Computer & Robotics

Visual Tracking using Learning Histogram of Oriented Gradients by SVM on Mobile Robot

Iman Zabbah [a,*], Shima Foolad [b], Ali Maroosi [c], Alireza Pourreza [b]

[a] Department of computer, Torbat-e-Heydariyeh branch, Islamic Azad University, Torbat-e-Heydariyeh, Iran
[b] Department of Computer, Semnan University, Semnan, Iran
[c] Department of Computer, Torbat Heydariyeh University, Torbat Heydariyeh, Iran

Received 20 June 2018; Revised 25 July 2018; Accepted 29 August 2018; Available online 18 September 2018

Abstract

The intelligence of a mobile robot is highly dependent on its vision. The main objective of an intelligent mobile robot is in its ability to the online image processing, object detection, and especially visual tracking which is a complex task in stochastic environments. Tracking algorithms suffer from sequence challenges such as illumination variation, occlusion, and background clutter, so an accurate tracker should employ the appropriate visual features to identify target. In this paper, we propose using the histogram of oriented gradient (HOG), as an important descriptor. The descriptor simulates the performance of the complex cells in the primary visual cortex (V1) and it has low sensitivity to the illumination changes. In the proposed method, firstly, an object model is generated by training the HOG of multi first frames via an SVM classifier. Then, in order to track a new frame, the HOG descriptors are extracted from the surrounding areas of the target in the previous frame and convolved with the object model. Finally, the location with the highest score is defined as the target. The experimental results demonstrate the proposed method has significant performance compare to the state-of-the-art methods. Furthermore, we apply our algorithm to the mobile robot built by the robotics team to ensure its performance in a real environment.

Keywords: *Histogram of Oriented Gradient, Support Vector Machine, Object Model, Mobile Robot, Target Tracking, Visual Tracking.*

1. Introduction

An intelligent mobile robot is an automatic multi-purpose machine that acts like human activities, even against unknown environments. The robot should simulate human prominent traits in its behavior, moving, intelligence and communication [1], so that it can play the role of a human partner such as servant robots, nursing robots, submarine robots, and vacuum cleaner robots. In the first step, the function of such robots depends on the hardware equipment, for instance, sensors, to recognize the environment. Next, they employ image processing algorithms to accurately detect a target. In dynamic environments, directing and controlling robot are both very complex, due to the target similarity with the background objects in terms of color and shape. The purpose of directing is to create an ability through which a robot can coordinate itself with the environment, for instance, recognition of a particular object and tracking it in an uncertain environment. Visual tracking has an important role in some machine vision applications such as human-computer interaction and video surveillance. It

* Corresponding author. Email: imanzabbah@gmail.com

deals with challenges such as occlusion, deformation and illumination change and background clutter [2]. Figure 1 shows the result of proposed method on challenging sequences.

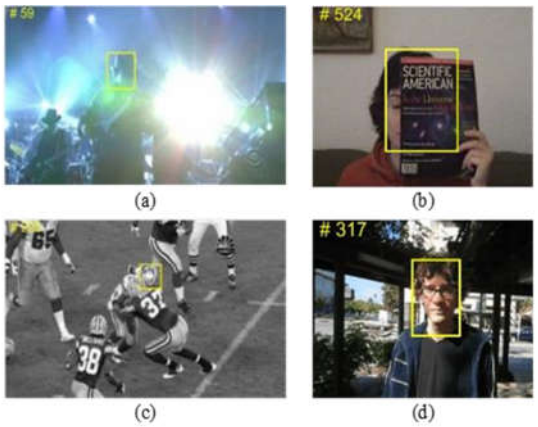

Fig. 1. Tracking results of the proposed method on challenging sequences. challenging factors include (a) illumination variation, (b) occlusion, (c) background clutter and (d) apperance change.

An accurate tracker should use the suitable visual descriptors which are not sensitive to the illumination, occlusion, clutter, and etc. Some trackers [3]–[6] apply color histogram that represents a distribution of colors by counting the number of pixels in each of given color ranges. BKG [3] method utilized color histogram in hue, saturation, and value (HSV) color space to calculate the background confidence map. Also, SPT [4] method used the histogram in the HSV color space for each superpixel of the sequence. Zoidi et al. [5] discarded candidate objects by color histogram similarity to the target object, While ACT [7] and FRT [6] methods employed intensity histogram which represents a histogram of the pixel intensity values in grayscale. ACT algorithm applied gray level histogram of visual parts to describe the target's local appearance. Using histogram will cause the spatial information to be ignored, whereas the information is required for tracking algorithms. In addition, other descriptors have been adopted in tracking methods such as speeded up robust features (SURF) in FBT [8] method, Haar gradients in MIT [9] algorithm, optical flow in PROST [10] model and covariance descriptor in [11]. SURF descriptor is achieved by detecting interest points by Hessian matrix approximation and constructing square regions around them using sum of Haar wavelet responses. Optical flow estimate motion of target object by computing brightness variations between frames. Recently, the histogram of oriented gradients (HOG) is added making more appropriate descriptors. Henriques et al. [12] proposed kernel correlation filter (KCF) and dual correlation filter (DCF) trackers on HOG descriptor. The HOG works like complex cells in the primary visual cortex (V1) [13] and due to its low sensitivity to the illumination changes, it has been the mainstay of researchers. The HOG is an effective descriptor applied in some applications such as object detection [14], face recognition [15], [16] and pedestrian detection [17]. Mostly, the HOG descriptor is learned using a support vector machine (SVM) classifier that is named HOG+SVM.

Most tracking methods [5], [6], [18] are model-based that first create an object model using prior information of the object and then search the most similar image regions to the model. Zoidi et al. [5] generated an object appearance model by storing the transformation of the object image such as rotation and scaling as object instances. Frag [6] tracker represented template model by selecting multiple image fragments arbitrarily. DML tracking algorithm [18] utilized a collection of templates as a template library to capture the object's appearance. Then the template library is updated if its similarity with the new appearance of the current frame is less than the similarity obtained from the previous frame.

This paper presents a method that firstly an object model is created by learning the HOG descriptors extracted from a few first frames via the SVM. Then, in order to track the new frame, the HOG descriptors are extracted from surrounding areas of the bounding box of the previous frame and convolved with the object model. Finally, we measure this algorithm on our robot, assistant robot, which has significant results.

2. Robot Technical Features

A robot consists of different parts which are described here.

2.1. Mechanic Infrastructures

Mobile robots have different mechanic infrastructures such as Chassis and propulsion systems and auxiliary devices. Producing appropriately-featured motion is the most important responsibility of the mechanic section. Therefore, different mechanisms such as wheels or feet are used. Each of such mechanisms is selected based on needed power, and cost and appropriate mobile features and the surface where robot moves. Wheels are favoured due to ease, cheapness, compatibility with the environment, high energy efficiency and good power. There is a different configuration for wheeled- robots. The difference is in location and wheels' role. There are three kinds of wheels in wheeled-mobile robots. Moveable wheels, to which engine propulsion is connected. Steering wheels cause the robot rotation and freewheels which help the robot retain its static balance without any connection to any engine or operator. Three-wheel mobile robot with propulsion differential system is a mobile robot. Figure 2 illustrates the mechanic infrastructure of the mobile robot. This robot is equipped with two DC electric motors horizontally located on the chassis, connected to the robot rear wheels. A particular mechanism has been used guiding robot front wheels to left and right. Its motor system is similar to real machines. This robot chassis is made of rubber which has been selected due to being available and economical.

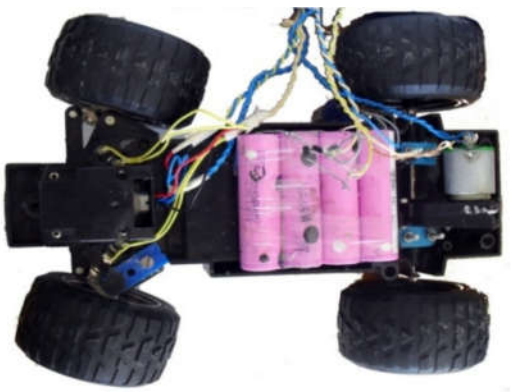

Fig. 2. Mechanic infrastructure of mobile robot.

2.2. Robot Electric and Electronic Infrastructure

This part includes processing units, interface circuit, DC Motor controllers in order to direct and control robot, and stepper motor controllers in order to control the camera. The mobile robot is connected to a notebook without a wire interface and only through the wireless network. Sending movies and receiving commands are occurred through the wireless gateway with bandwidth 2400 Hz. After receiving commands from the system, Interface circuit transmits them to DC motor drivers using micro 8051. Control system of DC motors is open- loop. This controller receives a command from an interface circuit and sends it to the motor drive. In order to ensure the accuracy of the commands received by the robot, all commands are displayed on Robot LCD.

2.3. Visual Unit

This robot is equipped with a CCD camera, IP camera, which transmits the received image through access wireless to the processor and it can be used for detecting the objects in a competition environment. The camera is controlled through two stepper motors with the capability of 270 rotations towards XY axis and 120 rotations towards Z axis. Since the circuit used for controlling motors is open, circuit for simulating motors needs to guarantee simulated motor coils in an appropriate order without any pulse. The object is detected through this camera after a series of processing operations, the minimal cadre surrounding the object is determined.

2.4. Communication Unit

Since this robot makes decisions individually and completely intelligently, it doesn’t require any specified communicational unit, but in order to be able to control and direct it in unavailable environments, a software interface has been developed so that manual communications are established if needed. Graphic interface and the robot appearance are shown in figure 3 and figure 4, respectively.

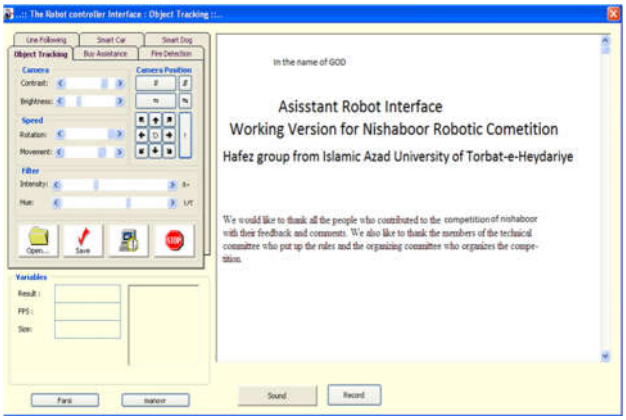

Fig. 3. Robot graphic interface.

Fig. 4. Robot appearance.

3. Proposed Method

Overview of the proposed method is illustrated in figure 5. First, an object model is learned by the k first frames which will be described in section A. Then, the target object of new frame is detected by the object model which will be described in section B.

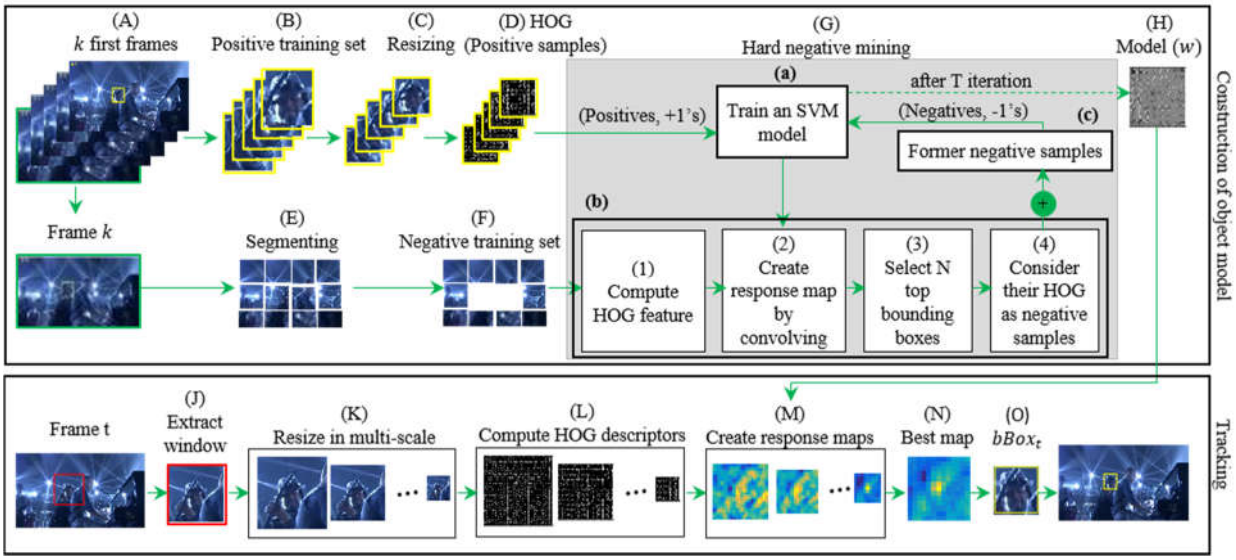

Fig. 5. Overview of the proposed method

3.1. Construction of Object Model

Given the bounding box of the target object in the first frame, the areas inside it in K first frames (figure 5-A) are assumed as the positive training set (figure 5-B). Each of the positive training instances is resized to 64×64 pixels (figure 5-C), and its HOG descriptor [19] is extracted (figure 5-D).

Figure 6 demonstrates the HOG calculation scheme. To compute HOG descriptor of an image, first, the gradient directions of the image is achieved (figure 6-b). Then, the image gradient is divided into small square areas termed cells (e.g. yellow squares shown in figure 6-b). Each cell is 4×4 as figure 6-d and a group of adjacent cells is called block (e.g. the block is 2×2 in figure 6-c). Afterward, for all pixels within each cell, a histogram of gradient directions are calculated (figure 6-e) and finally

the gathered histograms are normalized (figure 6-f). The HOG descriptor is considered as positive samples. On the other hand, the last frame (frame K) is segmented into patches 128×128 pixels (figure 5-E), and patches without target object are assumed as the negative training set (figure 5-F). Due to a large number of negative examples, we use hard negative mining [14] to find a set of key negative samples, (figure 2-G). This technique starts training a model without negative samples using 1-class SVM model [20]. The following steps are performed for T iterations:

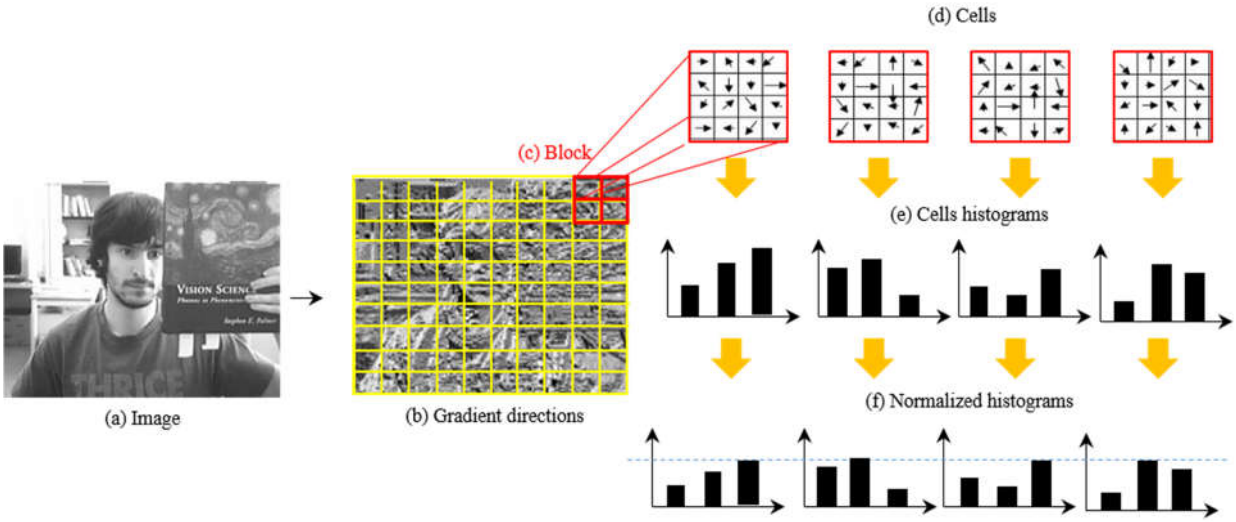

Fig. 6. Calculation scheme of the HOG descriptor. For a sample image (a), the gradient directions (b) is obtained and histogram of gradient directions (f) is computed after dividing the gradient into cells and blocks. In figure, blocks and cells are 2 × 2 and 4 × 4, respectively.

a) An SVM model is trained by the positive and negative samples and their labels (+1 for positives and -1 for negatives) as shown in figure 5-G (a). Then, an *Mw*×*Mh* template model is generated. In the first iteration, there are no negative samples.

b) Negative samples are extracted as shown in figure 5-G (b). In this way, each instance of the negative training set is resized in multi-scale and for each of them:

- The HOG is computed (figure 5-G (b.1)).
- It is convolved with the template model and a response map are achieved (figure 5-G (b.2)).
- Some bounding boxes are generated as follows:

$$\boldsymbol{bBox}_i^s = [\boldsymbol{x_i^s - 0.5}\,.\,\boldsymbol{y_i^s - 0.5}\,.\,\boldsymbol{x_i^s + HS \times Mw \times s - 0.5}\,.\,\boldsymbol{y_i^s + HS \times Mh \times s - 0.5}] \quad (1)$$

Where $bBox_i^s$ denotes the bounding box i of the resizing instance with scale s, HS indicates HOG cell size, and Mw and Mh refer the model width and height, respectively. Also, (x_i^s, y_i^s) indicates the instance pixel coordinates computed by units of HOG cell (hx_i^s, hy_i^s) as follows:

$$\boldsymbol{x_i^s = (hx_i^s - 1) \times HS \times s + 1} \quad (2)$$

$$\boldsymbol{y_i^s = (hy_i^s - 1) \times HS \times s + 1} \quad (3)$$

A $score_i^s$ is defined for each bounding box ($bBox_i^s$) that refer to the value of (hx_i^s, hy_i^s) position in the map. For each resized instance, 100 top bounding boxes are selected. Next, the non-maximum suppression algorithm is applied to ignore those overlapped bounding boxes greater than a threshold (here, 0.5). Then, 10 top bounding boxes are kept (figure 5-G (b.3)) to efficiency and their HOG are considered as negative samples (figure 5-G (b.4)).

c) The negative samples are concatenated into the former negative samples. Then, the duplicate samples are removed (figure 5-G (c)).

After applying hard negative mining technique for T iteration, the last template model is considered an object model (w) (figure 5-H).

3.2. Tracking

To determine the bounding box in the new frame(t), we first extract a square window centered at p_{t-1}^{c} with dimensions of $2 \times \max(w_{t-1}, h_{t-1})$ where w_{t-1}, h_{t-1} and p_{t-1}^{c} are width, height, and center position of the bounding box in frame $t-1$ respectively (figure 5-J). Second, we resize the window in multi-scale (figure 5-K) and compute the HOG for each of them (figure 5-L). Third, we convolve each HOG with the model w and obtain a response map (figure 5-M). Finally, from among the response maps, a point with the maximum value is chosen (figure 5-N) and a bounding box ($bBox_t$) is generated as Eq. 1 (figure 5-O). Since the $bBox_t$ is a square bounding box, the following conditions are applying to change its width or height:

$$\begin{aligned} \boldsymbol{h_t} &= \boldsymbol{w_t} \times \left({}^{\boldsymbol{h_{t-1}}}/_{\boldsymbol{w_{t-1}}}\right) \quad \boldsymbol{if\ w_{t-1} > h_{t-1}} \\ \boldsymbol{w_t} &= \boldsymbol{h_t} \times \left({}^{\boldsymbol{w_{t-1}}}/_{\boldsymbol{h_{t-1}}}\right) \quad \boldsymbol{otherwise} \end{aligned} \tag{4}$$

Where w_t and h_t denote width and height of the bounding box in frame t.

Logically, the size and position of $bBox_t$ are close to the bounding box in the previous frame ($bBox_{t-1}$). according to this principle, incorrect bounding boxes are replaced with $bBox_{t-1}$ as follows:

$$\begin{aligned} \boldsymbol{bBox_t} = \boldsymbol{bBox_{t-1}} \quad & \boldsymbol{if} \sqrt{(\boldsymbol{x_t} + \boldsymbol{x_{t-1}})^2 + (\boldsymbol{y_t} + \boldsymbol{y_{t-1}})^2} > \\ & \boldsymbol{or} \sqrt{(\boldsymbol{a_t} + \boldsymbol{a_{t-1}})^2} > \boldsymbol{th_{size}} \end{aligned} \tag{5}$$

Where

$bBox_t = [x_t, y_t, w_t, h_t]$,$bBox_{t-1} = [x_{t-1}, y_{t-1}, w_{t-1}, h_{t-1}]$.

$a_t = w_t \times h_t$ and $a_{t-1} = w_{t-1} \times h_{t-1}$

refer the areas of bounding boxes in the frame t and $t-1$, respectively. $th_{pos} = 0.25 \times$ window width and $th_{size} = 0.5 \times a_{t-1}$.

After creation of the bounding box in the frame t, the object model w is updated in two ways: (1), in each frame, the $bBox_t$ is resized to 64×64 pixels and its HOG is concatenated into positive samples. The model w is updated by training the SVM model. (2), after the U frame, the negative samples are changed by applying the hard negative mining technique, and the model w is updated by training the SVM model. The main steps of the proposed method are summarized in Algorithm 1.

Algorithm 1: main step of the proposed method

Construction of object model:

t=1 (frame)

for *k*=1 to ***K***

1) extract areas inside the target bounding box (in the first frame) from the frame ***k*** manually and resize it to 64×64

2) compute the HOG descriptor and consider it as a positive sample

end

Segment the frame ***K*** into patches 128×128, and consider the patches without the target object as the negative training set.

Apply the hard negative mining to extract the negative samples:

for *i*=1 to ***T***

1) generate a template model by training an SVM (if $\boldsymbol{i = 1}$, SVM is 1-class)

2) **for** each instance of the negative training set:

a) **for *s***=1 to ***S***

- resize it to scale $\frac{\mathbf{1}}{\boldsymbol{s}}$ and compute the HOG
- convolve with the template model and create a response map
- generate some bounding boxes by Eq. 1
- define a score for each bounding box

end

b) select 100 top bounding boxes

c) apply the non-maximum suppression algorithm

d) consider the HOG of 10 top bounding boxes as negative samples

end

3) Concatenate the obtained negative samples with the former negatives and eliminate the duplicate samples.

end

Consider the last template model as object model ($\boldsymbol{w}$)

Tracking:

for $\boldsymbol{t}$=2 to the end of the video

1) extract a window around the bounding box of the frame $\boldsymbol{t-1}$

2) Resize it into multi-scale and compute the HOG for each of them.

3) Convolve each HOG with the model $\boldsymbol{w}$ to obtain a response map.

4) choose $\boldsymbol{bBox_t}$ with maximum value in the maps

5) replace $\boldsymbol{bBox_t}$ with $\boldsymbol{bBox_{t-1}}$ according to Eq. 5

6) update the model $\boldsymbol{w}$

a) add the HOG of $\boldsymbol{bBox_t}$ to the positive samples for each frame

b) change the negative samples after $\boldsymbol{U}$ frame

end

4. Experimental Results

In this section, we discuss in detail the experiments carried out to evaluate the proposed method.

4.1. Experimental Setup

Our results have been achieved on 1.80 GHz Core i5 CPU with 6 GB RAM in MATLAB. We used The VLFeat [21] to compute the HOG and SVM. Furthermore, we applied SDCA Solver with parameters of $\lambda = 1/(number\ of\ samples)$ and $\varepsilon = 0.01$ in SVM.

In order to generate the object model, we used 5 first frames $(K = 5)$ in our experiments. Also, the object model has been set to 8×8 $(Mw = Mh = 8)$ dimensions. In hard negative mining technique, each instance of the negative training set was resized to 15 scales (0.5, 0.6, 0.74, 0.9, 1.1, 1.34, 1.64, 2, 2.43, 2.97, 3.62, 4.41, 5.38, 6.56, 8). The negative samples are changed at every 20 frames $(U = 20)$ in order to update the model w.

4.2. Quantitative Evaluation

For quantitative assessments, two measures of center location error and success rate are adopted. The first measure, center location error, can be computed by the Euclidean distance between the center of the tracked object and that of ground truth. The second measure, success rate, is calculated based on evaluation metrics of the PASCAL VOC challenge [22]. Given the ground truth bounding box $bBox_{GT}$ and the tracked bounding box $bBox_t$ of the frame t, the success rate (OR) is defined as follows:

$$\boldsymbol{OR} = \frac{\boldsymbol{area}\ (\boldsymbol{bBox_t} \cap \boldsymbol{bBox_{GT}})}{\boldsymbol{area}\ (\boldsymbol{bBox_t} \cup \boldsymbol{bBox_{GT}})} \quad (6)$$

The tracking result of one frame is successful when OR is above 0.5. To assess the performance of tracker using this measure, we compute the average success rate on the whole frames of each sequence.

We evaluate the proposed method against the state-of-the-art trackers including fragmented-based (Frag) method [6], online adaptive boosting (OAB) method [23], tracking by detection (TLD) method [24] and real-time compressive tracking (RTCT) method [25] on 5 sequences. The sequences of FaceOcc1, FaceOcc2 from [26], Trellis from [27], football from [28] and Shaking are used in the experiments.

To compare the proposed method with the state-of-the-art trackers, we exploit the mentioned measures. Table I and II show average success rate and average center location errors of trackers on the sequences, respectively. As can be seen, our method has obtained the best or second best results in more sequences. Although table I shows the Frag and OAB trackers on the FaceOcc1 sequence and the OAB and TLD methods on the FaceOcc2 sequence have achieved better results than our method, table II indicates the average center location errors of these trackers in the aforementioned sequences are more than our method. By considering the results average on the entire sequences, our method achieves lower average center location error (12.28) than other trackers. Furthermore, our method (0.47) and the Frag

tracker (0.50) attain higher average success rate than the others.

Table. 1. Average success rate of trackers on 5 sequences.

	Frag	OAB	TLD	RTCT	ours
football	0.59	0.23	**0.60**	0.02	**0.60**
Shaking	**0.41**	0.01	0.33	0.02	0.40
FaceOcc1	**0.87**	0.77	0.57	0.73	0.58
FaceOcc2	0.38	**0.59**	0.57	0.54	0.47
Trellis	0.29	**0.46**	0.21	0.22	0.32
average	**0.50**	0.41	0.45	0.30	0.47

For each sequence, bold text indicates the best result and underlined one indicates the second best.

Table. 2. Average center location errors of trackers on 5 sequences.

	Frag	OAB	TLD	RTCT	ours
football	6.3	53.3	**6.0**	123.3	6.3
Shaking	15.3	100.3	21.0	86.6	**14.7**
FaceOcc1	17.9	17.2	14.8	19.0	**12.3**
FaceOcc2	48.2	20.8	13.3	**6.0**	11.9
Trellis	55.7	41.5	50.9	42.4	**16.2**
average	28.6	46.6	21.2	55.4	**12.28**

For each sequence, bold text indicates the best result and underlined one indicates the second best.

4.3. Qualitative Evaluation

Figure 7 illustrates the tracking precision of our method against the other trackers on four sequences. The football sequence contains scenes with cluttered background and objects similar to the target object. The Frag, OAB and RTCT methods fail to track the target object at different frames as shown in figure 7a, while the proposed method tracks it successfully. In the Shaking sequence, the target appearance changes significantly due to illumination variation. Moreover, background clutter and partial occlusion are the challenging factors on this sequence, as shown in Figure 7b. Figure 7c presents some tracking results on the FaceOcc1 sequence that the target object is occluded by a book. When a heavy occlusion occurs in the sequence, the OAB and RTCT methods drift away from the target face, whereas the other methods track the target well. In addition, the FaceOcc2 sequence deals with the occlusion challenging and pose variation. The trellis sequence results in figure 7d illustrate the challenging factors such as illumination variation and changing the appearance of the target. Due to these challenges, some methods provide poor performance.

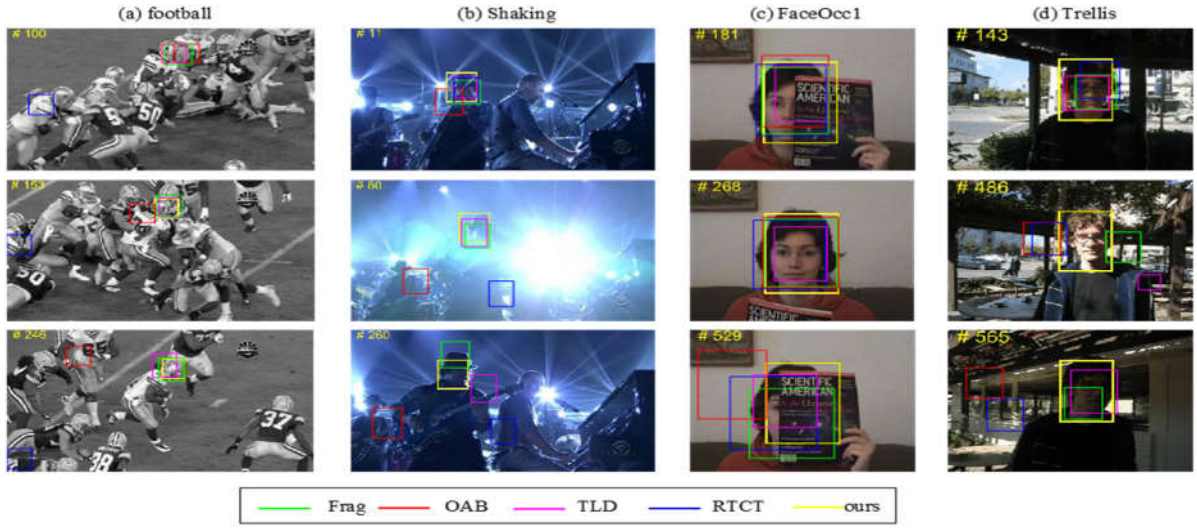

Fig. 7. Tracking resulats of five methods (ours is the proposed method) on four sequences. The name of sequences are shown in first row.

Fig. 8. Tracking results of the mobile robot in the graphic interface and deciding to move it. After object detecion, the robot moves to forward (a) or right (b) or left (c) and it will stop if it is so close to target.

Fig. 9. The results of tracking by the robot.

One of the most important problems in robotics is the ability of online image processing. Most of studied tracking methods have evaluated their experiments solely on recorded sequences.

In addition to applying the proposed method on the sequences used in the state-of-the-art methods, we assessed the performance of our method on a mobile robot built by the robotic team. Tracking results of the mobile robot in the graphical interface are shown in figure 8. After target detection, the direction of robot movement (left, right and forward) is identified relative to the center of the detected bounding box, and then the robot moves toward it. Whenever the robot is very close to the target (as figure 8d), it stops. The proximity of the target is determined by its size. Furthermore, figure 9 and table III provide the results of robot tracking on toy bear and owlet objects in a dynamic environment. In this case, among 1950 test frames on both sequences 1541 frames were correctly detected with 79% accuracy.

Table. 3. Robot performance on 2 sequences.

	# of frames	# of frames detected correctly	Accuracy (# of frames detected correctly / # of frames)
Toy bear	930	715	0.77
Toy owlet	1020	826	0.81
Average	1950	1541	0.79

5. Conclusion

In this paper, we proposed a method using an object model constructed by training the HOG descriptor from few first frames via SVM. The HOG descriptor is similar to the performance of complex cells in the primary visual cortex which is not sensitive to the illumination changes. In order to track the new frame, object model was convolved with the extracted descriptor from surrounding areas of the bounding box of the previous frame. Experimental results demonstrate the proposed method can achieve comparable performance with the other trackers and successfully track the target object. In addition, we applied our method to a mobile robot to evaluate the performance in dynamic environments.

References

[1] Fukuda, T.; Michelini, R.; Potkonjak, V.; Tzafestas, S.; Valavanis, K.; Vukobratovic, M., "How far away is 'artificial man'?", IEEE Robotics & Automation Magazine, vol. 8, no. 1, pp. 66–73 (2001).

[2] Smeulders, A. W. M.; Chu, D. M.; Cucchiara, R.; Calderara, S.; Dehghan, A.; Shah, M., "Visual tracking: An experimental survey", IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 7, pp. 1442–1468 (2014).

[3] Li, A.; Yan, S., "Object Tracking With Only Background Cues", IEEE Trans. Circuits Syst. Video Technol., vol. 24, no. 11, pp. 1911–1919 (2014).

[4] Wang, S.; Lu, H.; Yang, F.; Yang, M.-H., "Superpixel tracking", 2011 International Conference on Computer Vision, (2011).

[5] Zoidi, O.; Tefas, A.; Pitas, I., "Visual object tracking based on local steering kernels and color histograms", IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 5, pp. 870–882 (2013).

[6] Adam, A.; Rivlin, E.; Shimshoni, I., "Robust fragments-based tracking using the integral histogram", in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 798–805 (2006).

[7] Čehovin, L.; Kristan, M.; Leonardis, A., "An adaptive coupled-layer visual model for robust visual tracking", in International Conference on Computer Vision, pp. 1363–1370 (2011).

[8] Chu, D. M.; Smeulders, A. W. M., "Color invariant SURF in discriminative object tracking", in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 6554 LNCS, no. PART 2, pp. 62–75 (2012).

[9] Babenko, B.; Yang, M.-H.; Belongie, S., "Visual Tracking with Online Multiple Instance Learning", IEEE Trans. Pattern Anal. Mach. Intell., pp. 983–990 (2010).

[10] Santner, J.; Leistner, C.; Saffari, A.; Pock, T.; Bischof, H., "PROST: Parallel robust online simple tracking", in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 723–730 (2010).

[11] Gao, J.; Ling, H.; Hu, W.; Xing, J., "Transfer Learning Based Visual Tracking with Gaussian Processes Regression", in 13th European Conference of Computer Vision, Springer International Publishing, pp. 188–203 (2014).

[12] Henriques, J. F.; Caseiro, R.; Martins, P.; Batista, J., "High-speed tracking with kernelized correlation filters", IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 3, pp. 583–596 (2015).

[13] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J., "Rich feature hierarchies for accurate object detection and semantic segmentation", in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014).

[14] Felzenszwalb, P. F.; Girshick, R. B.; Mcallester, D.; Ramanan, D., "Object Detection with Discriminatively Trained Part Based Models", IEEE Trans. Pattern Anal. Mach. Intell., pp. 1–20 (2009).

[15] Shu, C.; Ding, X.; Fang, C., "Histogram of the oriented gradient for face recognition", Tsinghua Sci. Technol., vol. 16, no. 2, pp. 216–224 (2011).

[16] Déniz, O.; Bueno, G.; Salido, J.; De La Torre, F., "Face recognition using Histograms of Oriented Gradients", Pattern Recognit. Lett., vol. 32, no. 12, pp. 1598–1603 (2011).

[17] Xu, G.; Wu, X.; Liu, L.; Wu, Z., "Real-time pedestrian detection based on edge factor and histogram of oriented gradient", in 2011 IEEE International Conference on Information and Automation, ICIA 2011, pp. 384–389 (2011).

[18] Tsagkatakis, G.; Savakis, A., "Online Distance Metric Learning for Object Tracking", Circuits Syst. Video Technol. IEEE Trans., vol. 21, no. 12, pp. 1810–1821 (2011).

[19] Dalal, N.; Triggs, B., "Histograms of oriented gradients for human detection", in Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. I, pp. 886–893 (2005).

[20] Shalev-Shwartz, S.; Zhang, T.; Shalev-Shwartz, S.; Zhang, T., "Stochastic dual coordinate ascent methods for regularized loss minimization", J. Mach. Learn. Res., vol. 14, no. 1, pp. 567–599 (2013).

[21] Vedaldi, A.; Fulkerson, B., "VLFeat - An open and portable library of computer vision algorithms", MM '10 Proceedings of the 18th ACM international conference on Multimedia, pp. 1469–1472 (2010).

[22] Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; Zisserman, A., "The pascal visual object classes (VOC) challenge", Int. J. Comput. Vis., vol. 88, no. 2, pp. 303–338 (2010).

[23] Grabner, H.; Grabner, M.; Bischof, H., "Real-Time Tracking via On-line Boosting", Proc. Br. Mach. Vis. Conf., vol. 1, pp. 1–10 (2006).

[24] Kalal, Z.; Matas, J.; Mikolajczyk, K., "P-N learning: Bootstrapping binary classifiers by structural constraints", in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 49–56 (2010).

[25] Zhang, K.; Zhang, L.; Yang, M.-H., "Real-Time Compressive Tracking", Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7574 LNCS, no. PART 3, pp. 864–877 (2012).

[26] Babenko, B.; Yang, M.-H.; Belongie, S., "Visual tracking with online multiple instance learning", in Proc. IEEE Comput. Vis. Pattern Recognit. Conf., San Diego, CA, pp. 983–990 (2009).

[27] Wu, Y.; Lim, J.; Yang, M.-H., "Online object tracking: A benchmark", in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 2411–2418 (2013).

[28] Ren, X.; Malik, J., "Tracking as repeated figure/ground segmentation", in Proc. IEEE Conf. Comput. Vis. Pattern Recognit, pp. 1–8 (2007).