL, WMIL, significance-MIL, or P-MIL algorithm.

	MIL	WMIL	Significance-MIL	P-MIL
David indoor	14.29	0	1.421	0
Face occluded	16.05	8.21	0.798	0.24
Tiger	23.13	12.14	0.431	0.42
Dollar	0	0	0	0

illumination variations and partial occlusion in the tracking process. The MIL, WMIL, and significance-MIL algorithms update the classifiers with a fixed learning rate. Consequently, the classifiers are “overupdating” or “underupdating” due to the partial occlusion and illumination changes. Unlike other algorithms with a fixed learning rate, the proposed P-MIL algorithm discriminates these issues and updates the learning rate of the classifier accordingly. As a result, the P-MIL algorithm outperforms other MIL based tracking algorithms as the problems mentioned above occur. The results in Table 1 show that the P-MIL algorithm achieves the lowest failure rate.

**3.4. The Center Location Error.** This section details the center location error between central locations of the tracking results and the centers of the ground truth. The results are shown in Figure 4. The smaller the area below the center error line is, the better the tracking algorithm performs. The red lines are the center location error for the proposed MIL algorithm. The tracking results illustrate that the areas below the red lines are the smallest. The blue lines are for the MIL

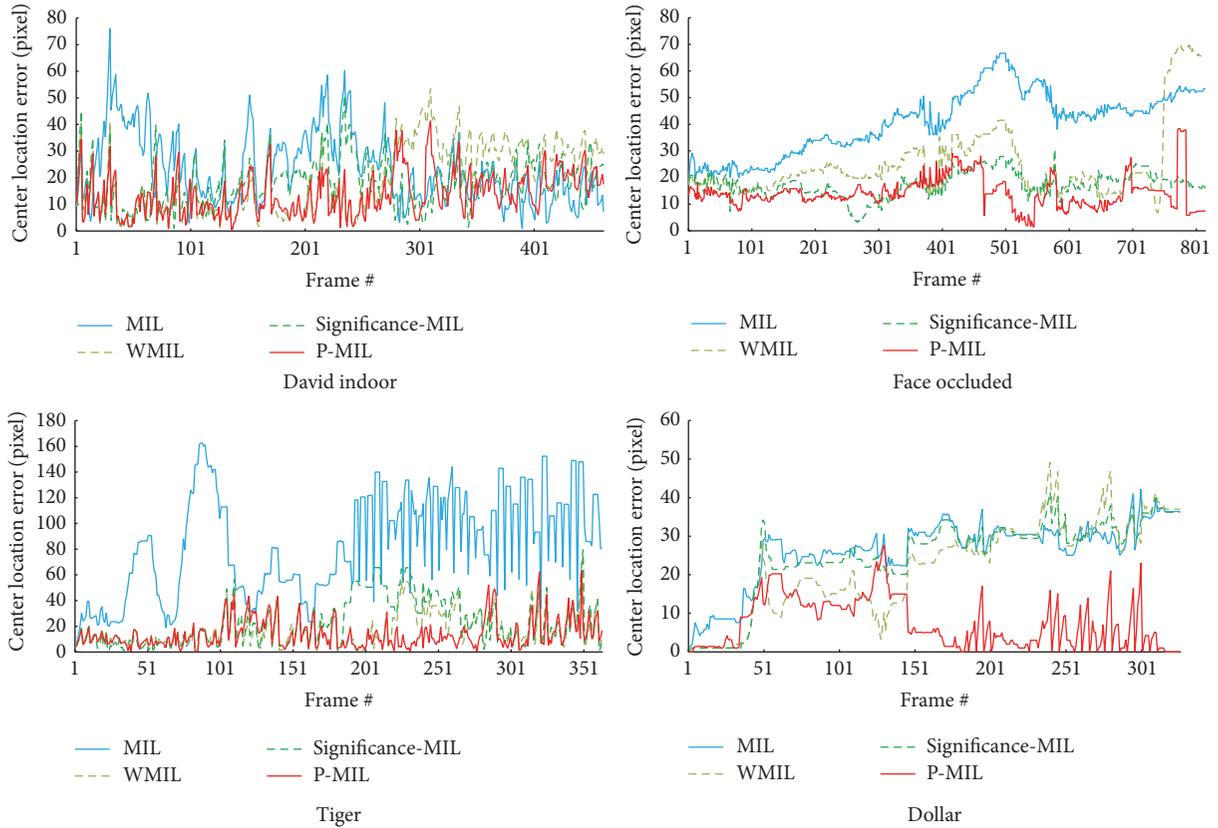


FIGURE 4: The center location error by using the MIL, WMIL, significance-MIL, and P-MIL algorithms.

TABLE 2: The average computing time (ms) by using the MIL, WMIL, significance-MIL, or P-MIL algorithm.

	MIL	WMIL	Significance-MIL	P-MIL
David indoor	1.122	0.108	1.031	0.112
Face occluded	1.601	0.104	0.123	0.097
Tiger	1.092	0.104	1.091	0.121
Dollar	1.156	0.102	0.982	0.085

algorithm. The areas under the blue lines are the largest. The yellow lines are for the WMIL, while the green lines are for the significance-MIL. The results show that the P-MIL algorithm outperforms the MIL, WMIL, and significance-MIL algorithms.

**3.5. The Average Computing Time.** This section details the average computing time of the MIL, WMIL, significance-MIL, and P-MIL algorithms. The average computing time is the average processing time of a frame. The results are shown in Table 2. The lower the average computing time is, the better computational efficiency the algorithm has. The average computing time of the MIL based algorithms depends on parameters including the number of the weak classifiers in the classifier pool, the number of the selected weak classifiers, and the classifier selected strategy. The parameters in Section 3.1 illustrate that the number of the weak classifiers in the classifier pool and the number of the selected weak

classifiers for the WMIL and significance-MIL algorithms are 150 and 30, which are smaller than those of the MIL algorithm. Furthermore, the WMIL and significance-MIL algorithms present efficient classifier selection strategies to avoid computing the bag probability and instance probability  $M$  times before selecting powerful weak classifiers, which also efficiently saves computing time. For the P-MIL algorithm, we learn about 180 weak classifiers for 9 blocks (about 20 weak classifiers are learned for a block), of which 27 weak classifiers are selected (for a block, we select 3 weak classifiers to generate a strong classifier). Moreover, we employ the weak classifier selecting strategy used in the WMIL algorithm. Therefore, the P-MIL algorithm avoids suffering from a high computational load. The results demonstrate that the P-MIL algorithm realizes real-time object tracking.

## 4. Conclusion

A patch based MIL algorithm was proposed to resolve the issues of illumination and pose variations and partial occlusion. The object was divided into 9 blocks and the MIL algorithm was applied on each block for generating 9 strong classifiers. Furthermore, the strong classifiers were used to detect the sample with the maximum average classification score as the tracking result. By analyzing the average classification score and classification scores of the sample's blocks, the illumination, pose, and occlusion problems were detected. Moreover, an adaptive learning rate update strategy

was presented to avoid overupdating or underupdating. The strategy varied the learning rate based on the detected tracking situations. Thus, the updated classifiers overcome the influence from the issues of pose and illumination variations and partial occlusion. Finally, the experimental results have shown that the proposed P-MIL algorithm outperformed other related MIL algorithms in terms of computational efficiency and ability of overcoming the problems of illumination and pose variations and partial occlusion.

The object tracking algorithms are often evaluated in the classical videos which are from the Internet. However, for real videos, for example, the video from the surveillance, there is noise and the images are with low resolution. These algorithms often fail to track the object successively. Therefore, we will focus on learning a robust tracker for real videos in the future.

The MIL related algorithms cropped candidate samples around the tracking position in the previous frame. When the object moves fast, the algorithms will lose the object. To deal with the problem, we will consider motion prediction algorithm in the MIL frame.

## Competing Interests

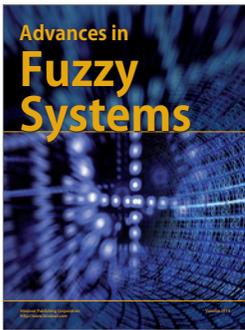
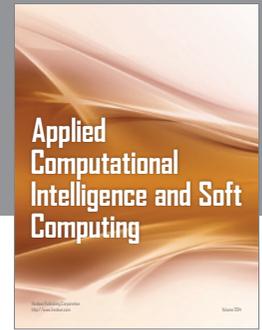
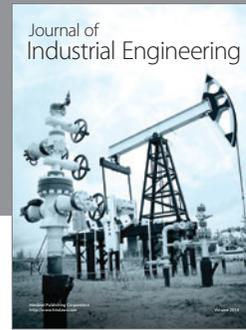
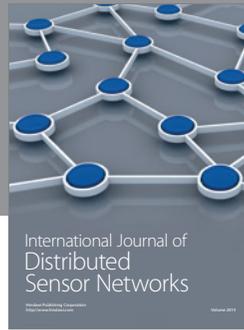
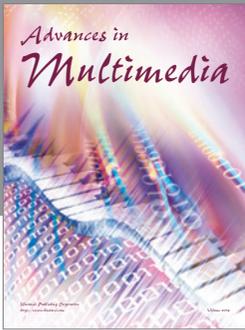
The authors declare that they have no competing interests.

## Acknowledgments

The authors would like to thank the Hebei Province Science and Technology Support Program (14275601D) and the Hebei Education Department Self-Financing Program (ZC2016124).

## References

- [1] J. Ning, J. Yang, S. Jiang, L. Zhang, and M. Yang, "Object tracking via dual linear structured SVM and explicit feature map," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*, pp. 4266–4274, Las Vegas, Nev, USA, June 2016.
- [2] H. Li, C. Shen, and Q. Shi, "Real-time visual tracking using compressive sensing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 1305–1312, IEEE, Providence, RI, USA, June 2011.
- [3] K. Zhang, L. Zhang, and M. Yang, "Real-time compressive tracking," in *Computer Vision—ECCV 2012*, vol. 7574 of *Lecture Notes in Computer Science*, pp. 864–877, Springer, Berlin, Germany, 2012.
- [4] Y. Motai, S. Kumar Jha, and D. Kruse, "Human tracking from a mobile agent: optical flow and Kalman filter arbitration," *Signal Processing: Image Communication*, vol. 27, no. 1, pp. 83–95, 2012.
- [5] A. H. Mazinan and A. Amir-Latifi, "Applying mean shift, motion information and Kalman filtering approaches to object tracking," *ISA Transactions*, vol. 51, no. 3, pp. 485–497, 2012.
- [6] H. Cheng, Z. Liu, L. Yang, and X. Chen, "Sparse representation and learning in visual recognition: theory and applications," *Signal Processing*, vol. 93, no. 6, pp. 1408–1425, 2013.
- [7] S. Zhang, H. Yao, H. Zhou, X. Sun, and S. Liu, "Robust visual tracking based on online learning sparse representation," *Neurocomputing*, vol. 100, pp. 31–40, 2013.
- [8] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.
- [9] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: structured output tracking with kernels," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 263–270, Barcelona, Spain, November 2011.
- [10] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.
- [11] L. Wen, Z. Cai, Z. Lei, and S. Li, "Online spatio-temporal structural context learning for visual tracking," in *Proceedings of the 12th European Conference on Computer Vision (ECCV '12)*, Florence, Italy, October 2012.
- [12] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. Hengel, "Part-based visual tracking with online latent structural learning," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2363–2370, IEEE, Portland, Ore, USA, June 2013.
- [13] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proceedings of the 7th British Machine Vision Conference (BMVC '06)*, pp. 47–56, September 2006.
- [14] C. Xu, W. Tao, Z. Meng, and Z. Feng, "Robust visual tracking via online multiple instance learning with Fisher information," *Pattern Recognition*, vol. 48, no. 12, pp. 3917–3926, 2015.
- [15] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised online boosting for robust tracking," in *Computer Vision—ECCV 2008*, pp. 234–247, Springer, 2008.
- [16] K. Zhang and H. Song, "Real-time visual tracking via online weighted multiple instance learning," *Pattern Recognition*, vol. 46, no. 1, pp. 397–411, 2013.
- [17] T. Zhou, Y. Lu, and M. Qiu, "Online visual tracking using multiple instance learning with instance significance estimation," <http://arxiv.org/abs/1501.04378v1>.
- [18] L. J. Wang and H. Zhang, "Visual tracking based on an improved online multiple instance learning algorithm," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 3472184, 9 pages, 2016.
- [19] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 798–805, IEEE, New York, NY, USA, June 2006.
- [20] Z. Wang, S. Yoon, S. J. Xie, Y. Lu, and D. S. Park, "Visual tracking with semi-supervised online weighted multiple instance learning," *The Visual Computer*, vol. 32, no. 3, pp. 307–320, 2016.
- [21] T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja, "Robust visual tracking via exclusive context modeling," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 51–63, 2016.



# Hindawi

Submit your manuscripts at  
<https://www.hindawi.com>

