

Research Article

Real-Time Tracking by Double Templates Matching Based on Timed Motion History Image with HSV Feature

Zhiyong Li, Pengfei Li, Xiaoping Yu, and Mervat Hashem

College of Information Science and Engineering, Hunan University, Changsha 410082, China

Correspondence should be addressed to Zhiyong Li; zhiyong.li@hnu.edu.cn

Received 30 August 2013; Accepted 21 November 2013; Published 23 January 2014

Academic Editors: Y. Lu and F. Yu

Copyright © 2014 Zhiyong Li 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.

It is a challenge to represent the target appearance model for moving object tracking under complex environment. This study presents a novel method with appearance model described by double templates based on timed motion history image with HSV color histogram feature (tMHI-HSV). The main components include offline template and online template initialization, tMHI-HSV-based candidate patches feature histograms calculation, double templates matching (DTM) for object location, and templates updating. Firstly, we initialize the target object region and calculate its HSV color histogram feature as offline template and online template. Secondly, the tMHI-HSV is used to segment the motion region and calculate these candidate object patches' color histograms to represent their appearance models. Finally, we utilize the DTM method to trace the target and update the offline template and online template real-timely. The experimental results show that the proposed method can efficiently handle the scale variation and pose change of the rigid and nonrigid objects, even in illumination change and occlusion visual environment.

1. Introduction

As a hot research topic in computer vision, moving object tracking has numerous applications such as video surveillance, visual navigation, and human-computer interaction. However, it remains a tough problem to track the target under complex environment due to the scale variation, pose change, illumination change, occlusion, and real-time processing requirement. To overcome these difficulties, many object tracking methods have been proposed in recent years [1].

Traditional tracking algorithms such as mean-shift [2] and particle filter [3] are well developed in the past few years. Many extensions have emerged. Recently, Leichter [4] proposed a tracker using cross-bin metrics based on mean-shift, which is a simple and efficient method. Mei and Ling [5] treated tracking problem as a sparse approximation problem in a particle filter framework. As we know, an important aspect that determines the performance of the tracking algorithm is the object's appearance model. However, these traditional methods cannot represent an enjoyable appearance model, which results in poor jobs on handling the scale variation, pose change, and some complicated visual environment such as occlusion and illumination change.

Kass et al. [6] proposed snakes models to deal with the pose change, but the tracking process is a solution optimization process with high computational complexity. Level sets [7], which represent a contour using a signed distance map, were used to draw accurate active contour for tracking. The condensation algorithm [8] used the B-spline curve to parameterize the contour and particle filtering method for tracking. Ross et al. [9] used an incremental subspace model to adapt appearance changes. However, these methods are undesirable because a drift problem may appear once the target object's appearance changes significantly during the tracking process.

Online learning has spawned the approach of tracking by detection, which treats the tracking as a binary classification problem. Collins et al. [10] utilized online feature selection to trace the target. Avidan [11] proposed ensemble tracking method that extended mean-shift with adaBoost. Grabner and Bischof [12] proposed an online boosting tracker which built a feature selection framework for tracking. More recently, Santner et al. [13] built a sophisticated tracking system called PROST which is robust. Zhang et al. [14] proposed a compressive tracking algorithm that used

compressing sensing theories to extract features for real time tracking. Nevertheless, because the appearance model and background must be learned at frame rate and the training data for classifier is numerous, thus, these tracking methods are not efficient for real-time tracking. Meanwhile, it also often leads to tracking drift problem since the appearance model is updated with noise and a few negative examples.

In this paper, we propose a novel method in which the appearance model is described by object templates based on tMHI-HSV. Firstly, a novel moving object segmentation and modeling method named tMHI-HSV are proposed, which is robust and efficient to select and describe candidate object patches. Moreover, the DTM strategy is applied to ensure the accuracy and robustness of the tracking. The online template is dynamically updated in real time to adapt the change of target appearance, and the offline template is updated to deal with the overfitting problem that is caused by the fast changes of environment.

The rest of this paper is organized as follows. We introduce the motion history (MHI) method in Section 2. Section 3 presents the DTM tracking method based on tMHI-HSV. Experimental results are shown and discussed in Section 4. Finally Section 5 concludes the paper.

2. MHI Method

2.1. Motion History Image. The motion history image (MHI) method is an approach based on template matching. The MHI with current motion pixels of images (>2 frames) updated using a timestamp can provide abundant motion information of a moving object.

Ahad et al. [15] had made a survey on MHI method and its applications. As an impressive method for motion representation, MHI is a standard technique in computer vision. Bobick and Davis [16] utilized MHI as temporal templates to recognize the human movement. Zhaozheng and Collins [17] proposed a forward-backward MHI method to locate the moving object in thermal imagery. Lin et al. [18] used MHI method to segment motion region that was more robust than segmenting objects in one frame for tracking moving object. Davis et al. [19] separated human motion patterns from the noise categories by representing the moving object with a minimum spatial size and temporal length based on MHI approach. Using MHI method to represent the motion appearance is simple and effective. An MHI image is computed as follows.

Let $f = [f_1(x, y), \dots, f_{i-1}(x, y), f_i(x, y), \dots, f_n(x, y)]$ denote a video frame stream. The absolute value of frame differencing result $D_t(x, y)$ is computed as

$$D_t(x, y) = |f_i(x, y) - f_{i-1}(x, y)|, \quad (1)$$

where $f_i(x, y)$ and $f_{i-1}(x, y)$ are two adjacent frames of current time t from video frames. The update function

$B_t(x, y)$ is defined by $D_t(x, y)$ based on a threshold σ . Consider

$$B_t(x, y) = \begin{cases} 1 & \text{if } D_t(x, y) > \sigma \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

then, the $MHI_t(x, y)$ is computed according to the update function $B_t(x, y)$:

$$MHI_t(x, y) = \begin{cases} t & \text{if } B_t(x, y) = 1 \\ MHI_{t-1}(x, y) & \text{otherwise.} \end{cases} \quad (3)$$

Since the motion information is preserved in the MHI, it can represent the motion object in a continuous way. Thus, the MHI template is insensitive to some interference, like illumination change and occlusion. These advantages make MHI method suitable for motion analysis in challenging scenarios.

2.2. Timed MHI Generation. Timed motion history image (tMHI) is a smart motion segmentation method [20] that is extended from MHI. A history of temporal changes is kept at each pixel location and then decays over time. The tMHI utilizes a floating-point MHI where new silhouette values are represented with floating-point timestamp. Meanwhile, the tMHI is updated with the timestamp of current system. The more recent pixels of the moving object have higher intensity. The tMHI image is computed as

$$tMHI_\delta(x, y) = \begin{cases} \tau & \text{if current silhouette at } (x, y) \\ 0 & \text{else if } tMHI_\delta(x, y) < (\tau - \delta), \end{cases} \quad (4)$$

where τ is the current timestamp and δ is the decay parameter that determines the motion length. Figure 1 shows the tMHI of a car. From Figure 1, we can conclude that the tMHI provides coherent motion information to represent the motion trail of moving object over time.

3. tMHI-HSV-Based DTM Tracking Method

This section provides a detailed description of the proposed method. The tracking algorithm starts with the work by initializing the tracking window and gets the same offline template and online template in the first frame by user specifying the object region. Next, for each frame of the next video frames stream, some potential candidate moving object patches are screened out by using tMHI-HSV. The *Bhattacharyya distance* measure is used to measure the similarities between each candidate patch and online template, as well as offline template. The patch with the minimum *Bhattacharyya distance* is chosen as the best candidate object in current frame, and its location position and silhouette are outputted as the current object spatial information. Furthermore, the online template is updated by the current frame object patch. Meanwhile, the difference of the offline template and the newest online template is analyzed; the offline template may be updated if the difference is too large. So the tracking and updating cycle will be continued for the whole video steam; the block diagram is shown in Figure 2.



FIGURE 1: (a) The original image of the car. (b) The tMHI of the car. Here, $\sigma = 20$ and $\delta = 10$ s.

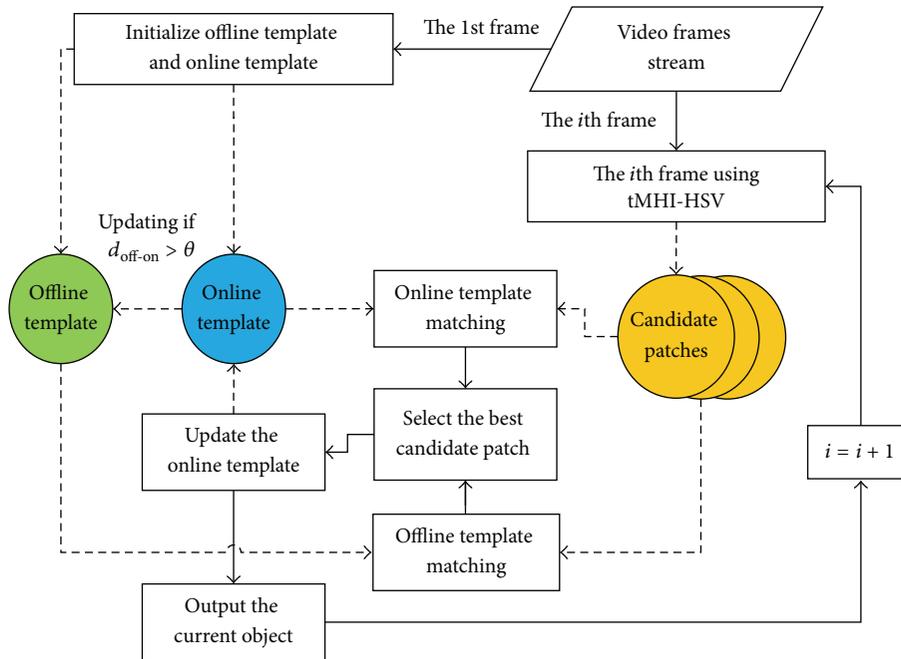


FIGURE 2: Block diagram of the proposed method.

3.1. tMHI-HSV Representation. The tMHI method is used to detect the moving region and segment the moving objects to obtain MHI silhouettes. Before the segmentation, a median filter is employed to eliminate salt and pepper noise. Then, the morphological operation is adopted to remove the target's discontinuous hollow. A threshold Δ_{MinSize} , which is determined by the target's spatial size, is set to find the potential candidate MHI external rectangle silhouettes. The MHI silhouettes whose sizes are larger than Δ_{MinSize} are screened out as the candidate MHI silhouette sets \mathbf{C} as follows:

$$\mathbf{C} = \{c_1, c_2, \dots, c_m \mid S_c \geq \Delta_{\text{MinSize}}, i = 1, 2, \dots, m\}. \quad (5)$$

Here, S_c denotes spatial size of the silhouettes c , and m is the total number of silhouettes. Generally $S_c = r_{\text{width}} \times r_{\text{height}}$; r_{width} and r_{height} are the width and height of the rectangle

silhouette c in pixels, respectively. This processing can help to remove noise pixels from camera jitter and small motion such as shaking leaves.

However, the MHI silhouette cannot describe the whole of the candidate patch information except the spatial motion feature. Meanwhile, it is lack of robustness and accuracy to track the target just according to the spatial size and temporal length. Hence, one kind of global feature, color feature, is employed and fused with tMHI to exactly represent these candidate patches' specialties. Compared to the RGB system, HSV system describes more accurately than RGB color system on perception links and remains computationally simple. Therefore, the HSV system is chosen to represent those patches' color feature.

Thus, each candidate patch c can be described by the HSV color histogram \mathbf{H} of its tMHI region. $\mathbf{H} = \{h(b) \mid b =$

$1, 2, \dots, q$ }, which is named tMHI-HSV feature in this paper, and $h(b)$ denotes the number of pixels that belong to the color bin b . The color histogram \mathbf{H} can be computed as follows:

$$\mathbf{H} = \{h(b)\} = \left\{ \sum_{i=1}^N Q(p_i, b) \right\}, \quad b = 1, 2, \dots, q, \quad (6)$$

$$Q(p_i, b) = \begin{cases} 1 & \text{if } p_i \text{ belong to color bin } b, \\ 0 & \text{otherwise,} \end{cases}$$

where b is one of the color bins, q is the total number of color bins, N is the number of pixels in this candidate patch c , and p_i is the color value of the i th pixel. Rather than only the spatial feature, the color feature is used for modeling the candidate patches and templates.

3.2. Double Template Matching. During the whole tracking process, the general method [21] initializes the first frame to delimit the tracking window as the template region. However, a series of changes such as scale variation and pose change may occur on the target during the tracking process. In addition, changes in the visual environment can cause interference. The appearance change and circumstance disturbing always lead to a drift problem and even tracking lost, because the original template cannot accurately represent the appearance model of the current target. Therefore, updating template online in real time during tracking is very crucial. However, an over-fitting problem would occur for online template matching when the visual environment and appearance change rapidly and noisily. For example, the illumination changes sparkingly. For the sake of avoiding the online template over-fitting learning problem, one stable offline template is reserved as the second matching template. The patch can be chosen as the best candidate current object if it has the minimum similarity degree to the online template and offline template. This method is called the double template matching (DTM) here.

Generally, the target's appearance changes only a little between two adjacent frames f_{i-1} and f_i . It is a reasonable way that uses the object patch in the previous frame as the online template to trace the target in the next frame. Here, the *Bhattacharyya distance* of tMHI-HSV color histogram is used to measure the similarity between the online templates T_{Online} and the candidate object patch c_i .

According to the definition of *Bhattacharyya distance* based on two vectors $\mathbf{p} = \{p_1, p_2, \dots, p_k\}$ and $\mathbf{q} = \{q_1, q_2, \dots, q_k\}$, we have

$$d_{\text{Bhattacharyya}}(\mathbf{p}, \mathbf{q}) = \sqrt{1 - \frac{\sum_k \sqrt{p_k \times q_k}}{\sum_k p_k \times \sum_k q_k}}. \quad (7)$$

Thus, the online template matching and the offline template matching solve this optimization problem as follows:

$$\begin{aligned} j_{\text{Online}} &= \arg \min_i \{d_{\text{Bhattacharyya}}(c_i, T_{\text{Online}})\}, \\ j_{\text{Offline}} &= \arg \min_i \{d_{\text{Bhattacharyya}}(c_i, T_{\text{Offline}})\}. \end{aligned} \quad (8)$$

Moreover, the best candidate patch c_{Best} can be chosen from the two minimum *Bhattacharyya distance* patches as follows:

$$\begin{aligned} c_{\text{Best}} &= \{c_j \mid j \\ &= \arg \min_{j_{\text{Online}}, j_{\text{Offline}}} \{d_{\text{Bhattacharyya}}(c_{j_{\text{Online}}}, T_{\text{Online}}), \\ &\quad d_{\text{Bhattacharyya}}(c_{j_{\text{Offline}}}, T_{\text{Offline}})\}\}. \end{aligned} \quad (9)$$

So, c_{Best} can be the best candidate object in the current frame to be outputted. Here, double template matching is adopted to avoid online template over-fitting problem. When the online template has overlearned some noise change from the last several frames, the tracking drift will occur. This means that the tracking method will match the current object to wrong patch. The proposed method can ameliorate this phenomenon because an offline template has been adopted at the same time. The offline template is a more stable template in this situation, because it learns less from the object appearance than online template.

3.3. Template Updating. In order to keep the tracking accuracy, templates updating is the key procedure in this proposed method. Here, the current best candidate object patch c_{Best} is as the newest online template of the next frame as follows:

$$T_{\text{Online}} = c_{\text{Best}}. \quad (10)$$

Meanwhile, an over-fitting problem caused by online template matching may occur when the visual environment changes rapidly and randomly. For example, the illumination suddenly changes sparkingly or the object goes through some obstruct continuously and quickly. To avoid this problem, a threshold θ , which is set to 0.3 by experimental picking in this paper, is used to evaluate whether the offline template needs to be updated or not. Compared to the current online template T_{Online} , the offline template can be updated as follows:

$$\begin{aligned} d_{\text{Off-On}} &= d_{\text{Bhattacharyya}}(T_{\text{Offline}}, T_{\text{Online}}), \\ T_{\text{Offline}} &= \begin{cases} T_{\text{Online}} & \text{if } d_{\text{Off-On}} > \theta \\ T_{\text{Offline}} & \text{otherwise.} \end{cases} \end{aligned} \quad (11)$$

As above, the offline template is updated only when it differs much from the online template. So this strategy guarantees that our method is more robust under the frequently changing environment.

3.4. Algorithm Pseudocode. This proposed method's pseudocode is shown in Algorithm 1.

4. Experimental Results and Analysis

The proposed algorithm is evaluated against scale variation, pose change, occlusion, and illumination change. Our tracker

```

(1) Input: Video frames stream  $f_1, f_2, \dots, f_t, \dots$ 
(2) Output: Object center position  $P_t$  and rectangle silhouette  $C_t^{\text{Best}}$ .
(3) for each  $f_i, i = 1, 2, \dots, t, \dots$  do
(4)   if  $i = 1$ 
(5)     Initialize the  $f_1$  to establish the offline template  $T_{\text{Offline}}$ , the online template  $T_{\text{Online}}$  and the algorithm parameter, such as  $\sigma, \delta, \Delta_{\text{MinSize}}, \theta$  and so forth.
(6)   else
(7)     Segment the moving object by using tMHI
(8)     Get the candidate object patches  $C_t^j (j = 1, 2, \dots, m)$ 
(9)     for each  $C_t^j, j = 1, 2, \dots, m$  do
(10)      Calculate the HSV color histograms  $H_t^j$  of  $C_t^j$ .
(11)      Calculate the best candidate object  $c_{\text{Best}}$  and  $d_{\text{Off-On}}$ 
(12)      Update the online template,  $T_{\text{Online}} \leftarrow c_{\text{Best}}$ 
(13)      Update the offline template,  $T_{\text{Offline}} \leftarrow T_{\text{Online}}$  if  $d_{\text{Off-On}} > \theta$ 
(14)      Output the center position  $P_t$  and rectangle silhouette  $C_t^{\text{Best}}$  from the best candidate patch  $c_{\text{Best}}$ 
(15)     end for
(16)   end if
(17) end for

```

ALGORITHM 1: tMHI-HSV-based DTM tracking.

TABLE 1: Video parameters.

Video sequence	Video parameters		
	Frame size/pixel	Total frames	Target
Blue car	768×576	50	Blue car
White car	768×576	114	White car
Girl pose	640×480	111	Girl
Intelligent room	320×240	210	Man
Girl occlusion	640×480	150	Girl
Car	320×240	89	Car

TABLE 2: Algorithm parameters.

Video sequence	Algorithm parameters		
	σ	δ	Δ_{MinSize}
Blue car	20	0.2	500
White car	20	0.7	500
Girl pose	25	1.5	1000
Intelligent room	25	0.8	1000
Girl occlusion	30	1.5	1000
Car	20	0.3	500

is compared with two state-of-art algorithms, the online boosting tracker (OBT) [12], and compressive tracking (CT) [14] methods. The *red, yellow, and green rectangle windows* are used to mark the target tracked by our method, OBT and CT, respectively. The source codes, which are provided by the authors, are used for the comparison purpose.

The parameters of the video sequences are displayed in Table 1, and the initialization algorithm parameters are shown in Table 2. The binary threshold σ is set from 20 to

TABLE 3: Success rate (%).

Video sequence	Method		
	CT	OBT	Ours
Blue car	80	76	98
White car	14	13	90
Girl pose	69	70	87
Intelligent room	44	29	90
Girl occlusion	70	64	95
Car	38	88	87
Average SR	52	56	91

30 and the offline template updating threshold θ is set as 0.3, whereas Δ_{MinSize} and δ can be set depending on the practical application as in Table 2. We run the experiments on a PC with Intel Pentium 2.70 GHz CPU and 2 GB RAM. The tracking rates frames per second (FPS) obtained by using the proposed method can reach 16FPS, 21FPS, and 62FPS corresponding to the video sizes 768×576 , 640×480 , and 320×240 , respectively, which demonstrates that our method is well positioned to meet the real-time requirement.

4.1. Qualitative Evaluation

Scale Variation. We utilize the *PETS2000* sequences of cars to test the performance of OBT, CT, and our method in handling the scale variation. The scale of the car changes from small to big in the *blue car* sequence and it changes from big to small with rotation in the *white car* sequence. As we can see from the tracking results shown in Figure 3, the three trackers are all able to trace the car. However, we can see that CT and our method perform well, while the OBT method fails when

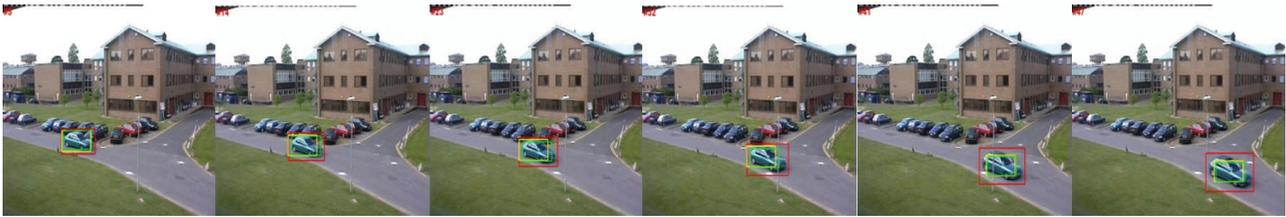


FIGURE 3: Tracking results of *PETS2000* sequence of *blue car*. Frames 5, 14, 23, 32, 41, and 47 are displayed.



FIGURE 4: Tracking results of *PETS2000* sequence of *white car*. Frames 5, 25, 45, 65, 85, and 105 are displayed.



FIGURE 5: Tracking results of *girl pose* sequence. Frames 5, 25, 45, 65, 85, and 105 are displayed.



FIGURE 6: Tracking results of *intelligent room* sequence. Frames 16, 32, 72, 123, 161, and 193 are displayed.

the scale of the white car becomes smaller as in Figure 4. In addition, the tracking windows of OBT and CT cannot adaptively change with the scale variation of the car.

Pose Change. We use the *girl pose* sequence captured by ourselves and *intelligent room* sequence to evaluate the trackers' abilities in handling pose change problem. Some tracking results are illustrated in Figures 5 and 6. From Figure 5, we note that our tracker outperforms OBT and CT methods. No matter how the girl moves, our tracker is satisfied while the others are not. As for the *intelligent room* sequence, the man has a pose change. Into the bargain, his scale changes a lot. Both of the OBT and CT methods produce drift problems. OBT tracker even lost the target, whereas our method utilizes tMHI and an online template is updated in real time to adapt the target's pose change. Thus, we obtain better results.

Occlusion. The *girl occlusion* sequence is used to test the performance of our tracker when the target is under heavy

or even complete occlusion. Part of the tracking results is presented in Figure 7. From 83th and 93th frames, we note that the three trackers all succeed in tracking the target when the partial occlusion occurs. However, the OBT and CT methods fail to track the target when the occlusion is heavy or even complete while our method performs well. The tMHI and DTM make contribution to handle this thorny problem. A detector is included in tMHI, and the motion trail of the target can learn from it. We cannot learn the online template to describe the target, whereas the offline template is valuable when the complete occlusion happens. Therefore, a satisfied result is produced by our method.

Illumination Change. For the tracking results of *car sequence* shown in Figure 8, the illumination changes significantly. Our method and OBT perform well in this situation, while the CT tracker fails to track the target when the car moves from bright region to dark region. Our online tracking template is updated in real time to adjust to the target's appearance



FIGURE 7: Tracking results of *girl occlusion*. Frames 83, 93, 103, 113, 123, and 133 are displayed.



FIGURE 8: Tracking results of *car sequence*. Frames 6, 15, 24, 33, 42, and 51 are displayed.

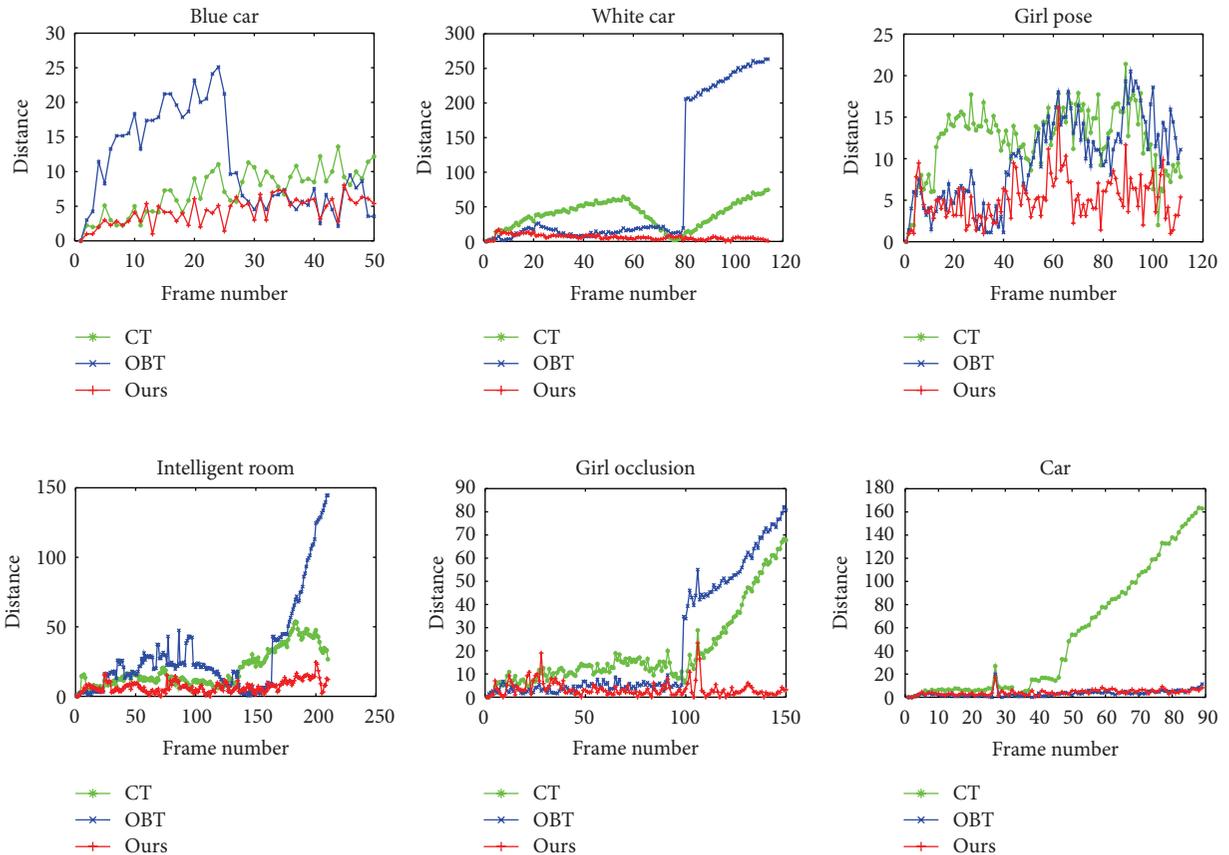


FIGURE 9: Results of CLE.

variation. In addition, the tMHI-HSV feature is insensitive to illumination change. These advantages make our approach robust to illumination change.

4.2. Quantitative Evaluation. In addition to the qualitative evaluation, the success rate (SR) and center location error (CLE) measured with manually labeled ground truth data are

used for the quantitative evaluation. The score of SR is defined as

$$\text{score} = \frac{\text{area}(\text{ROI}_T \cap \text{ROI}_G)}{\text{area}(\text{ROI}_T \cup \text{ROI}_G)}, \quad (12)$$

where ROI_T is the tracking bounding box and ROI_G is the ground truth tracking box. We consider the tracking result as

a success if the score is larger than 0.5 in one frame. The SRs presented in Table 3 demonstrate that our method achieves the best result or the second best result. The distance (pixel) between the center location of the tracked target and the ground truth is used to measure the CLE. The results of CLE shown in Figure 9 illustrate that our method outperforms the other two methods. For the *blue car* sequence, the OBT performs slightly worse, although the three trackers have low CLEs. The OBT fails to track the car after the 80th frame in the *white car* sequence. CT and our method succeed in tracking the white car, whereas our method produces very few CLEs. The distances calculated from *girl pose* sequence are all lower than 25, yet our method shows a better result on the whole sequence. In terms of *intelligent room* sequence, OBT lost the target after the 160th frame and the distances produced by CT are larger than that generated through our method. As to the *girl occlusion* sequence, both OBT and CT are unable to track the girl when she is under heavy or even complete occlusion. However, our method maintains the performance in such scenario. From the results of *car* sequence shown in Figure 9, we note that the CT cannot handle the illumination change problem, while OBT and our method perform well.

5. Conclusions and Future Work

A real-time tracking method with object tMHI-HSV appearance model and double templates (offline and online) matching has been presented. The initial offline template and online template are generated using the original shape and HSV color histogram feature of the target. On the current frame's candidate patches, we utilize the tMHI to segment them. Rather than the spatial pixel template, the HSV color feature is used in our method, which reduces the computational complexity and increases the robustness. These advantages make the motion appearance expression simple and effective. Double template matching is adopted to exactly determine the target location. Meanwhile, the online template is updated real-timely, and offline template is updated as needed. We evaluate our method in six scenarios. The tracking rates illustrate that the proposed algorithm meets the real-time requirement, and the comparative experiments demonstrate that our method outperforms the other two schemes in terms of accuracy and robustness.

For future work, we will extend the proposed method and apply it on a moving camera rather than a fixed one, as we used in our experiment. Considering that features are very important for moving object tracking, other features except HSV color histogram can be used.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is partially supported by the Natural Science Foundation of China (Grant no. 61173107, 91320103), the

Research Foundation of Industry-Education-Research Cooperation in Guangdong Province, the Ministry of Education and Ministry of Science & Technology, China (Grant no. 2011A091000027) and the National High-Tech, R&D Program (863), China (no. 2012AA01A301-01).

References

- [1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Computing Surveys*, vol. 38, no. 4, p. 13, 2006.
- [2] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.
- [4] I. Leichter, "Mean shift trackers with cross-bin metrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 695–706, 2012.
- [5] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2259–2272, 2011.
- [6] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [7] N. Paragios, "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 3, pp. 266–280, 2000.
- [8] M. Isard and A. Blake, "Condensation—conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.
- [9] 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. 1–3, pp. 125–141, 2008.
- [10] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631–1643, 2005.
- [11] S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261–271, 2007.
- [12] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 1, pp. 260–267, June 2006.
- [13] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "PROST: parallel robust online simple tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 723–730, June 2010.
- [14] K. Zhang, L. Zhang, and M. H. Yang, "Real-time compressive tracking," in *Proceedings of the 12th European Conference on Computer Vision (ECCV '12)*, pp. 864–877, Springer, Berlin, Germany, 2012.
- [15] M. A. R. Ahad, J. K. Tan, H. Kim, and S. Ishikawa, "Motion history image: its variants and applications," *Machine Vision and Applications*, vol. 23, pp. 255–281, 2010.

- [16] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001.
- [17] Y. Zhaozheng and R. Collins, "Moving object localization in thermal imagery by forward-backward MHI," in *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '06)*, p. 133, June 2006.
- [18] Y. Lin, Q. Yu, and G. Medioni, "Efficient detection and tracking of moving objects in geo-coordinates," *Machine Vision and Applications*, vol. 22, no. 3, pp. 505–520, 2011.
- [19] J. W. Davis, A. M. Morison, and D. D. Woods, "Building adaptive camera models for video surveillance," in *Proceedings of the 7th IEEE Workshop on Applications of Computer Vision (WACV'07)*, p. 34, February 2007.
- [20] G. R. Bradski and J. W. Davis, "Motion segmentation and pose recognition with motion history gradients," *Machine Vision and Applications*, vol. 13, no. 3, pp. 174–184, 2002.
- [21] R. Kaucic, A. G. A. Perera, G. Brooksby, J. Kaufhold, and A. Hoogs, "A unified framework for tracking through occlusions and across sensor gaps," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 990–997, June 2005.



Hindawi

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

