Research Article

Adaptive Randomized Ensemble Tracking Using Appearance Variation and Occlusion Estimation

Weisheng Li and Yanjun Lin

Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Correspondence should be addressed to Weisheng Li; liws@cqupt.edu.cn

Received 30 October 2015; Revised 30 December 2015; Accepted 4 January 2016

Academic Editor: Daniel Zaldivar

Copyright © 2016 W. Li and Y. Lin. 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.

Tracking-by-detection methods have been widely studied with promising results. These methods usually train a classifier or a pool of classifiers in an online manner and use previous tracking results to generate a new training set for object appearance and update the current model to predict the object location in subsequent frames. However, the updating process may easily cause drifting in terms of appearance variation and occlusion. The previous methods for updating the classifier(s) decided whether or not to update the classifier(s) by a fixed learning rate parameter in all scenarios. The learning rate parameter has a great influence on the tracker's performance and should be dynamically adjusted according to the change of scene during tracking. In this paper, we propose a novel method to model the time-varying appearance of an object that takes appearance variation and occlusion of local patches into consideration. In contrast with the existing methods, the learning rate for updating classifier ensembles adaptively is adjusted by estimating the appearance variation with sparse optical flow and the possible occlusion of the object between consecutive frames. Experiments and evaluations on some challenging video sequences have been done and the results demonstrate that the proposed method is more robust against appearance variation and occlusion than those state-of-the-art approaches.

1. Introduction

Visual object tracking, which is one of cardinal problems in computer vision, has a wide range of applications including video surveillance, human computer interaction, video retrieval, and autonomous navigation. Despite the fact that numerous single object tracking methods have been proposed, many of them only achieve favorable performance in simple environment with slow motion and slight occlusion. Hence, it remains a challenging work to research a robust algorithm for complex and dynamic scenes caused by the distractions such as heavy occlusion, appearance variation, and cluttered background.

The purpose of object tracking is to estimate the states of a moving target in a video. A tracking system usually has three components including appearance model, motion model, and model update. An appearance model is used to represent the object with proper features and verify predictions by object representations. A motion model is exploited to predict the most likely state of the target. A model update scheme

is applied to make the tracker adapt to appearance variation and occlusion of the target object. Existing trackers can be classified as either generative or discriminative. For generative methods [1–6], tracking is simplified as searching for the most similar position to the target object within many neighborhood positions of current location. The target is often represented by a set of templates. The fragment-based (Frag) tracker [1] addresses the partial occlusion problem by modeling object appearance with histograms of local patches. The incremental visual tracker (IVT) method [2] utilizes an incremental subspace model to adapt appearance variation. The visual tracking decomposition (VTD) approach [3] extends the conventional particle filter framework with multiple motion and observation models to account for appearance variation. The ℓ_1 tracker [4] first applies sparse representation to visual tracking with designed trivial templates to handle occlusions and treats the tracking as finding the image region with minimal reconstruction error using ℓ_1 minimization. The distribution field (DFT) tracker [5] introduces a method for building an image descriptor using distribution fields

(DFs), a representation that allows smoothing the objective function without destroying information about pixel values. Li et al. [6] construct an appearance model using the 3D discrete cosine transform and propose an incremental 3D-DCT algorithm and then embed the discriminative criterion into a particle filtering framework for object state inference. Most of these methods use holistic representations to describe objects and hence do not handle occlusions or distracters well. For discriminative methods, tracking is treated as a binary classification problem which aims at designing an ensemble classifier to distinguish the target object from the background [7–14]. They utilize both the target and the background information. Avidan [7] combines a set of weak classifiers into a strong one and develops an ensemble tracking method. The OAB tracker [8] proposes an online boosting method to update discriminative features to handle the drifting problem in object tracking. Bai et al. [9] propose randomized ensemble tracker that is extended from the online boosting algorithm [8] and ensemble tracker [7] by characterizing the ensemble weight vector as a random variable and evolving its distribution with recursive Bayesian estimation. Babenko et al. [10] introduce multiple instance learning into online object tracking where samples are considered within positive and negative bags or sets. Kalal et al. [11] propose the P-N learning algorithm to exploit the underlying structure of positive and negative samples in learning classifiers for object tracking. Zhang et al. [12] propose a real-time compressive tracking algorithm by adopting random projection to project a datum in high-dimensional space to a low-dimensional vector. The structured output tracking (Struck) method [13] is proposed by adopting kernelized structured output support vector machine to avoid the labeling ambiguity when updating the classifier during tracking. Wang et al. [14] incorporate online distance metric learning into visual tracking based on a particle filter framework, and thus the appearance variations of an object are effectively learned via an online metric learning mechanism. Furthermore, several algorithms have been proposed to exploit the advantages of both generative and discriminative models [15, 16].

In this paper, we mainly focus on model update scheme because the part has a great impact on the performance of the trackers. Existing strategies usually concentrate on updating the weights of classifiers and ignore the update of learning rate of the classifiers. To address the problem, we introduce appearance variation estimation and occlusion estimation to control the learning rate of the classifiers motivated by [15, 17–19]. We estimate the appearance variation using sparse optical flow and possible occlusion by the reconstruction error of local patches. In summary, we propose a robust tracking method with an adaptive appearance model. During tracking, we exploit the local patches to handle occlusion and appearance variation. The model is adaptively updated with the consideration of occlusions to account for variations and alleviate drifts.

The remainder of this paper is organized as follows. The work related to visual tracking and sparse representation is reviewed in Section 2. In Section 3, the proposed method with appearance variation and occlusion estimation is introduced. Experimental results and demonstrations are reported

and analyzed in Section 4 and the conclusion is given in Section 5.

2. Related Work

In this section, we discuss the related online tracking algorithms handling appearance variation and occlusion of target object. Although much progress has been made for visual tracking, a lot of challenges remain to design an effective and robust tracker due to some distractions including appearance variation, varying illumination, occlusions, and background clutter.

Tracking-by-detection [7, 9] is increasingly popular due to its top performance recently. The motivation of these methods treats tracking as a detection and classification problem, that is, to avoid modeling object dynamics especially when abrupt motion and occlusions occur. They require training a classifier to distinguish the object from the background for detecting the object in each frame. One common approach is to use an ensemble classifier that linearly combines many weak classifiers with different associated weights, for example, [7, 9]. This is done by constructing a strong classifier which combined a pool of weak classifiers using the initial frame, then updating the weights of classifiers by abandoning some bad weak classifiers, and adding some new good weak classifiers trained on next frame at each time step.

Appearance variation and occlusion are two problems for visual tracking. Some current algorithms have been proposed using holistic and local representation schemes to handle appearance variation and occlusion. The IVT tracker [2] with adaptive appearance model that aims to account for appearance variation or limited deformable motion is presented. It is less effective in handling heavy occlusion as a result of the adopted holistic appearance model. The ensemble tracker [7] formulates tracking as a binary classification problem. Although this method is able to make a distinction between target and background, it is rather limited to handle heavy occlusion. The Frag tracker [1] aims to solve partial occlusion with a representation based on histograms of local patches. The template combining votes of matching local patches is not updated and thereby it is not expected to handle appearance variation in terms of large variation in scale and shape deformation. The MIL tracker [10] is able to reduce drift, but it is not able to handle large shape deformation. The ℓ_1 tracker [4] does not exploit the appearance information from the background and thus it is ineffective in handling heavy occlusion.

As mentioned above, how to handle appearance variation and occlusion are two major problems that are also two hot research topics for object tracking. Motivated by [15, 17–19], we introduce methods to estimate appearance variation and possible occlusion in the process of model update. An optical flow algorithm calculates the motion information of the moving target from one frame to the next by using the intensity values of neighboring pixels. So, we can use it to estimate the rate of appearance variation between consecutive frames. Motivated by the widely successful applications of

sparse representation in many tasks, a new sparsity-based occlusion estimation method is designed.

3. The Proposed Method

In this section, we introduce the appearance variation estimation based on sparse optical flow and sparsity-based occlusion estimation methods of our tracker. We solve the problem that learning rate of classifier could not be dynamically adapted with the change of scene during tracking. Our tracker is a variant of RET [9].

3.1. The RET Tracker. We give a brief introduction of the RET tracking method. The tracker puts emphasis on estimating the state of the classifier rather than the state of the object. It characterizes the ensemble weight vector that combines weak classifiers as a random variable and evolves its distribution with recursive Bayesian estimation. The object bounding box is divided into local patches of size $8 * 8$. For each patch, it extracts a histogram of gradients (HOG) descriptor. At each time step, the method starts with the pool of weak classifiers $C = \{c_1, c_2, \dots, c_N\}$, a distribution $\text{Dir}(D)$ over the weight vector D , and input data x . It samples the distribution $\text{Dir}(D)$ to get M instantiations $D^{(1)}, D^{(2)}, \dots, D^{(M)}$ of the weight vector D and combines them with the output of weak classifiers to yield M ensembles of weak classifiers $f_{D^{(1)}}, f_{D^{(2)}}, \dots, f_{D^{(M)}}$. These M ensembles can be interpreted as M instantiations of the randomized classifier f_D and are used to compute the approximation F of the expected output of the randomized classifier f_D . The approximation F is considered as the output of the strong classifier for input data x .

The classification method of the tracker is described as follows. Given a weight vector D and input data x , we obtain an ensemble binary classifier f_D of the pool by thresholding the linear combination of outputs $c_i(x)$ of all weak classifiers:

$$f_D(x) = \begin{cases} 1 & \text{if } \sum_{i=1}^N d_i c_i(x) \geq \tau \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where d_i is a component of weight vector associated with c_i , N is the dimensionality of weight vector D , and τ is a model parameter.

$S^{(0)}, S^{(1)} \dots$ denote the series of sequentially arriving datasets. At time step t , given input data $x \in S^{(t)}$, the probabilistic label value y^* is computed:

$$\begin{aligned} y^* &= E[y | x] = \int y p(y | x, S^{(0)} \dots S^{(t-1)}) dy \\ &= \int y \int p(y | x, D) p(D | S^{(0)} \dots S^{(t-1)}) dD dy \\ &= \int p(D | S^{(0)} \dots S^{(t-1)}) \left(\int y p(y | x, D) dy \right) dD \\ &= \int f_D(x) p(D | S^{(0)} \dots S^{(t-1)}) dD, \end{aligned} \quad (2)$$

where $H = (h_1, h_2, \dots, h_N)$ is base distribution, which is the expectation of vector D , and α is the concentration parameter and $p(D | S^{(0)} \dots S^{(t-1)})$ follows the Dirichlet distribution $\text{Dir}(D; \alpha, H)$.

The final strong classifier is obtained for input data x by voting $f_{D^{(1)}}, f_{D^{(2)}}, \dots, f_{D^{(M)}}$ and thresholding:

$$F(x) = \begin{cases} 1 & \text{if } \frac{1}{M} \sum_{i=1}^M f_{D^{(i)}}(x) \geq 0.5 \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

3.2. Appearance Variation Estimation Based on Sparse Optical Flow. In visual tracking, appearance variation is an important factor that affects the performance. Most existing methods solve the problem by involving some feature descriptors and ignore estimating the rate and degree. The methods proposed in [17, 20] have shown that appearance variation can be detected by observing changes in the induced optical flow. The magnitude of the induced optical flow vectors will depend on the distance to the object as well as the velocity difference between the object and the video capture devices. The sampling rate of a certain video sequence is constant. If the velocity of the video capture devices is constant, the change in the optical flow vector magnitude will reflect changes in the distance to the object. So, the rate can be computed. If the velocity difference between the object and the video capture devices is small, the change in the optical flow vector magnitude will reflect the degree of appearance variation. Sparse optical flow method is introduced due to the fact that the traditional optical flow algorithms cannot satisfy the request of real-time because they need a large amount of computation cost. Many methods such as Harris corners, Canny edges, SIFT, and SURF can be used to extract points of interest in the process of sparse optical flow calculation.

Motivated by the demonstrated success of optical flow [17, 21], we propose a method based on sparse optical flow to estimate the appearance variation of the object between consecutive frames. The workflow of our method is described in Figure 1. For each frame, we firstly employ the Canny edge descriptor to detect corners with the purpose of extracting suitable and accurate key points in a real-time way. Secondly we calculate the optical flow using the image pyramids and the Lucas-Kanade algorithm and estimate the image velocities with subpixel precision because of possible large displacement. Thirdly we get an optical flow descriptor composed of the position of key points and the magnitude of the corresponding flow vectors. Finally, we get the estimation result by Euclidean distance.

Sparse Optical Flow Descriptors. Given the frame f , let D_f represent the description of the optical flow vectors within

FIGURE 1: The workflow of appearance variation estimation method using the sparse optical flow.

FIGURE 2: The diagram of sparsity-based occlusion estimation method.

the frame. For this frame, the sparse optical flow descriptor is given as follows:

$$D_f = \begin{bmatrix} x_{f1} & y_{f1} & A_{f1} \\ x_{f2} & y_{f2} & A_{f2} \\ \vdots & \vdots & \vdots \\ x_{fT} & y_{fT} & A_{fT} \end{bmatrix}, \quad (4)$$

where (x_{fi}, y_{fi}) represent the position of i th corner, A_{fi} is the magnitude of flow vector at the corner, and T denotes the number of corners in the frame f .

Distance Measure. We can estimate the appearance variation after getting the descriptor. In order to estimate the changes of object, we compute the similarity between the consecutive descriptors. The frames i, j indicate two consecutive frames, and D_i, D_j denote their sparse optical flow, respectively. The similarity can be computed by using the Euclidean distance as follows:

$$E_{i,j} = \sqrt{\sum_{k=1}^T ((x_{ik} - x_{jk})^2 + (y_{ik} - y_{jk})^2 + (A_{ik} - A_{jk})^2)}, \quad (5)$$

where the bigger the value is, the faster and heavier the appearance variation is—and vice versa. To measure the rate of appearance variation, we can estimate the rate by computing the displacement of the object because the time

is a fixed value. Then we commonly estimate the rate by thresholding the similarity. Consider

$$R_i = \begin{cases} E_{i,j} & \text{if } E_{i,j} \leq \varepsilon_0 \\ \varepsilon_0 & \text{otherwise,} \end{cases} \quad (6)$$

where ε_0 is a predefined threshold.

3.3. Sparsity-Based Occlusion Estimation. In order to estimate the possible occlusion, motivated by some successful applications of sparse representation [15, 19, 22–24] and the effectiveness in handling occlusions for object tracking, we develop a new sparsity-based occlusion estimation method for object tracking.

In our method, each image patch is sparsely represented by its neighboring patches. If the patch is occluded, its true neighboring patches are unable to be found and thus causing large reconstruction error. The reconstruction error can be used to evaluate the occlusion state of each image patch. The larger the reconstruction error of the patch is, the greater the possibility of the patch which is occluded is. If the error of the patch is bigger than a threshold, we regard the patch as an occlusion patch. So, we can estimate the occlusion between two consecutive frames. The diagram of our sparsity-based occlusion estimation method is shown in Figure 2.

We use overlapped sliding windows on the images to obtain some patches with the same size. Each patch is converted to a vector. For better description of the algorithm, we use Figure 2 to explain the process. Taking patch P_{ij} centered at the i th row and the j th column of image I and

the vector of the patch $V_{P_{ij}}$ into consideration, the sparse coefficient vector $\beta_{P_{ij}}$ of each patch is computed by

$$\min_{\beta_{P_{ij}}} \|V_{P_{ij}} - D_{P_{ij}}^J \beta_{P_{ij}}\|_2^2 + \lambda \|\beta_{P_{ij}}\|_1, \quad (7)$$

where λ is a constant, $D_{P_{ij}}^J$ is the dictionary which is made up of several special selected patches from image J and can be denoted as $D_{P_{ij}}^J = \{U_{mn} \mid m = i - l, \dots, i, \dots, i + l; n = 1, 2, \dots, C\}$, J_{mn} represents the patch centered at the m th row and the n th column of image J , l is a constraint parameter in horizontal direction, and C is a constant.

The construction error can be used to estimate the occlusion. The construction error of the patch P_{ij} according to image J is denoted as $E_{P_{ij}}^J$ and can be computed as follows:

$$E_{P_{ij}}^J = \|V_{P_{ij}} - D_{P_{ij}}^J \beta_{P_{ij}}\|_2^2. \quad (8)$$

Then we obtain the vector O ; its each element $O_{P_{ij}}$ is an indicator of occlusion of the corresponding patch and is obtained:

$$O_{P_{ij}} = \begin{cases} 1 & \text{if } E_{P_{ij}}^J \leq \varepsilon_1 \\ \varepsilon_1 & \text{otherwise,} \end{cases} \quad (9)$$

where ε_1 is a predefined threshold which determines if the patch is occluded or not.

From the above equation, the occlusion values of image I according to image J can be obtained:

$$\text{Occ}_i = \frac{\sum_{i=1, j=1}^{\text{Wr, Hr}} (E_{P_{ij}}^J * O_{P_{ij}})}{\sum_{i=1, j=1}^{\text{Wr, Hr}} E_{P_{ij}}^J}, \quad (10)$$

where Wr , Hr denotes the ratio of the width of image and the width of patch and the ratio of the height of image and the height of patch, respectively.

3.4. Model Update. In tracking, the object appearance often significantly changes because of a number of factors such as appearance variation and occlusion. Hence it is necessary and important to update the classifier over time. In RET tracker, it updates both the Dirichlet distribution of weight vectors and the pool of weak classifiers after the classification stage in each time step. It updates the Dirichlet parameters and H in a Bayesian manner and updates the pool of weak classifiers by using a fixed learning rate parameter. We improve the method by using an adaptive classifier update. Our approach takes not only the appearance variation of the object but also possible occlusion into consideration.

Once a new set of positive and negative samples is identified, a tracker should decide whether or not to use it to update the classifiers. RET updates the pool of classifiers by the following equation:

$$F(u)^t = (1 - \lambda) F(u)^{t-1} + \lambda F(u), \quad (11)$$

where $F(u)$ is the classifier coefficients and λ is a learning rate parameter. RET uses a fixed learning rate, which means that the appearance will be updated without adaptation to specific frames. Once the tracker loses the object, the whole model will be contaminated in the remaining frames. We tackle this problem by adaptively updating the learning rate parameter λ .

It is apparent that the model of local patches with fast appearance variation should be updated to learn the target appearance, while the model of local patches with heavy occlusion should not be updated to avoid introducing errors. The learning rate for the pool of classifier is set according to the estimation of appearance variation and occlusion. Our classifier update method is defined as follows:

$$\lambda_i = \begin{cases} 0.5 + R_i & \text{if } R_i \geq \text{Occ}_i \\ 0.5 - \text{Occ}_i & \text{otherwise} \end{cases} \quad (12)$$

$$F(u)_i^t = (1 - \lambda_i) F(u)_i^{t-1} + \lambda_i F(u)_i.$$

We update the model at each time step. So, our adaptive classifier update can keep the first model and take new models which learn the target appearance with occlusion and other changes into consideration, thus making our tracker more robust to appearance variation and occlusion.

4. Experiments

To evaluate the performance of our proposed tracker, we test our tracker on 19 publicly available video sequences. The sequences come from VTD dataset, MIL dataset, and visual tracker benchmark [25, 26], some of which cover most challenging situations in visual tracking: heavy occlusion, complicated motion, large appearance variation, and so forth. For comparison, we run 11 leading algorithms with the same initial position of the target in the same video sequences. These algorithms are Tracking-Learning-Detection (TLD) [11], visual tracking decomposition (VTD) [3], ℓ_1 tracking ($\ell_1\text{T}$) [4], compressive tracking (CT) [12], distribution field tracking (DFT) [5], multiple instance learning (MIL) [10], incremental visual tracking (IVT) [2], fragments-based tracking (Frag) [1], Randomized Ensemble Tracking [9], Sparsity-based Collaborative Model (SCM) [15], and Online AdaBoost Boosting (OAB) [8]. Here, MIL, OAB, CT, TLD, and RET are discriminate trackers, and IVT, DFT, Frag, VTD, and $\ell_1\text{T}$ are generative trackers. OAB and ET particularly are our baselines. Compared with these different kinds of methods, we demonstrate the effectiveness and robustness of our method. The proposed algorithm is implemented in Matlab2011b on a PC with an Intel(R) Core(TM) i5-4590u processor and 4 G RAM. For fairness, we use the publicly available source or binary codes and tracking result provided by the authors.

Note that we fix the parameters of our tracker for all sequences to demonstrate its robustness and effectiveness. We search for the target of interest in a standard sliding-window method, and the object bounding box is divided into a regular grid of $8 * 8$ small patches. For each patch,

we extract its standard histogram of gradients (HOG), which yields a 100-dimensional feature vector. Each weak classifier corresponding to the local patch is a standard linear SVM, which is trained with its own buffer of 49 positive and 50 negative examples. The sample buffers and the weak classifiers are initialized with the ground truth bounding box in the first frame. During tracking, whenever a new example is added to the buffer, the weak classifier is retrained. The combination between HOG and SVM performs well in computer vision, which is proved by Bernt Schiele [24]. The threshold ε_0 in Section 3.2 is set to be 0.15, and the threshold ε_1 in Section 3.3 is set to be 0.20 according to the experiments.

4.1. Quantitative Evaluation. To quantitatively evaluate the performance of each tracker, we use three widely accepted evaluation metrics including the successful tracking rate (STR), the average center location error (ACLE), and the average overlap rate (AOR). The successful tracking rate is the ratio of the number of successful tracking frames and the number of the sequence. If the overlap between the predicted and ground truth bounding box is more than 0.5, we label the frame as the successful tracking frame. The center location error is the Euclidean distance between the center of the tracking result and the ground truth for each frame. The overlap rate is based on the Pascal VOC criteria. Given the tracked bounding box ROI_T and the ground truth bounding box ROI_{GT} , the overlap rate is computed by $\text{area}(ROI_T \cap ROI_{GT}) / \text{area}(ROI_T \cup ROI_{GT})$. In order to evaluate the tracking performance of the algorithms, we compute the average center location error and the average overlap rate across all frames of each video sequence as done in most tracking literature. For fairness, we evaluate our proposed algorithm by taking the average over five runs. Due to the space limitation, we list the results of 11 leading algorithms. Tables 1 and 2 report the quantitative comparison results, respectively.

Tables 1 and 2 show AOR, STR, and ACLE obtained by the tracking algorithms with different similarity metrics on the 19 video sequences; our proposed algorithm achieves the first best overall performance at 12 sequences and achieves the second best overall performance at 2 sequences according to the STR metric. Compared with RET, our proposed algorithm achieves the better overall performance at 17 sequences excluding coke and basketball sequences with the situations of rotation and fast motion. In the girl, singer1, singer2, shaking sequences with heavy occlusion, and appearance variation, the proposed algorithm performs better overall performance than RET due to the fact that the adaptive classifier updates by combining the state of occlusion and appearance variation at each time step. Compared with other methods, our proposed algorithm achieves the first three best overall performances almost against all the sequences and is more effective and robust than other state-of-the-art algorithms, especially when there is significant appearance variation and heavy occlusion.

4.2. Qualitative Evaluation. We also plot some tracking results of 11 trackers on 12 video sequences for qualitative

comparison as shown in Figure 3. The results are discussed based on the main challenging factors such as occlusion and appearance variation.

Occlusion. In the sequences of david, faceoccl, faceocc2, tiger1, skating1, liquor, and singer1, occlusion is the main challenge. From Figure 3(b), except for TLD and our tracker, other algorithms fail to track the target at frames 1372 and 1478 after the target was occluded. At frame 1584, our method and RET locate the liquor; others are not able to relocate the bottle. In the tiger1 sequence, when the tiger is occluded by the distractor, all other trackers fail to locate the object, but our proposed method handles the problem well and tracks the target accurately, as shown in Figure 3(k). In the other sequences with occlusion, our tracker achieves at least the second best performance through the entire sequence. Once the target is partially occluded or heavily occluded, our method can be able to relocate the target due to the adaptive classifier update method and our method takes occlusion into consideration.

Scale Variation. The shaking, singer1, and skating1 sequences contain significant scale change. In the skating1 sequence, a player is occluded by other players at frame 206, and CT, $\ell 1T$, and IVT lost to track the player. All the other approaches drifted at frames 289 and 339, but our tracker, VTD, and RET successfully locate the target and our tracker locate it better than VTD at frame 339, as shown in Figure 3(i). From Figure 3(g), CT and Frag failed to track the singer at frame 145; the others can look onto the target. At frames 220 and 324, the scale of the singer changes significantly. While most trackers lost tracking the singer, TLD, RET, and our tracker can track the object. Our tracker achieves the best and the second best performance in the singer1 and skating1 sequence, respectively.

Shape Change. In the shaking, david, singer2, and tiger1 sequences, the objects undergo heavy shape variation, especially in the shaking and singer2. From Figure 3(a), at frame 48 all the other methods start to drift, but our proposed approach locates the singer well. At frame 195, only Frag and our tracker successfully track the target well. With heavier shape change, Frag fail to track the singer at frame 284, while our tracker still tracks the target object well. In the shaking sequence, due to the heavy shape variation caused by sharp illumination change, OAB, CT, $\ell 1T$, and Frag lost the target at frame 60. At frame 119, IVT, DFT, and TLD also lost the object, but SCM, MIL, VTD, and ours can track the target and locate it accurately. With heavier shape change, only MIL, SCM, and ours can track the target, as shown in Figure 3(l).

As mentioned in scale variation and shape change, our tracker can handle appearance variation better than the other methods.

5. Conclusion

Based on RET, we propose a novel adaptive classifier update method. Instead of traditional model update, our proposed method takes appearance variation and possible occlusion

TABLE 1: AOR (STR). The best two results are shown in bold and italic fonts. The results of some algorithm come from visual tracker benchmark [25].

Sequence	TLD	CT	Frag	MIL	SCM	DFT	OAB	IVT	ℓ IT	VTD	RET	Ours
Deer	0.60 (0.73)	0.04 (0.04)	0.17 (0.21)	0.12 (0.13)	0.07 (0.03)	0.26 (0.31)	0.72 (0.96)	0.03 (0.03)	0.60 (0.72)	0.06 (0.04)	0.72 (1)	0.71 (1)
David	0.72 (0.97)	0.50 (0.43)	0.17 (0.12)	0.43 (0.23)	0.72 (0.92)	0.30 (0.23)	0.39 (0.15)	0.65 (0.80)	0.54 (0.70)	0.56 (0.68)	0.81 (1)	0.84 (1)
Faceocc	0.59 (0.83)	0.64 (0.85)	0.82 (1)	0.60 (0.77)	0.79 (1)	0.69 (0.80)	0.66 (0.91)	0.73 (0.98)	0.75 (1)	0.68 (0.93)	0.86 (1)	0.87 (1)
Faceocc2	0.62 (0.83)	0.61 (0.74)	0.65 (0.76)	0.67 (0.94)	0.73 (0.87)	0.77 (1)	0.60 (0.75)	0.73 (0.91)	0.69 (0.80)	0.74 (0.99)	0.80 (1)	0.81 (1)
Girl	0.57 (0.76)	0.31 (0.18)	0.45 (0.54)	0.40 (0.29)	0.68 (0.88)	0.56 (0.25)	0.72 (0.94)	0.17 (0.19)	0.73 (0.97)	0.55 (0.65)	0.69 (0.84)	0.69 (0.90)
Sylvester	0.67 (0.93)	0.67 (0.83)	0.58 (0.69)	0.53 (0.55)	0.68 (0.89)	0.38 (0.41)	0.56 (0.68)	0.52 (0.68)	0.40 (0.43)	0.62 (0.80)	0.61 (0.80)	0.61 (0.80)
Tiger1	0.38 (0.46)	0.41 (0.25)	0.27 (0.31)	0.12 (0.10)	0.16 (0.13)	0.53 (0.68)	0.11 (0.10)	0.10 (0.09)	0.31 (0.35)	0.12 (0.12)	0.74 (0.92)	0.75 (0.94)
Tiger2	0.26 (0.17)	0.45 (0.37)	0.12 (0.12)	0.46 (0.45)	0.09 (0.11)	0.68 (0.81)	0.15 (0.14)	0.09 (0.08)	0.25 (0.27)	0.30 (0.17)	0.73 (0.96)	0.73 (0.96)
Coke	0.40 (0.29)	0.23 (0.10)	0.04 (0.03)	0.20 (0.12)	0.33 (0.37)	0.11 (0.09)	0.33 (0.17)	0.12 (0.13)	0.18 (0.20)	0.14 (0.14)	0.37 (0.23)	0.36 (0.22)
Singer1	0.73 (0.99)	0.35 (0.25)	0.20 (0.22)	0.36 (0.28)	0.87 (1)	0.36 (0.48)	0.34 (0.23)	0.60 (0.68)	0.29 (0.38)	0.49 (0.43)	0.73 (0.97)	0.80 (1)
Singer2	0.21 (0.10)	0.08 (0.08)	0.20 (0.20)	0.51 (0.48)	0.17 (0.16)	0.63 (0.70)	0.05 (0.03)	0.04 (0.04)	0.04 (0.04)	0.42 (0.45)	0.38 (0.50)	0.59 (0.72)
Football	0.49 (0.41)	0.61 (0.79)	0.70 (0.92)	0.59 (0.74)	0.49 (0.59)	0.66 (0.84)	0.34 (0.37)	0.56 (0.72)	0.55 (0.67)	0.56 (0.77)	0.62 (0.82)	0.64 (0.86)
Lemming	0.53 (0.59)	0.55 (0.68)	0.31 (0.41)	0.65 (0.81)	0.14 (0.17)	0.41 (0.47)	0.59 (0.74)	0.14 (0.17)	0.14 (0.17)	0.44 (0.49)	0.64 (0.82)	0.64 (0.82)
Liquor	0.52 (0.58)	0.20 (0.21)	0.33 (0.37)	0.22 (0.20)	0.32 (0.32)	0.22 (0.23)	0.45 (0.48)	0.23 (0.21)	0.20 (0.32)	0.50 (0.58)	0.69 (0.96)	0.73 (0.97)
Skating1	0.20 (0.23)	0.09 (0.10)	0.13 (0.12)	0.13 (0.10)	0.47 (0.42)	0.14 (0.16)	0.40 (0.34)	0.07 (0.08)	0.10 (0.13)	0.53 (0.57)	0.48 (0.52)	0.49 (0.54)
Shaking	0.39 (0.40)	0.10 (0.04)	0.08 (0.07)	0.43 (0.23)	0.69 (0.90)	0.64 (0.83)	0.01 (0.01)	0.03 (0.01)	0.08 (0.04)	0.71 (0.94)	0.44 (0.53)	0.60 (0.74)
Waking	0.45 (0.38)	0.52 (0.50)	0.54 (0.51)	0.55 (0.54)	0.71 (0.96)	0.56 (0.55)	0.54 (0.48)	0.77 (1)	0.75 (1)	0.61 (0.81)	0.77 (1)	0.77 (1)
Soccer	0.12 (0.11)	0.12 (0.13)	0.08 (0.14)	0.25 (0.20)	0.27 (0.18)	0.17 (0.20)	0.09 (0.08)	0.13 (0.16)	0.24 (0.20)	0.35 (0.21)	0.27 (0.30)	0.31 (0.32)
Basketball	0.11 (0.07)	0.30 (0.18)	0.62 (0.61)	0.24 (0.26)	0.07 (0.06)	0.28 (0.35)	0.04 (0.05)	0.17 (0.08)	0.20 (0.10)	0.73 (0.81)	0.54 (0.64)	0.32 (0.35)

TABLE 2: ACLE (STR). The best two results are shown in bold and italic fonts. The results of some algorithm come from visual tracker benchmark [25].

Sequence	TLD	CT	Frag	MIL	SCM	DFT	OAB	IVT	$\ell 1T$	VTD	RET	Ours
Deer	31 (0.73)	246 (0.04)	105 (0.21)	101 (0.13)	104 (0.03)	99 (0.31)	7 (0.96)	183 (0.03)	24 (0.72)	135 (0.04)	7 (1)	8 (1)
David	5 (0.97)	10 (0.43)	82 (0.12)	17 (0.23)	4 (0.92)	43 (0.23)	22 (0.15)	5 (0.80)	14 (0.70)	12 (0.68)	6 (1)	5 (1)
Faceocc	28 (0.83)	26 (0.85)	11 (1)	30 (0.77)	13 (1)	24 (0.80)	25 (0.91)	18 (0.98)	17 (1)	20 (0.93)	7 (1)	6 (1)
Faceocc2	12 (0.83)	19 (0.74)	16 (0.76)	14 (0.94)	9 (0.87)	8 (1)	20 (0.75)	7 (0.91)	13 (0.80)	8 (0.99)	9 (1)	9 (1)
Girl	10 (0.76)	19 (0.18)	21 (0.54)	14 (0.29)	3 (0.88)	24 (0.25)	4 (0.94)	22 (0.19)	3 (0.97)	9 (0.65)	19 (0.84)	17 (0.90)
Sylvester	7 (0.93)	9 (0.83)	15 (0.69)	15 (0.55)	8 (0.89)	45 (0.41)	15 (0.68)	34 (0.68)	26 (0.43)	20 (0.80)	12 (0.80)	12 (0.80)
Tiger1	49 (0.46)	30 (0.25)	74 (0.31)	109 (0.10)	93 (0.13)	41 (0.68)	95 (0.10)	107 (0.09)	58 (0.35)	108 (0.12)	4 (0.92)	4 (0.94)
Tiger2	38 (0.17)	28 (0.37)	114 (0.12)	27 (0.45)	141 (0.11)	12 (0.81)	252 (0.14)	105 (0.08)	65 (0.27)	41 (0.17)	4 (0.96)	4 (0.96)
Coke	25 (0.29)	40 (0.10)	125 (0.03)	47 (0.12)	57 (0.37)	71 (0.09)	36 (0.17)	83 (0.13)	50 (0.20)	69 (0.14)	13 (0.23)	13 (0.22)
Singer1	8 (0.99)	16 (0.25)	89 (0.22)	16 (0.28)	3 (1)	19 (0.48)	13 (0.23)	11 (0.68)	53 (0.38)	4 (0.43)	7 (0.97)	5 (1)
Singer2	58 (0.10)	127 (0.08)	89 (0.20)	23 (0.48)	113 (0.16)	22 (0.70)	186 (0.03)	176 (0.04)	181 (0.04)	44 (0.45)	82 (0.50)	20 (0.72)
Football	14 (0.41)	11 (0.79)	5 (0.92)	12 (0.74)	16 (0.59)	9 (0.84)	72 (0.37)	14 (0.72)	15 (0.67)	14 (0.77)	7 (0.82)	7 (0.86)
Lemming	16 (0.59)	32 (0.68)	127 (0.41)	12 (0.81)	186 (0.17)	78 (0.47)	18 (0.74)	182 (0.17)	178 (0.17)	79 (0.49)	16 (0.82)	16 (0.82)
Liquor	38 (0.58)	186 (0.21)	100 (0.37)	142 (0.20)	99 (0.32)	221 (0.23)	69 (0.48)	119 (0.21)	213 (0.32)	60 (0.58)	13 (0.96)	11 (0.97)
Skating1	145 (0.23)	150 (0.10)	149 (0.12)	139 (0.10)	16 (0.42)	174 (0.16)	43 (0.34)	247 (0.08)	159 (0.13)	9 (0.57)	27 (0.52)	16 (0.54)
Shaking	37 (0.40)	80 (0.04)	192 (0.07)	24 (0.23)	11 (0.90)	26 (0.83)	192 (0.01)	86 (0.01)	110 (0.04)	9 (0.94)	49 (0.53)	15 (0.74)
Waking	10 (0.38)	7 (0.50)	9 (0.51)	3 (0.54)	2 (0.96)	6 (0.55)	5 (0.48)	2 (1)	3 (1)	6 (0.81)	3 (1)	2 (1)
Soccer	80 (0.11)	80 (0.13)	111 (0.14)	37 (0.20)	38 (0.18)	129 (0.20)	92 (0.08)	201 (0.16)	85 (0.20)	21 (0.21)	93 (0.30)	41 (0.32)
Basketball	137 (0.07)	57 (0.18)	13 (0.61)	82 (0.26)	294 (0.06)	120 (0.35)	143 (0.05)	180 (0.08)	115 (0.10)	6 (0.81)	22 (0.64)	93 (0.35)

FIGURE 3: Comparison of our method with 11 trackers on 12 image sequences with challenging situations.

into consideration at each time step. We estimate the appearance variation by using sparse optical flow method and estimate the possible occlusion by using sparsity-based occlusion estimation method. Then we combine the appearance variation and occlusion estimation to adaptively update the classifier and model. Extensive experimental results and evaluations against several state-of-the-art methods on some challenging video sequences demonstrate the effectiveness and robustness of our proposed algorithm. In the future, we are interested in extracting some kinds of features and fuse them. Besides, we will explore the integration of a strong motion prediction model to solve the problem of tracking-by-detection method which is sensitive to the gating parameters.

Conflict of Interests

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

Acknowledgments

This work was supported in part by Natural Science Foundation of China (no. 61272195, 61472055, and U1401252), Program for New Century Excellent Talents in University of China (NCET-11-1085), Chongqing Outstanding Youth Found (cstc2014 jcyj40001), and Chongqing Research Program of Application Foundation and Advanced Technology (cstc2012 jjA40036).

References

- [1] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 798–805, New York, NY, USA, June 2006.
- [2] 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.
- [3] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 1269–1276, IEEE, San Francisco, Calif, USA, June 2010.
- [4] X. Mei and H. Ling, "Robust visual tracking using ℓ_1 minimization," in *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV '09)*, pp. 1436–1443, Kyoto, Japan, September 2009.
- [5] L. Sevilla-Lara and E. Learned-Miller, "Distribution fields for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1910–1917, Providence, RI, USA, June 2012.
- [6] X. Li, A. Dick, C. Shen, A. van den Hengel, and H. Wang, "Incremental learning of 3D-DCT compact representations for robust visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 863–881, 2013.
- [7] S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261–271, 2007.
- [8] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proceedings of the British Machine Vision Conference (BMVC '06)*, pp. 47–56, Edinburgh, UK, September 2006.
- [9] Q. Bai, Z. Wu, S. Sclaroff, M. Betke, and C. Monnier, "Randomized ensemble tracking," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 2040–2047, Sydney, Australia, December 2013.
- [10] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 983–990, IEEE, Miami, Fla, USA, June 2009.
- [11] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.
- [12] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Computer Vision—ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., vol. 7574 of *Lecture Notes in Computer Science*, pp. 864–877, Springer, Berlin, Germany, 2012.
- [13] S. Hare, S. Golodetz, A. Saffari et al., "Struck: structured output tracking with kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.
- [14] J. Wang, H. Z. Wang, and Y. Yan, "Robust visual tracking by metric learning with weighted histogram representations," *Neurocomputing*, vol. 153, pp. 77–88, 2015.
- [15] Z. Wei, L. Huchuan, and Y. Ming-Hsuan, "Robust object tracking via sparsity-based collaborative model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1838–1845, 2012.
- [16] T. B. Dinh and G. Medioni, "Co-training framework of generative and discriminative trackers with partial occlusion handling," in *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV '11)*, pp. 642–649, Kona, Hawaii, USA, January 2011.
- [17] N. Nourani-Vatani, P. V. K. Borges, J. M. Roberts, and M. V. Srinivasan, "On the use of optical flow for scene change detection and description," *Journal of Intelligent & Robotic Systems: Theory & Applications*, vol. 74, no. 3–4, pp. 817–846, 2014.
- [18] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, 2011.
- [19] B. Huang, J. Chen, Y. Wang, C. Liang, Z. Wang, and K. Sun, "Sparsity-based occlusion handling method for person re-identification," in *MultiMedia Modeling*, X. He, S. Luo, D. Tao, C. Xu, J. Yang, and M. Hasan, Eds., vol. 8936 of *Lecture Notes in Computer Science*, pp. 61–73, Springer International, 2015.
- [20] N. Nourani-Vatani, P. V. K. Borges, J. M. Roberts, and M. V. Srinivasan, "Topological localization using optical flow descriptors," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV '11)*, pp. 1030–1037, IEEE, Barcelona, Spain, November 2011.
- [21] N. Nourani-Vatani, P. V. K. Borges, and J. M. Roberts, "A study of feature extraction algorithms for optical flow tracking," in *Proceedings of the Australasian Conference on Robotics and Automation (ACRA '12)*, Wellington, New Zealand, December 2012.
- [22] H. Zhao, X. Wang, and M. Liu, "Robust object tracking with occlusion handling based on local sparse representation," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 7, no. 3, p. 407, 2014.

- [23] J. Xu, L. Huchuan, and Y. Ming-Hsuan, "Visual tracking via adaptive structural local sparse appearance model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 1822–1829, 2012.
- [24] T. Zhang, K. Jia, C. Xu, Y. Ma, and N. Ahuja, "Partial occlusion handling for visual tracking via robust part matching," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1258–1265, IEEE, Columbus, Ohio, USA, June 2014.
- [25] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: a benchmark," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2411–2418, Portland, Ore, USA, June 2013.
- [26] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: an experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014.

Hindawi

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