J Inf Process Syst, Vol.14, No.6, pp.1307~1317, December 2018
https://doi.org/10.3745/JIPS.02.0097

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Object Tracking with the Multi-Templates Regression Model Based MS Algorithm

Hua Zhang* and Lijia Wang**

Abstract

To deal with the problems of occlusion, pose variations and illumination changes in the object tracking system, a regression model weighted multi-templates mean-shift (MS) algorithm is proposed in this paper. Target templates and occlusion templates are extracted to compose a multi-templates set. Then, the MS algorithm is applied to the multi-templates set for obtaining the candidate areas. Moreover, a regression model is trained to estimate the Bhattacharyya coefficients between the templates and candidate areas. Finally, the geometric center of the tracked areas is considered as the object's position. The proposed algorithm is evaluated on several classical videos. The experimental results show that the regression model weighted multi-templates MS algorithm can track an object accurately in terms of occlusion, illumination changes and pose variations.

Keywords

Mean Shift Algorithm, Multi-Templates, Object tracking, Regression Model

1. Introduction

Object tracking plays an important role in computer vision, such as surveillance, robotics, human computer interaction, etc. In the past decade, many successful algorithms have been proposed for robust object tracking in the complex environment [1-4]. However, the object tracking is still a challenging task due to appearance variations caused by occlusion, pose variations, abrupt motion, and illumination variations.

In general, object tracking algorithms can be classified into two groups: the generative methods and the discriminative methods. The generative methods model an object in the first frame and search for the area with the most similar appearance as the result [5]. The generative methods include MS tracker [6], fragments based tracker [7], incremental tracker (IVT) [8], and visual tracking decomposition (VTD) [9]. The mean-shift (MS) tracker often represents an object with color histograms and determines the tracking result with the highest matching score by using the iterative method [6]. The fragments based tracker models an object by using multiple image fragments or patches and determines the tracking result by combining patches instead of the whole object [7,10]. The IVT tracker learns and

Manuscript received April 18, 2017; first revision June 16, 2017; accepted June 21, 2017.

Corresponding Author: Lijia Wang (wanglijia1981@hotmail.com)

* College of Electrical and Electronic Engineering, Shijiazhuang University of Applied Technology, Shijiazhuang, China (zhanghua_2514@163.com)

** Dept. of Intelligent Manufacture, Hebei College of Industry and Technology, Shijiazhuang, China (wanglijia1981@hotmail.com)

updates a low dimensional eigenspace representation to reflect an object's appearance changes [8]. It has demonstrated that these methods perform well when there are appearance changes caused by lighting and pose variations. However, they are less effective in dealing with the problems of heavy occlusion, serious lighting and pose variations [11]. VTD tracker addresses the problems of occlusion and appearance changes by decomposing the observation model into multiple basic observation models. In the VTD tracker, each basic observation model covers a specific appearance of an object. However, it suffers from expensive computation because it is realized through an interactive Markov Chain Monte Carlo (IMCMC) framework.

The discriminative methods formulate object tracking as a binary classification problem to distinguish an object from background. The online-boosting algorithm selects discriminative features for object tracking [12]. However, it often suffers from tracking drift because only one positive sample (the tracking result) is used for classifier updating [5]. Then, Grabner et al. [13] proposed a semi-supervised object tracking algorithm which labels the positive samples in the first frame. However, it suffers from ambiguity as tracking evolves. The multiple instance learning (MIL) algorithm was proposed to learn a strong classifier from multiple instances in the positive and negative bags [14]. Although the algorithm overcomes the problem of ambiguity in object tracking, it does not handle the problem of large non-rigid shape deformation [15,16].

Recently, sparse representation based object tracking algorithms have been studied [17-20]. The tracker represents [17] an object as a sparse linear combination of the object templates and trivial templates. The templates are useful for handling the problems of occlusion and appearance changes. The tracking results show that the algorithm can achieve good performance but with expensive computation.

In this paper, a regression model based MS algorithm is proposed for object tracking. In the algorithm, multiple templates are captured to deal with the problems of occlusion, lighting changes, and pose variations. These templates are divided into two classes. The first candidate templates are captured from the object, the object's transformational versions, and the occlusion templates. The second candidate templates are generated when tracking fails. The color-texture descriptor is used to represent these templates. Furthermore, to implement real-time and robust object tracking, a regression model is trained by using these templates. Then, the regression model estimates the values of the Bhattacharyya coefficients of the candidate areas for determining the object's location. At last, the proposed algorithm is evaluated on several videos. The experimental results show that the algorithm is robust to occlusion, illumination variations, and poses changes.

2. Overview of the System

The overview of the object tracking algorithm is shown in Fig. 1. To handle the problems of illumination changes, pose variations, and occlusion, the algorithm employs multiple templates including the object, the object's transformational versions, the illumination ones, and the occlusion templates. In the beginning of the tracking process, the MS algorithm is applied to these templates. Consequently, several candidate areas are obtained. The tracking result is determined by the geometric center of the candidate areas with larger similar scores (Bhattacharyya coefficient is used to measure the similarity). As tracking evolves, the templates are inserted into a training pool. As the number of the templates in the pool achieves a given value, a regression model is trained with these templates. Then, the regression model estimates the values of the Bhattacharyya coefficients between these templates and the candidate areas. At last, the object's location is determined by weighting the candidate areas which have larger similar scores. The contributions of this paper are as follows.

- Multiple templates are proposed to handle the problems of occlusion, illumination changes, and pose variations. The multi-templates are captured from the object, the object's transformational versions, the illumination models and occlusion templates.
- A regression model is presented to estimate the similarity of the object and candidate area. The more the number of the templates is, the better the algorithm performs. However, as the number of the templates increases, the algorithm will suffer from heavy computing time. The presented regression model trains the multi-templates to control the number of the templates.
- Regression model based MS algorithm is applied on the templates. The MS algorithm is applied to the obtained templates for detecting the candidate areas. Then, the regression model estimates the similar values of the candidate areas. The tracking result is determined by weighting the areas' similar scores with the estimated values.

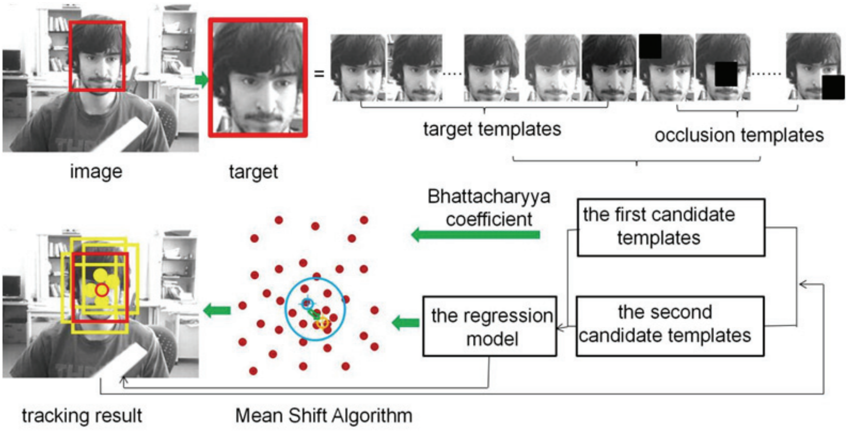


Fig. 1. The flow chart of the regression model based MS algorithm.

3. Multiple Templates

In many videos, objects are often occluded or corrupted by noise [21]. To address the problem, the context-aware exclusive sparse tracker (CEST) [11] employed a dictionary combining by three types of templates D_F, D_O and D_C. These templates incorporate information about the object, noise/occlusion, and context. It is shown that the algorithm performs well in challenging videos. Inspired by the CEST, a multiple templates based MS algorithm is proposed. In the algorithm, a final multi-templates set, which includes the first candidate templates and the second candidate templates, are defined. Then, the MS algorithm is applied on the final templates set for tracking the areas with larger similar scores.

The first candidate templates contain information about the object, noise, and occlusion. They are defined as $T_{1i}\{i=1,2,3\}$. The first group includes the target templates which are cropped from the tracking result and the transformed versions of the result. The transformational versions of the result are robust to the pose variations. The second group constituted by occlusion templates is shown in Fig. 1. The non-zero entries in the templates indicate that the pixels are occluded [11]. The last group includes the lighting templates.

The second candidate templates are obtained when tracking fails. They are indicated as $T_{2i}\{i=1,2,\cdots,n\}$. Then, the final templates $T_i\{i=1,2,\cdots,M\}$ $M=m+n$ are composed by the first and second candidate templates. The process for obtaining the second candidate templates and the final templates are detailed as follows.

In the beginning of the tracking process, the MS algorithm is applied to the first candidate templates. As a result, the candidate areas are obtained. The Bhattacharyya coefficient is used to measure the similarity of the candidate areas. If there are the areas with the similar scores larger than a given threshold ($\hat{\rho}(y_i)>\gamma$), the tracking results are determined by these areas' geometric center $L_{cen}=\frac{1}{N_{\mathrm{c}}}\sum_{i=1}^{N_c}L_i$ where N_{c} is the number of the selected areas and L_i is an area's center location. Furthermore, we re-extract samples for the first candidate templates based on the tracking result. Once all the candidate areas' similar scores are less than the given threshold, the MS based tracking algorithm fails. In such a case, the EKF [10] predicts the object's location. Meanwhile, the areas with larger similar scores are selected as the second candidate templates. At last, the final templates are obtained by the first candidate templates and the second candidate templates.

After obtaining the final templates, the MS algorithm is employed in the successive frames. Then, *m*+*n* candidate areas are obtained. If there are the areas with the similar scores larger than the given threshold, the object location will be determined by these areas. Meanwhile, the first candidate templates are cropped around the tracking result, while the second candidate templates are obtained as the step mentioned above. If tracking fails, the first candidate templates remain unchanged and the second candidate templates are updated with the *n* areas which have larger similar scores. In such a case, the EKF [10] predicts the object's location.

4. Regression Model

Multi-templates based object tracking algorithm is useful to deal with the problems of illumination changes and pose variations. The more the number of the templates is, the better the tracking algorithm performs. However, the computing time will increase as the number of the templates grows. To deal with the problem, multi-templates based regression model is proposed. The goal of the regression model is to estimate a candidate area's similar score.

4.1 Color-Texture Descriptor

Color feature is often used in computer vision because it is insensitive to rotation, translation, and scale. However, it ignores the spatial information [22]. The texture feature reflects the spatial distribution of the pixel's pray and makes up the shortcoming of the color feature [23]. Therefore, color-texture descriptor is used to all-sided represent an object. We use the HUE component in the HSV space as the color feature, while the uniform LBP feature is employed due to its lower computational complexity, scale invariability, and rotation invariability [24,25]. The color-texture descriptor is obtained as follows:

$$\hat{q}_{f,u} = C\sum_{i=1}^{M} k\left(\left\|\frac{x_i - x_0}{h}\right\|\right)\delta\left(b(x_i) - u\right) \ , \tag{1}$$

where $f = Co, Te$ indicates the color and texture information, respectively. $\hat{q}_{f,u}$ is the corresponding histogram. C is the normalization constant which guarantees $\sum_{u=1}^{M}\hat{q}_{f,\ u} = 1$. $\delta(\cdot)$ is Delta function, which determines whether the value of x_i belongs to the u^{th} bin. $k(\cdot)$ is the Epanechnikov kernel function. h is the bandwidth of the kernel function. x_0 is the center of the object area.

In the tracking process, the candidate area centered in y is expressed with its pixels $\{x_i{*}\}, i = 1,2,...,n$. Then, a candidate area's color-texture descriptor $\hat{p}_{f,u}$ is obtained as follows:

$$\hat{p}_{f,u} = C_h\sum_{i=1}^{M} k\left(\left\|\frac{x_i^* - y}{h}\right\|\right)\delta\left(b(x_i^*) - u\right) \ . \tag{2}$$

To measure the similarity between the candidate area and the template, the Bhattacharyya coefficient is used:

$$\hat{\rho}_f = \hat{\rho}_{Co}\hat{\rho}_{Te} = \sum_{u=1}^{M}\sqrt{\hat{p}_{Co,u}\hat{q}_{Co,u}}\sum_{u=1}^{M}\sqrt{\hat{p}_{Te,u}\hat{q}_{Te,u}} \ , \tag{3}$$

where $\hat{p}_{Co,u}$ and $\hat{p}_{Te,u}$ are the color histogram and texture histogram of the template, respectively. $\hat{q}_{Co,u}$ and $\hat{q}_{Te,u}$ are the color histogram and texture histogram of the candidate area, respectively. $\hat{\rho}_{Co}$ and $\hat{\rho}_{Te}$ are the color similarity and texture similarity, respectively. $\hat{\rho}_f$ is the obtained similarity.

4.2 Regression Model

In the multi-templates based MS algorithm, the number of the templates affects the algorithm's performance. The algorithm is more robust to the problem of occlusion, illumination variations and pose changes as the number of the templates increases. However, it means that the MS algorithm will be applied on more templates for generating the accurate result. As a result, the computing time of dealing with one frame will increase. To address the problem, a regression model is presented:

$$R(\hat{p}(y)) = \beta^T\hat{p}(y) \ , \tag{4}$$

where $R(\cdot)$ is the estimated value of the Bhattacharyya coefficient $\hat{\rho}_f$ for a candidate area. $\hat{p}(y) = (\hat{p}_{f0}(y), \hat{p}_{f1}(y), \cdots, \hat{p}_{fm}(y))^T$ is a vector with the color-texture descriptor as the elements and $\hat{p}_{f0}(y) = 1$. $\beta = (\beta_{f0}, \beta_{f1}, \cdots, \beta_{fm})^T$ is a template's parameter.

The regression model is trained online by using the final templates and the candidate areas. The goal of training is to estimate the parameter β which guarantees the loss function $L(\beta)$ be the minimum value:

$$\min_{\beta}\left(L(\beta)\right)=\min_{\beta}\left(\sum_{i=1}^{n}\left[\hat{\rho}(y_i)-R\left(\hat{p}(y_i)\right)\right]^2\right) , \tag{5}$$

where $\hat{\rho}(y_i)$ is the similar score between the i^{th} candidate area and the corresponding template.

The closed solution of the Eq. (5) can be obtained by using the algebra method:

$$\hat{\beta}=\left(P^T P\right)^{-1}P^T\hat{\rho} , \tag{6}$$

where $\hat{\beta}$ is the estimated value of the parameter. P^T is a matrix with dimension $(m+1)\times n$. P^T is composed by n samples' color-texture descriptors: $P^T=(\hat{p}(y_1),\hat{p}(y_2),\cdots,\hat{p}(y_n))$. $\hat{\rho}$ is a vector composed by the Bhattacharyya coefficients of n candidate areas:

$$\hat{\rho}=(\hat{\rho}(\mathrm{y}_1),\hat{\rho}(\mathrm{y}_2),\ldots,\hat{\rho}(\mathrm{y}_n))^T . \tag{7}$$

4.3 Regression Model based MS Algorithm

To implement robust and real-time object tracking, the regression model based MS algorithm is used. In the beginning, the MS algorithm is applied on the final templates for detecting an object. In such a case, the Bhattacharyya coefficients are employed to measure the similarity between the tracked candidate areas and the templates. The MS algorithm is implemented as follows:

Step 1: Initialize the searching area and center manually in the first frame;

Step 2: Compute the color-texture descriptor according to the Eq. (1);

Step 3: Take the searching area and center in the previous frame as the initialization values in the current frame;

Step 4: Apply the MS algorithm and compute the searching area as follows:

$$y_1=\sum_{i=1}^{n_h}x_i w_i g\left(\left\|\frac{y-x_i}{h}\right\|^2\right)\Big/\sum_{i=1}^{n_h}w_i g\left(\left\|\frac{y-x_i}{h}\right\|^2\right) , \tag{8}$$

where $g(x)=-k_E^{'}(x)$ is the Epanechnikov function. w_i is the weight for the pixels in the object area and it is calculated as follows:

$$w_i=\sum_{u=1}^{m}\sqrt{\frac{q_u}{p_u(y)}}\delta[b(x_i)-u] . \tag{9}$$

Step 5: If $\|y_1<y\|<\varepsilon$, then stop searching and obtain the final tracking position. Else, let $y=y_1$ and return to the step 4.

After detecting the object by using the MS algorithm, the final templates are inserted into a training pool. Once the number of the training samples reaches a value, the regression model is trained. Then, the estimated value generated by the regression model is used to measure the candidate areas' similar score instead of the Bhattacharyya coefficients. The areas with the estimated values larger than a given threshold are weighted by their values to determine the object's location:

$$L_{cen} = \frac{1}{N_{\mathrm{c}}} \sum_{i=1}^{N_c} R_i L_i \quad , \tag{10}$$

where L_{cen} is the final position of an object. L_i is the position of the detected area with the estimated value larger than the given threshold. N_{c} is the number of the detected areas with the estimated value larger than the given threshold.

5. Results

Experimental results are detailed in the section to validate the performance of our algorithm.

5.1 Data-Sets

To evaluate the regression model based MS algorithm, a set of challenging videos which are publicly available online are used [9]. The videos are "David indoor", "Occluded face", "Coke can", "Cliffer bar", "Tiger", and "Can". There are serious occlusion, illumination changes, and pose variations in these videos. Furthermore, our regression model based MS algorithm is compared to other state-of-the-art trackers: MIL [15], VTD [9], compressive tracker (CT) [5]. These algorithms are realized by using the source codes available online. The default parameters of the MIL [15], VTD [9], CT [5] algorithms are used. For our algorithm, the number of the final templates is 3, while that of the candidate templates is 2. In the regression model, the dimension of the color feature is 48, while dimension of the texture feature is 36.

5.2 Tracking Results

The tracking position is used to evaluate the performance of the proposed algorithm. The tracking results are shown in Fig. 2. The tracked object is indicated by a rectangle. The red rectangles are for the proposed algorithm. The purple ones are for the MIL algorithm. The results of the VTD algorithm are indicated by the green rectangles. The results of the CT algorithm are indicated by the blue rectangles.

The results of the "David indoor" sequences are shown in the first line. There are serious illumination change and pose variation in the frame "128" and "202". These tracking algorithms can detect the object in the frame "128". However, as tracking evolves, the MIL, CT, VTD trackers drift. The tracking results show that the proposed algorithm is robust to the illumination variations and pose changes. The results in the second line are for the "Occluded face" sequence. The "face" is occluded by a book or a hat from the frame "495". The proposed regression model based MS algorithm tracks the "face" by using multiple templates and updates these templates in different cases. Therefore, the algorithm can deal with the serious occlusion. The "Cliffer bar" moves fast and rotates in the tracking process. The proposed algorithm performs well over other algorithms. Especially, in the frame "412", only the proposed algorithm detects the object. The objects, in the "Coke can" and "Tiger" sequences, move fast and are often occluded by other objects. The tracking results show that the proposed algorithm tracks the object successfully. In the last line, the "dollar" is interfered by the similar background. All the trackers can detect the "dollar", but the MIL, CT, VTD trackers often drift away.

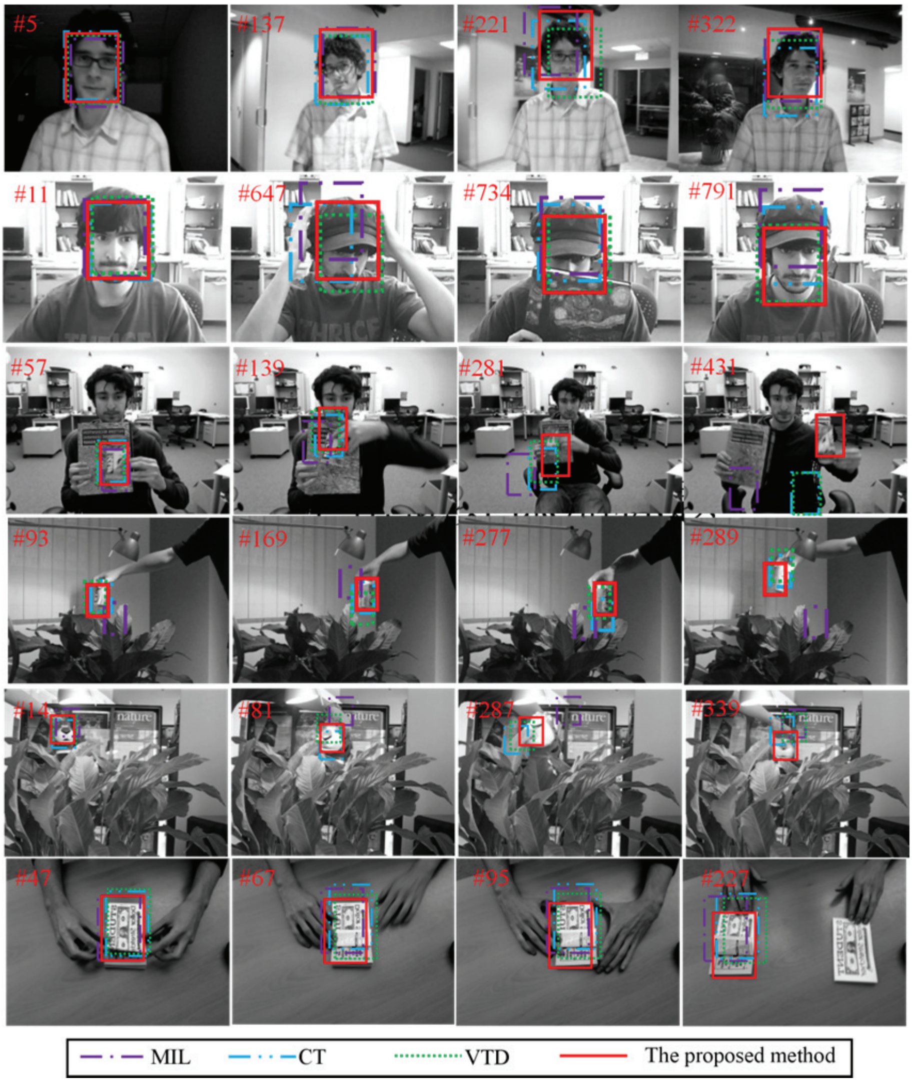


Fig. 2. The tracking results obtained by using the MIL, CT, VTD, and the proposed method.

The experimental results show that the proposed algorithm is robust to the illumination (e.g., the "face" in the "David indoor" video), serious occlusion (e.g., the "face" in the "Occluded face" video), pose variation (e.g., the "Can" in the "Coke can" video), and scale changes (e.g., the object in the "Cliffer bar" video).

5.3 Computational Cost

The proposed tracker has been demonstrated to perform well in visual tracking. Then, the algorithm is evaluated in term of computational cost. Table 1 shows the average per-frame computational cost of these algorithms.

The MIL tracker suffers from a heavy computational cost, because it computes the bag probability and instance probability *M* times when a powerful classifier is selected. The CT algorithm is a fast tracker which processes a frame with the least time. However, the CT tracker often fails in the case of serious illumination changes and poses variations. The VTD realizes the tracking algorithm in the framework of IMCMC. As a result, it suffers from a long computational time. Applying the MS algorithm to multiple templates will result in heavy computational cost. To decrease the average per-frame computational cost, the regression model is proposed in our algorithm. The experimental results show that the computing time of our method is lower than that of the VTD and MIL algorithms. Meanwhile, our method runs slower than the CT algorithm, but performs better especially when there are serious scale changes (e.g., in the "Cliffer bar" video). Overall, the presented algorithm can successfully track the object in real time.

Table 1. The average per-frame computational cost (in seconds) of the trackers MIL, CT, VTD, and the proposed tracker

Video clip	MIL	CT	VTD	Proposed tracker
David indoor	1.122	0.073	1.01	0.68
Occluded face2	1.601	0.071	0.904	0.71
Coke can	1.089	0.071	0.964	0.124
Cliff bar	1.092	0.068	0.98	0.179
Tiger2	1.092	0.073	1.003	0.194
Coupon book	1.164	0.071	1.103	0.46

6. Conclusion

In this paper, a regression model based MS algorithm is proposed for object tracking. First, multiple templates are extracted to deal with the problems of occlusion, illumination changes and pose variations. Second, a regression model is presented to estimate the similar score between the candidate area and template. The method decreases the number of templates and guarantees good performance. Third, the MS algorithm is applied to the templates, and several candidate areas with larger estimated similar scores are obtained. Then, the obtained candidate areas are weighted by the estimated similar scores for detecting the object. The experimental results have shown that the algorithm performs well in terms of illumination changes, pose variations and occlusion.

Acknowledgement

The authors would like to thank the key research and development program of the Hebei Province science (No. 18210329D) and the Natural Science Foundation of Hebei Province (No. F2018205178).

References

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Computing Surveys*, vol. 38, no. 4, pp. 1-45, 2006.

[2] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with Gaussian processes regression," in *Computer Vision-ECCV 2014*. Cham: Springer, 2014, pp. 188-203.

[3] J. Ning, J. Yang, S. Jiang, L. Zhang, and M. H. 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 2016)*, Las Vegas, NV, 2016, pp. 4266-4274.

[4] X. Mei and H. Ling, “Robust visual tracking and vehicle classification via sparse representation,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 33, no. 11, pp. 2259-2272, 2011.

[5] K. Zhang, L. Zhang, and M. H. Yang, “Real-time compressive tracking,” in *Computer Vision-ECCV 2012*. Heidelberg: Springer, 2012, pp. 864-877.

[6] F. Dadgostar, A. Sarrafzadeh, and S. P. Overmyer, “Face tracking using mean-shift algorithm: a fuzzy approach for boundary detection,” in *Affective Computing and Intelligent Interaction*. Heidelberg: Springer, 2005, pp. 56-63.

[7] A. Adam, E. Rivlin, and I. Shimshoni, “Robust fragments-based tracking using the integral histogram,” in *Proceedings of IEEE Computer Society Conference on Computer Vision & Pattern Recognition*, New York, NY, 2006, pp. 798-805.

[8] D. A. Ross, J. Lim, R. S. Lin, and M. H. Yang, “Incremental learning for robust visual tracking,” *International Journal of Computer Vision*, vol. 77, no. 1, pp. 125-141, 2008.

[9] J. Kwon and K. M. Lee, “Visual tracking decomposition,” in *Proceedings of IEEE Computer Society Conference on Computer Vision & Pattern Recognition*, San Francisco, CA, 2010, pp. 1269-1276.

[10] S. M. Jia, L. J. Wang, X. Z. Li, and L. F. Wen, “Person tracking system by fusing multicues based on patches,” *Journal of Sensors*, vol. 2015, article ID. 760435, 2015.

[11] 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.

[12] H. Grabner, M. Grabner, and H. Bischof, “Real-time tracking via online boosting,” in *Proceedings of the British Machine Vision Conference*, Edinburgh, UK, 2006, pp. 47-56.

[13] H. Grabner, C. Leistner, and H. Bischof, “Semi-supervised online boosting for robust tracking,” in *Computer Vision-ECCV 2008*. Heidelberg: Springer, 2008, pp. 234-247.

[14] C. Zhang, J. C. Platt, and P. A. Viola, “Multiple instance boosting for object detection,” *Advances in Neural Information Processing Systems*, vol. 18, pp. 1417-1424, 2006.

[15] 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.

[16] L. J. Wang and H. Zhang, “Visual tracking based on an improved online multiple instance learning algorithm,” *Computational Intelligence & Neuroscience*, vol. 2006, article no. 12, 2016.

[17] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210-227, 2009.

[18] 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.

[19] F. Chen, Q. Wang, S. Wang, W. Zhang, and W. Xu, “Object tracking via appearance modeling and sparse representation,” *Image and Vision Computing*, vol. 29, no. 11, pp. 787-796, 2011.

[20] 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.

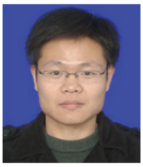
[21] P. Liang, Y. Pang, C. Liao, X. Mei, and H. Ling, “Adaptive objectness for object tracking,” *IEEE Signal Processing Letters*, vol. 23, no. 7, pp. 949-953, 2016.

[22] S. M. Jia, S. H. Wang, L. J. Wang, and X. Z. Li, “Human tracking based on multi-feature for intelligent robot under the CTF location strategy,” *Journal of Shanghai Jiaotong University*, vol. 48, no. 7, pp. 1039-1052, 2014.

[23] L. J. Wang, S. M. Jia, X. Z. Li, and Y. B. Lu, “Person tracking for robot using patches-based-multi-cues representation,” *Control and Decision*, vol. 31, no. 2, pp. 337-342, 2016.

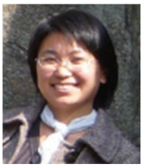
[24] G. B. Li and H. F. Wu, “Weighted fragments-based mean-shift tracking using color-texture histogram,” *Journal of Computer-Aided Design & Computer Graphics*, vol. 23, no. 11, pp. 2059-2066, 2011.

[25] J. Yang, Z. S. Gao, H. Z. Yuan, J. Yu, X. Q. Zhang, and C. Y. Liu, “Single sample face recognition based on LBP feature and Bayes model,” *Journal of Optoelectronics Laser*, vol. 22, no. 5, pp. 763-765, 2011.



Hua Zhang https://orcid.org/0000-0001-9826-9929

He received M.S. degrees in College of the Electrical Engineering & Automation from Beijing University of Technology in 2009. Since September 2009, he is with Department of Electrical and Electronic Engineering from Shijiazhuang University of Applied Technology as a lecturer.



Lijia Wang https://orcid.org/0000-0002-1907-171X

She received M.S. degree in College of the Electrical Engineering & Automation from Zhengzhou University in 2008. She is currently an associate professor of Hebei College of Industry and Technology. Her current research interests include machine vision and pattern recognition.