IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 13, 2016, accepted January 25, 2017, date of publication March 1, 2017, date of current version October 25, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2675438

Robust Adaptive Fusion Tracking Based on Complex Cells and Keypoints

SIXIAN CHAN[1], XIAOLONG ZHOU[1], (Member, IEEE), SHENGYONG CHEN[1,2], (Senior Member, IEEE)

[1]College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China
[2]College of Computer Science, Tianjin University of Technology, Tianjin 300384, China

Corresponding authors: X. Zhou (zxl@zjut.edu.cn) and S. Chen (sy@ieee.org)

This work was supported in part by the National Natural Science Foundation of China under Grant 61403342, Grant 61273286, Grant U1509207, and Grant 61325019 and in part by the Hubei Key Laboratory of Intelligent Vision Based Monitoring for Hydroelectric Engineering under Grant 2014KLA09.

ABSTRACT Although many successful algorithms have been proposed for visual tracking, it is still a challenging task due to occlusion, scale variation, fast motion, and deformation. To handle these challenges, we propose a collaborative model and focus on three key factors: 1) an effective representation to consider appearance variations; 2) an effective application of the keypoints; and 3) an incorporation of contextual information. In this paper, we propose a novel algorithm that takes into account the three key factors based on complex cells and keypoints. The complex cells can effectively explore the contextual information at multiple scales. Meanwhile, a keypoint is an ideal local representation. Keypoints-based tracking method is used to make coarse tracking. A precise tracking-by-detection whose samples come from keypoints-based tracking is followed by considering the scale information. In addition, measurement of appearance variation is measured by matching the current inner cell with template's individualistically. In the basis of the measurement, an adaptive learning rate parameter is estimated for updating the object appearance model to avoid noises. Experimental results demonstrate that our tracker is able to handle appearance variations and recover from drifts. In conjunction with tracking acceleration modules, the proposed method performs in real time and outperforms favorably many state-of-the-art algorithms for object tracking.

INDEX TERMS Computer vision, visual tracking, complex cells and keypoints, adaptive fusion tracking.

I. INTRODUCTION

Object tracking is one of the most important topic in computer vision. It has been used for wide applications, including intelligent surveillance, motion classification, recognition, autonomous robots and so on. The tracking of a priori unknown objects draws considerable interest and establishes its place in the tracking community under the template-matching visual tracking. Generally speaking, the initial region in the first frame of the image sequence is the only information that is provided to the tracker and the tracking task is to estimate the current position of the target. However, this task remains challenging because some situations like illumination variations, shape deformation, occlusion, or motion changes are hard to handle.

To well handle the above mentioned challenges, a descriptor of the target in a tracker is very important and should be robust enough to distinguish the object under appearance variations. Hence, most tracking algorithms [1]–[9] focus on the appearance representation and treat tracking as template matching in each frame in recent years. We call them tracking-by-matching. Keypoints-based or part-based appearance model is the kernel of tracking-by-matching method. Researchers have developed sophisticated methods of adapting these models during tracking. One solution is to use the descriptors of keypoints to represent an object appearance model while voting those tracked or matched keypoints to estimate the target location, for example, metric learning tracker [10], structured output learning tracker [11], consensus-based matching-tracking algorithm [4] and multi-store tracker [7]. Another method is to build a robust object appearance template and find the best candidate image patch to match the template, such as, incremental learning tracker [12], fragments-based tracker [13] and visual tracking decomposition [14]. Last approach is to model both the object and the background, and then to distinguish the object from the background using a discriminative classifier [15], [16] or

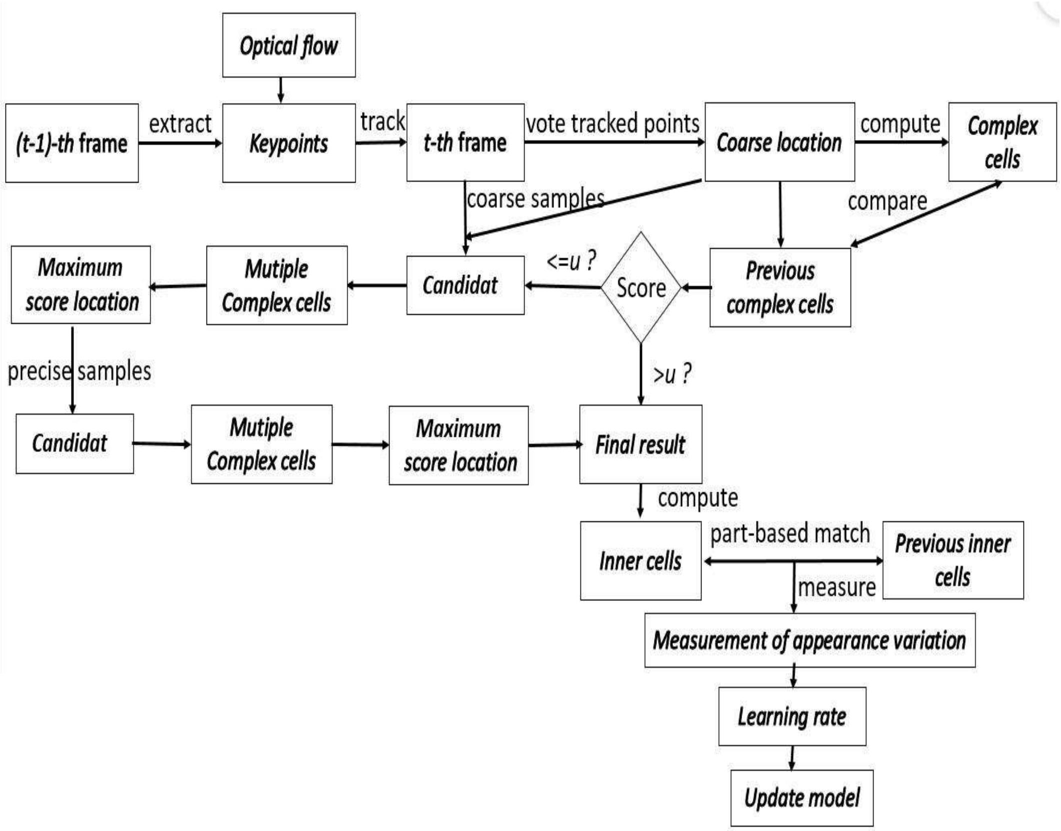


FIGURE 1. The framework of the proposed RAFT Tracking. System is divided into three components in accordance with the traditional way: object representation, tracking-by-matching and template updating.

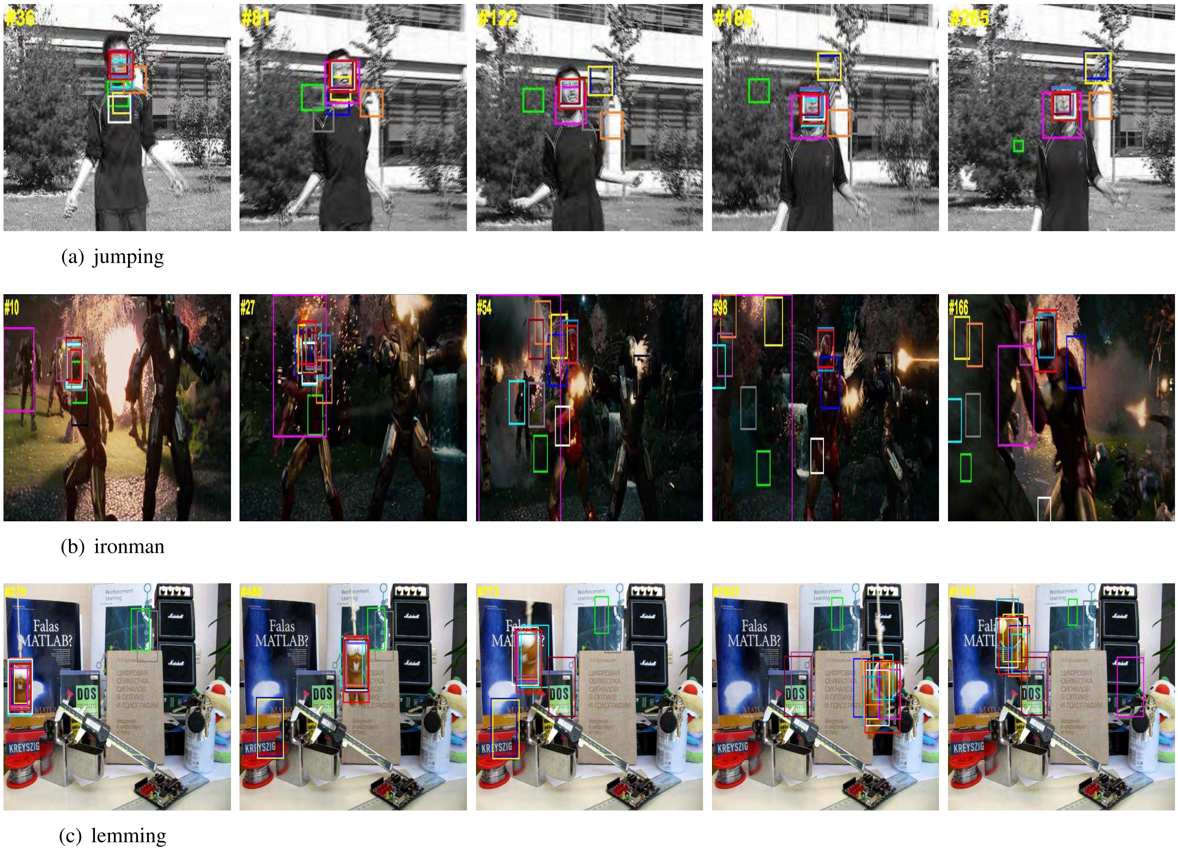
match the mixed template to calculate the maximum response value from candidates [2].

However, keypoint descriptor, a local representation or a global representation can only considers one aspect of above demands. Generally, keypoint descriptor and local representations represent the local regions information. They are usually robust to handle variation in partial occlusions or motion [13], [17], but easily fail in background clutters and heavy occlusion because of ignoring object structures and losing object information. On the contrary, global representations can search for large object's structure [15], [18] but absence of the local adaptivity makes them not conductive to handle occlusion or deformation. In addition, three kinds of representations only extract the information from the object. However, the contextual features from surrounding background are ignored absolutely, which can be applied for accurate tracking as well as variations in occlusions or deformations.

In this paper, we develop a simple tracking system framework and propose a robust adaptive fusion tracking method, named RAFT, which effectively integrates the three kinds of representations based on complex cells and keypoints. The framework is demonstrated in Fig. 1 and divided into three components in accordance with the traditional way: object representation, tracking-by-matching and template updating.

For object representation, to maintain the local stability, local region histogram features are adopted as the bases of our representation, named cells. These cells are formed by dividing the region within and around the bounding box into disjoint rectangular patches. To achieve the global robustness, similar to CCT tracker [2], different cells are integrated together, called complex cells. With the cells of different spatial arrangements, there is an effective exploration for the context information via complex cells at multiple scales. There are five types of complex cells that collect the relativity from local region, block neighbourhood, inter-region relations and surrounding background. In addition, to enhance the global distinctiveness, we add a new complex cell into the group of object representation. The details refer to Sec. III-B. Some complementary characteristics are highlighted by composing the five kinds of complex cells. Both inner and outer object information are employed and a multi-scale appearance is taken into account to balance the conflict between local flexibility with global stability. As a derivative, we extract the center of each inner cells as keypoints for coarse tracking in the next stage.

While taking account to tracking-by-matching, it is a two-stage approach. First, as we know, keypoint is an ideal local representation. Keypoints-based tracking method is wonderful about handling these model. We use the



(a) jumping

(b) ironman

(c) lemming

FIGURE 2. Comparison of our approach with state-of-the-art trackers in challenging situations. The example frames are the jumping, ironman and lemming sequences respectively. Qualitative comparison results of our approach with 12 state-of-the-art trackers in challenging situations. The example frames are the boy, bolt and motorrolling sequences respectively, in which the targets undergo motion blur and low resolution. Best viewed on color display. The results of IVT [12], VTD [14], CCT [2], VTS [21], CXT [22], CMT [4], CT [23], Struck [24], MEEM [25], MUSTer [7], CNT [26], SCM [27] and our RAFT are represented by green, blue, black, yellow, carmine, ultramarine, orange, purple, turquoise, white, crimson, grey and red boxes respectively.

LK optical-flow [19] based on keypoint for coarse tracking. After this, we achieve the final tracking result or provide more accurate parameter for next precise tracking. Second, we adopt the strategy proposed by Chen *et al.* [2], in which the matching template is composed of temporal varying cells. There are two layers which store the appearance of the target and background for matching respectively. Each cell is deemed as Gaussian distribution. The two-layer template is conductive to search contextual information. The background-weighted histogram algorithm proposed by Ning [20] to reduce the interference of background in target localisation in mean-shift tracking. Inspired by this inspiration, we track the object by matching the complex cells from candidates with the template for finding the maximal similarity. The state with maximal response means that is most similar to the template and thus can be treated as the tracked location.

While considering about model updating, we focus on the situation that the learning rate is changing with the corresponding object's appearance variation to avoid noises effectively. In order to pursue the mentioned ideal situation, we propose a novel strategy by matching the inner cells between current object with the template's individualistically. Each inner cell is belong to one part of the target. The success of each pair of cells match depends on the similarity between them. In other words, there is evidence that proves the phenomenon which the final matching result demonstrates the degree of the appearance variation of the target. Then a double sigmoid function [28] is employed to measure the learning rate parameter. Fig. 2 shows that our tracker achieves promising results as compared with other state-of-the-art methods and the key contributions of the proposed RAFT algorithm are summarized as follows.

- We develop a simple tracking system framework and propose a robust adaptive fusion tracking method, which effectively integrates the three kinds of representations based on complex cells and keypoints.
- We incorporate a novel complex cell into the group of object representation to enhance the global distinctiveness.
- We propose a novel strategy by matching the inner cells individualistically to measure the degree of the appearance variations.
- We present a new model update mechanism based on the measurement of appearance variations to preserve the stable features while avoid the noisy ones.

- We adopt a new coarse-to-fine search strategy to make the finally estimated location more accurate.

II. RELATED WORK

A. PARTS-BASED OBJECT TRACKING

There has been a considerable amount of work dedicated to parts-based object representations under tracking-by-matching. Adam *et al.* [13] propose a fundamental from of parts-based tracker. The object is divided into multiple image patches. The patches are assigned in a designed grid, and each patch votes for the location of the object inside a sliding-window. It is impossible to be permitted for rotating or articulating of the object under the rigid arrangement of those patches. Wu *et al.* [29] employ the theory of Markov networks to present a part-based method for visual tracking. The method is robust to appearance variations in occlusion and deformation due to the difference of the parts are been checked while rashly fusing all descriptors into a single one. Nejhum *et al.* [30] propose a tracker with part-based appearance model of the object. The foreground object is segmented into a small number of rectangular blocks. The algorithm aims at tracking the object by matching blocks intensity histograms, and updating the part-based appearance model. However, these methods require manual initialisation of part locations carefully and do not use the context information. To overcome those problems, Chen *et al.* [2] present a part-based tracker with a novel representation framework. They construct complex cells from local descriptors to represent multiple scale and contextual information. Using fixed learning parameter seems to be a only fly in the ointment to update its appearance model.

B. KEYPOINT-BASED OBJECT TRACKING

Efficient keypoint-based object tracking methods are utilized in lots of real-time visual applications. Hare *et al.* [11] present a keypoint-based approach for object tracking. The method learns a model for real-time keypoint-based object detection then matches those points for tracking. However, to combine feature learning, matching and pose estimation into a single unified framework is hard to balance the conflicts between them. Kala *et al.* [31] propose a tracking-learning-detection framework (TLD) for long-term tracking. This algorithm combines the traditional tracking algorithm with the traditional detection algorithm to solve the problem of deformation and drifts during tracking. While it cannot handle the heavy occlusion and lose the tracking result without returning bounding box. Nebehay *et al.* [4] reveal a novel keypoint-based method for long-term model-free object tracking in a combined matching-and-tracking (CMT) framework. The approach uses a keypoint-based object representation and treats the tracking as finding the corresponding keypoints in each frame. The algorithm is especially sensitive to the number of keypoints. If the number of keypoints is too large, it would lead to a heavy computational cost. On the contrary, too small number of keypoints will result in losing the target. Without the process of updating the model, when the object's angle changes drastically, the tracker will not match the keypoints and lead to the failure of tracking. Hong *et al.* [7] propose a novel multi-store tracker (MusTer) based on a cognitive psychology inspired for object tracking. The RANSAC estimation interacts with the keypoint descriptors in the long-term store and controls the final output and the short-term memory states. Due to the excellent performance of this method, we make a detailed analysis. A flexible description is designed to present the target, which is proved to adapt to the variances of object appearance during tracking. Meanwhile, a dual-component approach consisting of short- and long-term memory stores is proposed to deal with object appearance memories. A powerful and efficient Integrated Correlation Filter is also applied in the short-term store for short-term tracking. According to keypoint matching-tracking and RANSAC estimation, the long-term measure can interact with the long-term memory and supply enough information for controlling the tracking. Even though a reasonable keypoint feature database size is not easy to maintain, the MUSTer is still the best algorithm based on keypoint matching-tracking up to now.

III. ROBUST ADAPTIVE FUSION TRACKING

A. KEYPOINTS-BASED COARSE TRACKING

Decomposition of the target as parts makes models more robust for object tracking, since local changes only affect individual parts. Meanwhile, keypoint is an ideal local representation, which is utilized for coarse tracking in this paper. For component of keypoints-based tracking, we estimate the displacement of each keypoints in K_{t-1} from I_{t-1} to I_t by employing the Optical-Flow [19] tracker of Lucas and Kanade. For $t = 2$, points K_1 are obtained by extracting the center point of each inner cell. As shown in the left of Fig. 3, the inner cells are located within the red object's bounding box and the keypoints center in these cells.

Median-Flow tracker [32] presents the target by a bounding box and estimates its motion between consecutive frames. Internally, the set of tracked keypoints T is estimated based on their reliability, and voted with half of the most reliable displacements for the motion of the bounding box using the median. As shown in Fig. 3, the green bounding box is the result of the keypoints tracking. Then we extract the complex cells feature at location of the bounding box and calculate the similarity between current sample and previous template. A score function (Eq.3) is used to measure the similarity. After this, we extract the complex cells at location of the LK tracking and calculate the similarity with the template. Algorithm process lies in the three kinds of results corresponding to the similarity. If the similarity is beyond the threshold value, we will treat the initial result as the final result and get out of the loop to track the next frame to speed up the algorithm. If the similarity is lower than the threshold value, the result of LK tracking will be regarded as the initial parameter. Meanwhile, a more accurate sampling area can be provided for the next precise tracker. Moreover, if the similarity is quite beyond the pale, the preliminary result

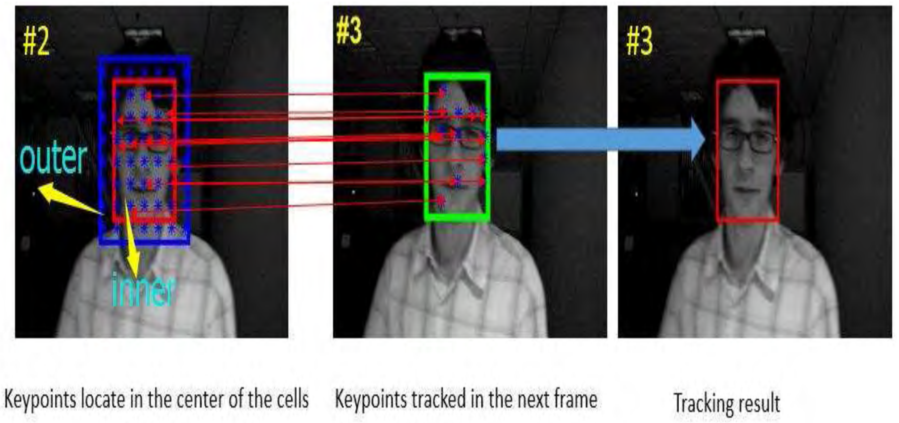

#2
#3
#3
outer
inner
Keypoints locate in the center of the cells
Keypoints tracked in the next frame
Tracking result

FIGURE 3. The process of the Keypoints-based Tracking. left: The initial keypoints located in the center of each inner cell. middle: Keypoints tracked by LK [19]. right: The result of tracking.

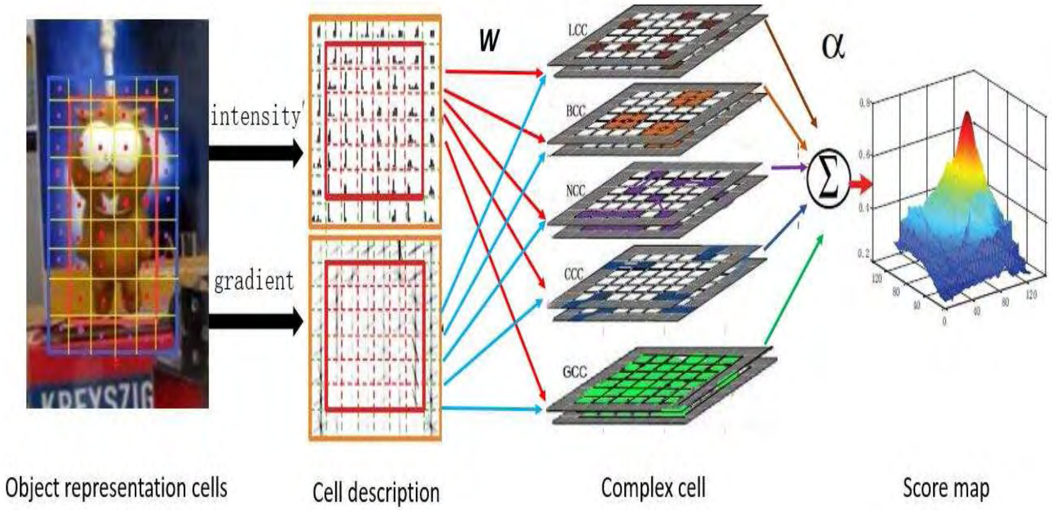


FIGURE 4. Overview of Complex Cells-based Precise Tracking. left: Object segment into cells; middle left: An intensity histogram (I) and an oriented gradient histogram (G) are combined to describe each cell; middle right: Five types complex cells composed of cells; right: Fusion score calculated from different complex cells with a linear function.

will be omitted to avoid noises. Inspired by the TLD [31] method, our algorithm also increases the component with failure detection. Hence, the third case generally does not appear. The more details refer to Sec. IV-D.

B. COMPLEX CELLS-BASED PRECISE TRACKING

Complex cells tracking (CCT) [2] is a template-matching method for object tracking, where the template is consisted of temporal complex cells and has two layers template to search the target and background appearance feature respectively. Due to the different combinations of complex cells, the template can be well explored in the context of information at multiple scales for visual tracking, which is conductive to improve the tracking performance. Fig. 4 demonstrates the overview of the complex cells-based tracking. The whole system is divided into three stages, namely, the extraction and expression of the cell region of the object, the composition of the complex cells, and finally the template matching.

1) EXTRACTION AND EXPRESSION OF THE CELL

An object is represented by a bounding box at frame t. Its element $X_t = (x_t; y_t; s_t)$ is a three-dimensional vector indicating position and scale. To obtain the local robustness, local region histogram features are adopted as the bases of our representation, named cells. These cells are formed by dividing the region within and around the bounding box into disjoint rectangular patches, written as M. As shown in the left of Fig. 4, the cells inside the red bounding box are called inner cells while the others are called outer cells. By the way, the keypiont is also determined and located at the center of each cell, corresponding to the red and purple dots in the left of Fig. 4. $M_{all} = M_{in} \bigcup M_{out}$, where the M_{in} and M_{out} are set as the inner cell and the outer cell respectively, with forming a whole M_{all}. An intensity histogram (I) and an oriented gradient histogram (G) are combined to describe each cell. Intensities stand for gray values with gamma normalization. The gradient feature is similar to HOG [33]. Both kinds of histograms contain 8 bins. The descriptor for cell m is a 16 dimensional vector obtained by combining with the two histograms, written as $h_m(x_t)$. The method of integral histogram [34] is emploied to calculate the histogram of a rectangle efficiently.

2) COMPOSITION OF THE COMPLEX CELLS

A complex cell is obtained by combining cells with different ways. Similar to CCT tracker [2], two basic operators are applied for building the complex cells. One is the merging which holds the histogram sum of selected cells. The other is the contrast which computes the histogram difference between a chosen cell pair. To enhance the global distinctiveness, we add a novel complex cell into the group of object representation. Finally, five kinds of complex cells are proposed to represent the local, global and contextual information. Local Complex Cell (LCC) is constructed by a single inner cell directly, and its descriptor is just the L_2-norm normalized cell descriptor; Block Complex Cell (BCC) takes neighbouring 2×2 cells to represent larger region of the object, and its descriptor is the merge of the cells; Non-local Complex Cell (NCC) is composed of a randomly selected inner cell pair, and its descriptor is the contrast of the cell pair; Background-Contrast Complex Cell (CCC) is composed of a neighbouring inner-outer cell pair, and its descriptor is the contrast of the two cells. The major challenge for object tracking is to account for drastic appearance variations. Zhong *et al.* [27] propose a robust appearance model that exploits both holistic templates and local representations. Inspired by this strategy, we create a new complex cell to enhance the global distinctiveness, named Global Complex cell (GCC) (as shown in Fig. 4). It is a special case of BCC when the block is merged as an object bounding box ($w \times h$, where $w > 2$ and $h > 2$ are the numbers of block's width and height). Even though the GCC is simple, but it denotes to better control the global information of object's appearance.

3) TEMPLATE MATCHING

1) *Template*: The foreground and background information are represented as a two-layer template separately. The foreground template O^{ta} is in accord with inner cells, however, the inner and outer cells are in line with the background template O^{bg}, which are occupied by the background. We roughly estimate the object continuous variations under Gaussian model with mean μ and variance D during tracking. The object template O can be represented as:

$$O^{ta} = \{\mu_m^{ta}, D_m^{ta} | m \in M_{in}\} \tag{1}$$

$$O^{bg} = \{\mu_m^{bg}, D_m^{bg} | m \in M_{out}\} \tag{2}$$

where $O^{all} = O^{ta} \bigcup O^{bg}$. We use μ as the cell descriptors for template, and take the inner cells from target template and the outer cells from background template to construct the complex cells. The complex cell is denoted as C_T.

2) *Measure the Similarity*: In this algorithm, tracking aims at searching for the state that is most similar to the template. A score function is proposed to measure the similarity, which is voted by likelihoods of all the complex cells.

$$S(x_t) = \sum_{i \in I} \alpha^i \sum_{j \in J^i} \omega_j m_j(C(x_t), C_T) \tag{3}$$

where I denotes complex cell types, written as $\{L, B, N, C, G\}$, and J^i are the complex cell indexes for a specific type i. $C(x_t)$ and C_T are complex cell descriptors for x_t and template T respectively. The optimal state $\hat{x}_t$ is the one with maximal score, namely $\hat{x}_t = \operatorname{argmax}_{x_t} S(x_t)$. $m_j(C(x_t), C_T)$ is the likelihood of j_{th} complex cell. To measure the likelihood, we introduce a kernel function y. Suppose d and f are the corresponding complex cell descriptors, function y integrates the two channel features by a linear combination:

$$y(d, f) = (d^I + f^I) + (d^G + f^G) \tag{4}$$

There are two types of weights are considered, namely, adaptive weight ω_j and fusion weight α_i. The ω_j is associated with each complex cell and is determined by

$$\omega_j = \frac{s_j \cdot o_j}{\sum_{j \in J^m} s_j \cdot o_j} \tag{5}$$

where s_j, o_j are the stability and occlusion factor corresponding to complex cell j.

$$s_j = \begin{cases} \log(\frac{Z_{in}}{Tr(D_m^{ta})}) & m \in M_{in} \\ \log(\frac{Z_{out}}{Tr(D_m^{bg})}) & m \in M_{out} \end{cases} \tag{6}$$

where the $Z_{in} = \sum_{M_{in}} Tr(D_m^{ta})$, and the $Z_{out} = \sum_{M_{out}} Tr(D_m^{bg})$.

$$o_j = \begin{cases} 1 \to 0, & \text{if } (\exists o_j = 0) \wedge (r(l, j) > \theta_{occ}) \\ 0 \to 1, & \text{if } y(h_m(\hat{x}_t), \mu_m^{ta}) > \theta_{deocc} \end{cases} \tag{7}$$

where $(h_m(\hat{x}_t)$ is the current cell descriptor extracted from the optimal state, and $(r(l, j) = \frac{y(h_m(\hat{x}_t), \mu_m^{bg})}{y(h_m(\hat{x}_t), \mu_m^{ta})}$ is the ratio of its affinities with the neighbouring background template and the affinity with the its target template.

The α_i is associated with each complex cell type and balanced between different complex cell types to preserve global distinctiveness. For i type complex cells, α^i is computed based on the samples in the previous frame $t-1$,

$$\alpha \propto \frac{S^m(\hat{x}_{t-1}) - median}{MAD} \tag{8}$$

where $S(x_t) = \sum_{j \in J^i} \omega_j m_j(C(x_t), C_T)$ is the score for corresponding complex cells. *median* is the median of S^m and MAD measures their deviation defined as $\text{median}(|S^m(x_{t-1}^k) - median|)$.

C. MODEL UPDATE

As we know, many algorithms (like CCT [2] and CT [23]) use fixed learning rate parameters to update their template models. The target model is particularly sensitive to noise. Hence, we focus on the situation that the learning rate is changing with the corresponding object's appearance variation. In this way, noise can be avoided effectively.

1) CELLS MATCHING

In order to pursue the mentioned ideal situation, we propose a novel strategy by matching the inner cells between current object with the template's individualistically. Each inner cell is belong to one part of the target. The success of each pair of cells match depends on the similarity between them. In other words, there is evidence that proves the phenomenon which the final matching result demonstrates the degree of the appearance variation of the target. In order to rule out the keypoints of the non matching relation, which is produced because of the occlusion and background confusion, the match method proposed by Lowe *et al.* [1] is employed. Measurement ϑ of the appearance variation is

$$\vartheta = \frac{M_{num}}{K_{num} + c} \tag{9}$$

where c is a constant. M_{num} and K_{num} are the number of matched keypoints and total number of keypoints in current frame, respectively.

2) LEARNING RATE MEASURING

Once the appearance change factor are determined, the corresponding learning parameters can be calculated. In such a principle, when the target is covered or the deformation is very intense, learning rate for the current target should be reduced. Conversely, learning rate for the previous target information should be increased. This can prevent the negtive impact of noise on the object model. In order to satisfy this characteristic, a double sigmoid function [28] is employed:

$$\varphi = \frac{1}{1 + exp(-2(\frac{\vartheta - t}{r}))} \tag{10}$$

$$r = \begin{cases} r_1 & \vartheta \leq t \\ r_2 & \text{other,} \end{cases} \tag{11}$$

The less the value indicates that the degree of appearance variation between the current target and the template is greater. In our experiment, $t = 0.5$, $r_1 = 0.2$ and $r_2 = 0.3$.

3) MODEL UPDATING

As cells descriptors in O^{ta} and O^{bg} are modeled as Gaussian distributions, we incrementally update the parameters $(\mu_m^{ta}; D_m^{ta})$ and $(\mu_m^{bg}; D_m^{bg})$ by current cell descriptor $\hat{h}_m(xt)$ that is also modeled as a Gaussian distribution $G(\hat{h}_m; D_0)$. The template updating is therefore operated as Gaussian merging. We first update the target template O^{ta}:

$$\tilde{\mu}_m = \varphi^{ta}\mu_m^{ta} + (1 - \varphi^{ta})\hat{h}_m \tag{12}$$

$$\tilde{D}_m = \varphi^{ta}(D_m^{ta} + \mu_m^{ta}\mu_m^{ta\mathrm{T}}) + (1 - \varphi^{ta})(D_0 + \hat{h}_m\hat{h}_m^{\mathrm{T}}) - \tilde{\mu}_m\tilde{\mu}_m^{\mathrm{T}} \tag{13}$$

$$\mu_m^{ta} = o_l\tilde{\mu}_m + (1 - o_l)\mu_m^{ta} \quad D_m^{ta} = o_l\tilde{D}_m + (1 - o_l)D_m^{ta} \tag{14}$$

where φ^{ta} is a learning rate parameter. When updating the object template O_{ta}, the $\varphi_{ta} = \varphi$ in the Eq. 10. Parameter update model can automatically adjust the learning rate parameter for updating the object model according to the change of target appearance. Thus, it is more adaptive, and the robustness of the algorithm is enhanced. The update rule for background template O^{bg} is similar to O^{ta}. However, there are two main differences. One is that we only update the $(\mu_m^{bg}; D_m^{bg})$ for the cell labelled as 0. The other is that we update with a fix learning rate φ_{bg} since it is difficult to find out the influence of background appearance variation on tracking.

IV. EXPERIMENTS

Our method is implemented in native Matlab/C++. The experiments are performed on an Intel i7 Quad-Core machine with 2.34 GHz CPU and 8 GB RAM. In this section, we compare our RAFT with 12 current state-of-the-art trackers on challenging sequences. The 12 trackers are as follows: incremental learning tracker (IVT) [12], decomposition tracker (VTD) [14], structured output tracker (Struck) [24], sampling trackers (VTS) [21], context tracker (CXT) [22], compressive tracker (CT) [23], sparsity-based collaborative model tracker (SCM) [27], complex cell tracker (CCT) [2], consensus-based matching-and-tracking tracker (CMT) [4], convolutional networks tracking (CNT) [26], multiple experts tracker (MEEM) [25] and MUlti-Store Tracker (MUSTer) [7]. These algorithms are related to keypoints-based or parts-based model under tracking-by-matching and achieve promising performance.

A. DATASETS

In order to verify the experiment, we employ the tracking benchmark dataset [35] which includes 50 fully-annotated videos with real image sequences (more than 29,000 frames). In addition, in order to ensure the applicability of the experiment, an animated video (bird2) is also adopted from [36]. For better evaluation and analysis of the strength and weakness of the tracking algorithms, the sequences are categorized according to 11 attributes, including illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background clutters (BC), and low resolution (LR), which are summarized in Table 1.

B. SETUP

Given a target location at the current frame, the radiuses for coarse-to-fine search are set as 20 and 4. The former takes

TABLE 1. Test videos categorized with 11 attributes.

Sequence	IV	OPR	SV	OCC	DEF	MB	FM	IPR	OV	BC	LR	Total
Basketball	√	√		√	√					√		5
Bolt		√		√	√			√				4
Boy		√	√			√	√	√				5
Car4	√		√									2
CarDark	√									√		2
CarScale		√	√	√			√	√				5
Coke	√	√		√			√	√				5
Couple		√	√		√		√			√		5
Crossing			√		√					√		3
David	√	√	√	√	√	√		√				7
David2		√						√				2
David3		√		√	√					√		4
Deer						√	√	√		√	√	5
Dog1		√	√					√				3
Doll	√	√	√	√				√				5
Dudek		√	√	√	√		√	√	√	√		8
FaceOcc1				√								1
FaceOcc2	√	√		√				√				4
Fish	√											1
FleetFace		√	√		√	√	√	√				6
Football		√		√				√		√		4
Football1		√						√		√		3
Freeman1		√	√					√				3
Freeman3		√	√					√				3
Freeman4		√	√	√				√				3
Girl		√	√	√				√				4
Ironman	√	√	√	√		√	√	√	√	√	√	10
Jogging-1		√		√	√							3
Jogging-2		√		√	√							3
Jumping						√	√					2
Lemming	√	√	√	√			√		√			6
Liquor	√	√	√	√		√	√		√	√		8
Matrix	√	√	√	√			√	√		√		7
MotorRolling	√		√			√	√	√		√	√	7
MountainBike		√						√		√		3
Shaking	√	√	√					√		√		5
Singer1	√		√	√	√							4
Singer2	√	√			√			√		√		5
Skating1	√	√	√	√	√					√		6
Skiing	√	√	√		√			√				5
Soccer	√	√	√	√		√	√	√		√		8
Subway				√	√					√		3
Suv				√				√	√			3
Sylvester	√	√						√				3
Tiger1	√	√		√	√	√	√	√				7
Tiger2	√	√		√	√	√	√	√				7
Walking			√	√	√							3
Walking2			√	√							√	3
Woman	√	√	√	√	√	√	√					7
Bird2		√	√	√	√		√	√				6
Total	25	28	30	20	12	18	32	40	6	21	4	

100 samples and the latter takes 50 samples. The number of inner cells is around 25 and the outer cells is corresponding to the bounding box. The number for different types of complex cells are set as follows: LCC equals to inner cells; BCC covers every possible 2 × 2 cell region; 60 inner cell pairs are randomly chosen for NCCs; 30 inner-outer cell pairs are chosen as CCCs across the bounding box boundary. The learning rate parameter for background model is set as 0.85. It is important to note that the parameters in our method are fixed through the experiments.

C. EVALUATION METRIC

Two metrics are used to evaluate the proposed algorithm with 12 state-of-the-art trackers. The first metric is the success plots based on the overlap metric [35], which is defined as, $score = \frac{area(ROI_T \bigcap ROI_G)}{area(ROI_T \bigcup ROI_G)}$, where ROI_T is the tracking bounding box and ROI_G is the ground truth. $\bigcap$ and $\bigcup$ represent the intersection and union of two regions. We define the successful frames as the frames whose overlap is larger than a given threshold of t_0. Another evaluation metric is the precision plots based on the location error metric. Center location error is a widely used evaluation metric for tracking precision. It is defined as the average Euclidean distance between the center locations of the tracked targets and the manually labeled ground truths. The precision plot shows the percentage of frames whose estimated location is within the given threshold distance of the ground truth. For fair evaluation, the Area Under Curve (AUC) of each success plot is preferred to measure the success ratio. The One-Pass Evaluation (OPE) which is basis of the average precision and

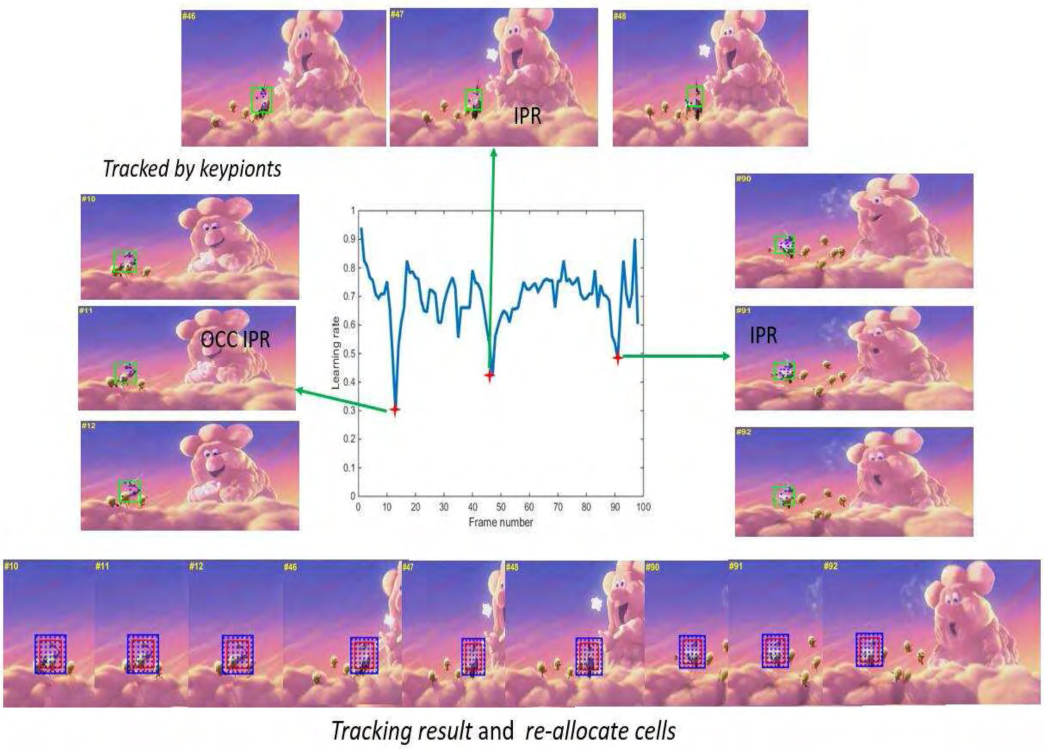


FIGURE 5. The relationship between learning rate parameters and the appearance variations of image. The green bounding box is the result of the initial tracking based on keypoints and points inside are the tracked by LK. The bottom row results are the final tracking results and the cells are re-allocated corresponding to the keypoints are determined and centered in each grid.

the success rate given the ground truth of the first frame is applying for evaluating the robustness of our algorithm.

D. EXPERIMENTAL ANALYSIS

1) EFFECTIVENESS OF OCCLUSION AND IN-PLANE ROTATION HANDLING

When the object's appearance changes drastically such as occlusion or in-plane rotation, the learning rate will be reduced for the model update. As a result, it can maintain the precious information while avoiding the noises. As shown in Fig. 5, the appearance variation of the bird is very dramatic. The frame #10, #11, #12, #46, #47, #48, #90, #91, #92 exist obvious changes of the object corresponding to the curve, the learning rates are descending. The bottom row results prove the validity of the measurement of the appearance variations.

2) FAILURE DETECTION FOR KEYPOINTS-BASED TRACKING

Median-Flow [32] tracking algorithm requires a prerequisite, that is, the visibility of objects. Therefore, it will inevitably fail if the object undergoes heavy occlussion or moves outside the view of the camera. In order to distinguish these conditions, we develop the following mechanism. Let d_p instruct the displacement of a single point of the Median-Flow tracker and d_m be the median one. $|d_p - d_m|$ is defined as the remain of a single displacement. If the median $|d_p - d_m| > 10$ pixels, the tracker is declared as failure. This mechanism is able to reliably identify failures caused by fast motion, out of view or fast occlusion of the target. If the failure is detected, the tracker does not return any bounding box and does not provide the candidate center for precise tracking. The algorithm directly adopts the object center located in the previous frame for sampling and tracking. In addition,

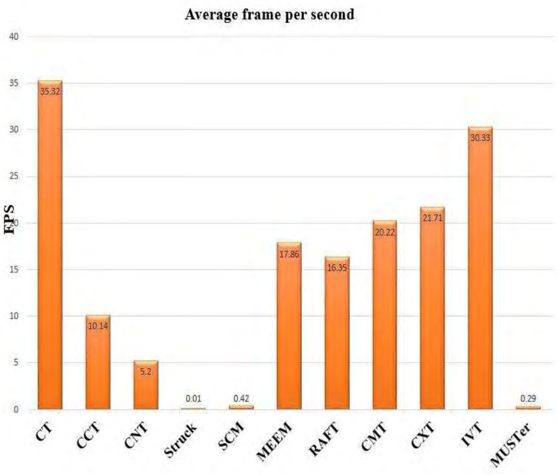
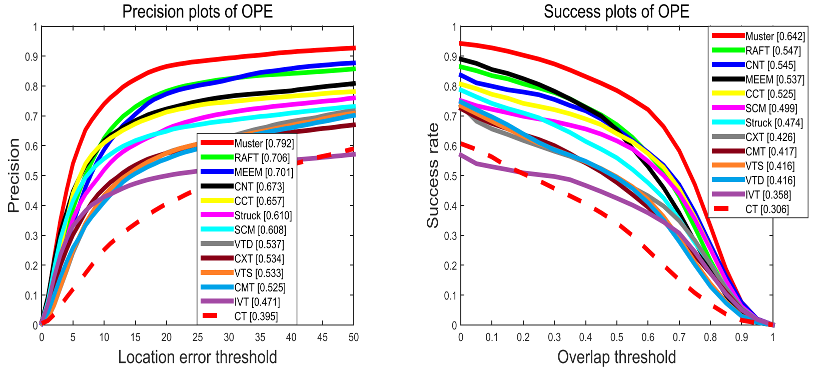


FIGURE 6. Comparison of the algorithms' efficiency. The integer number at the vertical coordinates indicates the speed of the tracking algorithm (frame per second). The horizontal coordinate notes the tracking algorithm.



(a) (b)

FIGURE 7. Precision and success plots over all 51 sequences. The performance scores for each tracker are reported in the legends. In both cases our tracker (RAFT) performs favorably to state-of-the-art tracking methods. The best viewing is through color display. (a) Precision plots. (b) Success plots.

TABLE 2. Score of success plot (Best viewed on a color display). The red fonts indicate the best performance. The blue fonts indicate the second best ones and the green fonts indicate the third best ones.

Algorithm	IV	OPR	SV	OCC	DEF	IPR	OV	BC	LR	MB	FM	Overall Score
RAFT	0.520	0.547	0.479	0.535	0.569	0.554	0.590	0.556	0.487	0.480	0.457	0.547
MUSTer	0.608	0.615	0.595	0.629	0.650	0.583	0.590	0.617	0.482	0.558	0.552	0.642
MEEM	0.515	0.540	0.471	0.532	0.525	0.511	0.591	0.521	0.195	0.448	0.473	0.537
CCT	0.493	0.528	0.468	0.528	0.561	0.471	0.519	0.483	0.153	0.444	0.437	0.525
SCM	0.473	0.470	0.518	0.487	0.448	0.458	0.361	0.450	0.279	0.298	0.296	0.499
CNT	0.456	0.501	0.508	0.503	0.524	0.495	0.439	0.488	0.437	0.417	0.404	0.545
VTS	0.429	0.425	0.400	0.398	0.368	0.298	0.443	0.428	0.168	0.304	0.300	0.416
Struck	0.428	0.432	0.425	0.413	0.393	0.444	0.459	0.458	0.372	0.433	0.462	0.474
VTD	0.420	0.434	0.405	0.403	0.377	0.430	0.446	0.425	0.177	0.309	0.302	0.416
CMT	0.381	0.415	0.398	0.416	0.411	0.366	0.380	0.371	0.168	0.309	0.357	0.417
CXT	0.368	0.418	0.389	0.372	0.324	0.452	0.427	0.338	0.312	0.369	0.388	0.426
IVT	0.306	0.323	0.344	0.325	0.281	0.330	0.274	0.291	0.238	0.197	0.202	0.358
CT	0.295	0.297	0.302	0.321	0.345	0.282	0.359	0.273	0.120	0.269	0.298	0.306

once the tracker is successful, we also calculate the distance between the tracked points and the object center located in the previous frame. If the distance beyond the area of the bounding box, corresponding keypoint will be removed from T.

3) COARSE-TO-FINE SEARCH

To effectively search for the optimal state $\hat{x}_t$, we propose a coarse-to-fine search strategy based on keypoints-based tracking to gradually approximate the high score region. Once keypoints-based tracking is successful, the complex cells feature are extracted at location of the bounding box and the similarity is computed between current sample and previous template. If the tracking result of keypoints-based tracker is not accepted as the final target location, we need to do the following precise tracking. Otherwise, we achieve the final tracking location directly to speed our algorithm (as shown in the Fig. 6. It is clear that our RAFT tacker is faster than CNT method [26], MUSTer tracking algorithm [7], SCM approach [27], Struck [24] and CCT tracker [2]). Obviously, sampling is essential for precise tracking. There are two stages, namely rough exploration and accurate positioning. For the former, we adopt the location of the keypoints-based tracker as the center, taking 100 samples in the region of 20 pixels as the radius. A state $\hat{x}_t$ with maximal score is

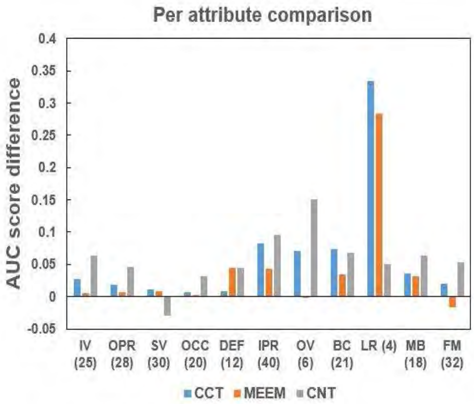
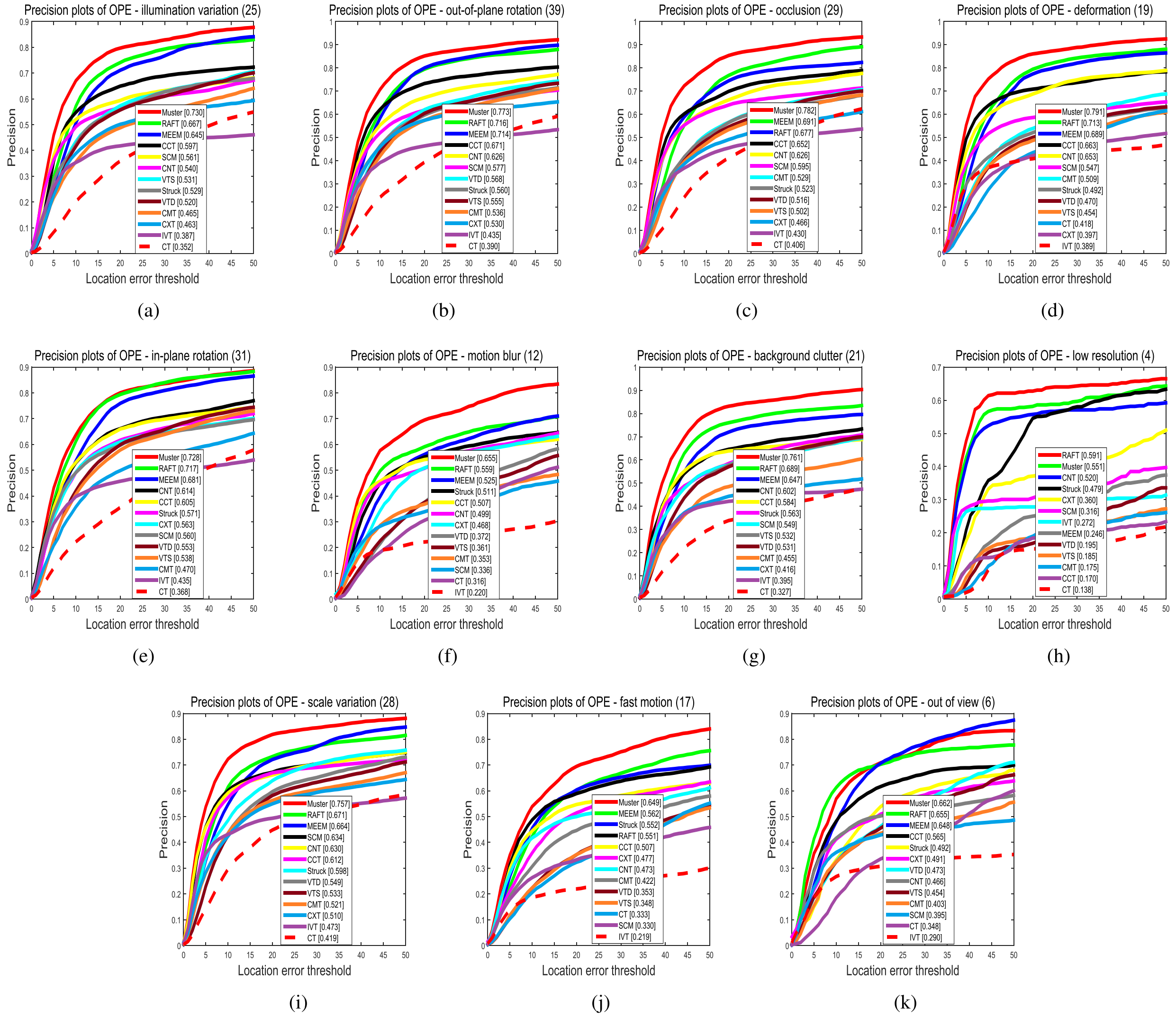


FIGURE 8. Per attribute comparison on AUC score of the proposed RAFT with CCT [2], MEEM [25] and CNT [26]. The bars are the performance difference between RAFT with CCT [2], MEEM [25] and CNT [26](Best viewed on a color display as shown in the chart). Positive means RAFT is better. The integer number at horizontal coordinate is the number of tracking sequences belonging to that group.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k)

FIGURE 9. Precision plots of videos with different attributes namely: IV, OPR, OCC, DEF, IPR, MB, BC, LR, SV, FM and OV (best-viewed on high-resolution display). (a) IV. (b) OPR. (c) OCC. (d) DEF. (e) IPR. (f) MB. (g) BC. (h) LR. (i) SV. (j) FM. (k) OV.

utilized as the new center for precise tracking. As to the former, 50 samples are taken at the new center in the region with a radius of 4 pixels. In this way, we can get a more accurate tracking results.

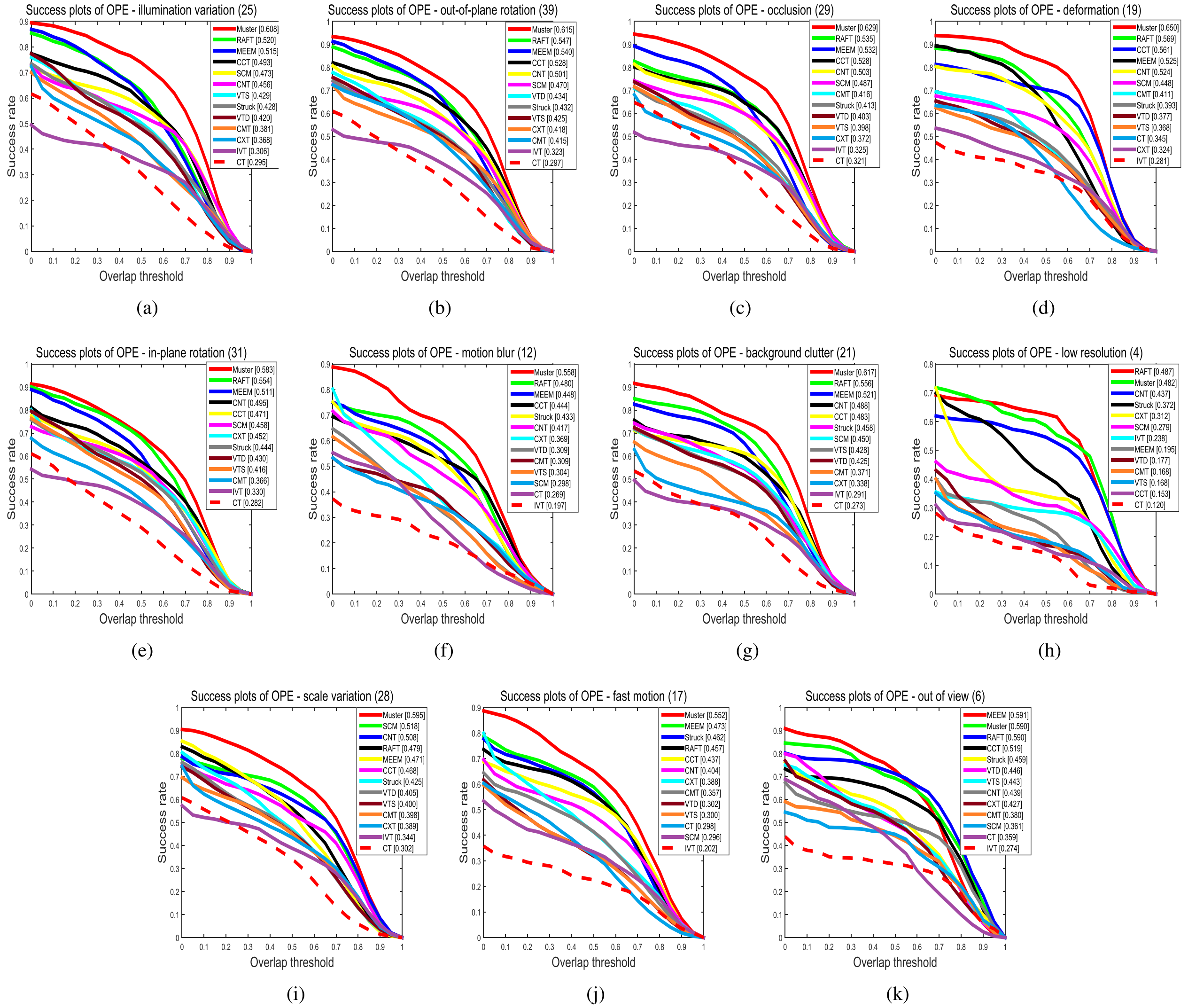
4) ANALYSIS OF THE CELLS MATCHING

The keypoint is extracted from the center of each inner cell. Let nature take its course, and the keypoint is described by the feature (G and I) of corresponding inner cell. In order to eliminate the interference, we match the key points for two times respectively based on the intensity (written as P_I) and the gradient (P_G). We fuse P_I and P_G into a set P_{IG}, where $P_{IG} = P_I \cup P_G$, discarding all matched keypoints based on intensity when there exists a matched keypoint associated with gradient. Intuitively, matched keypoints are more robust as they do not match on a 16 dimensional vector obtained by concatenating the two histograms (G and I).

E. EXPERIMENTAL EVALUATION

1) QUANTITATIVE COMPARISONS

- **Overall performance:** Fig. 7 illustrates overall performance of the 12 categories of evaluated tracking algorithms in terms of precision plot and success plot over the challenging sequences. Note that all the plots are generated using the code library from the benchmark evaluation [35], and the code of CMT [4], MEEM [25] CNT [26] and MUSTer [7] methods are provided by the authors. The corresponding ranked results are shown in the legends of each drawing. The proposed RAFT ranks 2nd in terms of both precision plot and success plot. In the precision plot, the precision score of RASF is 0.706, which is close to the MUSTer (0.792) methods, but outperforms MEEM [25] (0.701), CNT [26](0.673) and CCT [2] (0.657). Meanwhile, in the success plot, the proposed RAFT achieves the AUC of 0.547, which also outperforms CNT [26](0.545), MEEM [25] (0.537),



(a) (b) (c) (d)

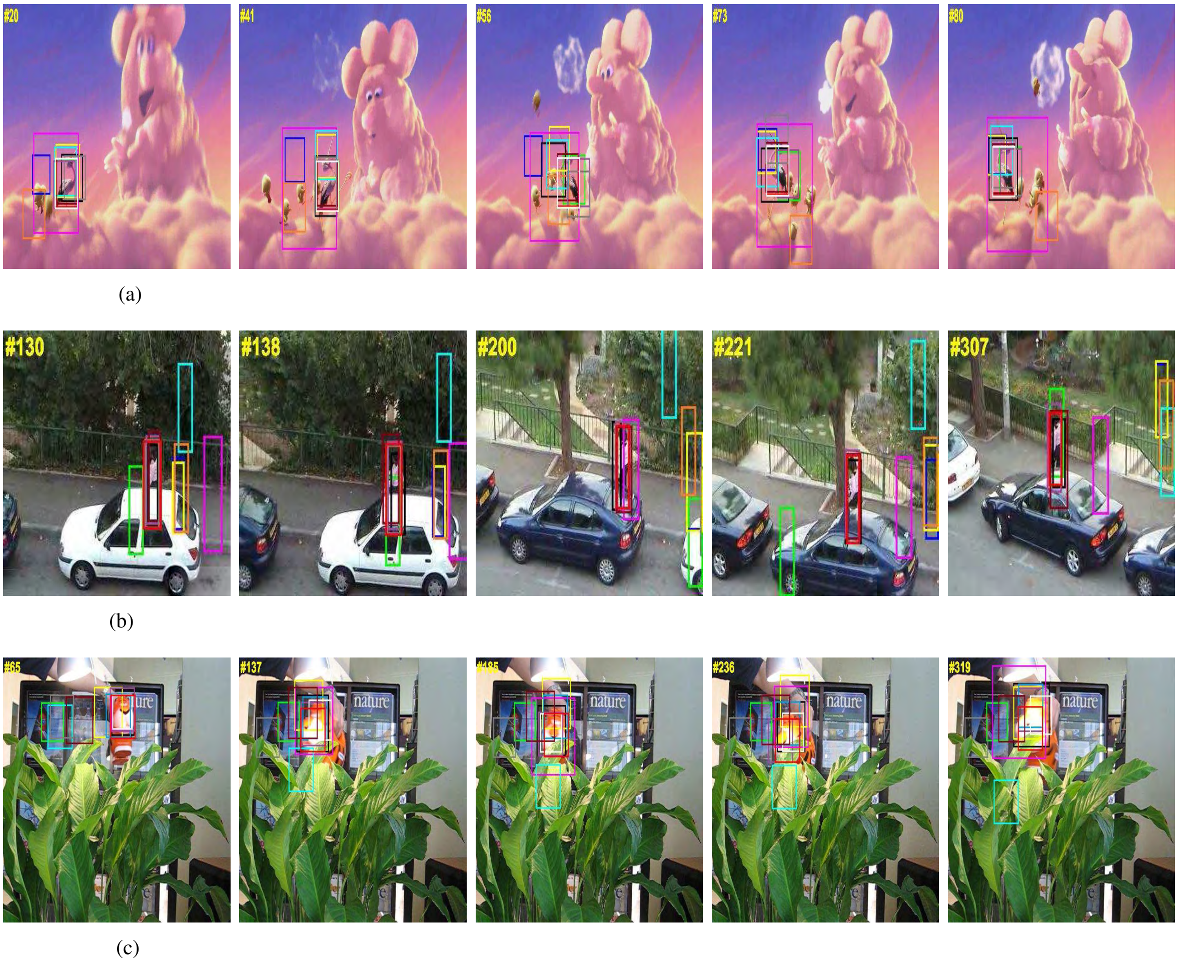
(e) (f) (g) (h)

(i) (j) (k)

FIGURE 10. **Precision plots of videos with different attributes namely: IV, OPR, OCC, DEF, IPR, MB, BC, LR, SV, FM and OV (best-viewed on high-resolution display). (a) IV. (b) OPR. (c) OCC. (d) DEF. (e) IPR. (f) MB. (g) BC. (h) LR. (i) SV. (j) FM. (k) OV.**

and CCT [2] (0.525). As described earlier, optical flow is used in our approach first to filter out motion inconsistent candidates. Specifically, given the pixels covered by the predicted box in the previous frame and the estimated optical flow, we know where those pixels are in the current frame. It narrows the scope for searching the target and we throw more particles in this region. As a result, not only improve our tracker's accuracy, but also more efficient. Even though the proposed RAFT is less than MUSTer [7](0.642), it still achieves promising results compared to other state-of-the-art methods in both success plot and precision plot. The promising scores at mild thresholds indicates our tracker hardly misses targets while the competitive scores at strict thresholds presents that our approach also searches tight bounding boxes for targets.

- **Attribute-based performance:** Several factors can affect the performance of a visual tracker. To analyze the strength and weakness of the proposed algorithm, we further test the tracker on challenging sequences with 11 attributes. The results are as follows: Fig. 9 shows an example of precision plots of different attributes. Fig. 10 shows an example of success plots of different attributes. We note that the proposed RAFT tracker ranks within top 3 on 9 out of 11 attributes in success plots, and outperforms the CCT method on all 11 attributes. In the precision plots, the RAFT algorithm ranks top 3 on 9 out of 11 attributes, and outperforms the CCT method on all attributes. Since the AUC score of the success plot is more informative than the score at one position in the precision plot, in the following we analyze the results based on these values. Table 2 summarizes the tracking results in terms of success plots. Despite the MUSTer [7] tracker achieves the best results, RAFT still reaches state-of-the-art performance, being tantalizingly close to MUSTer.



(a)

(b)

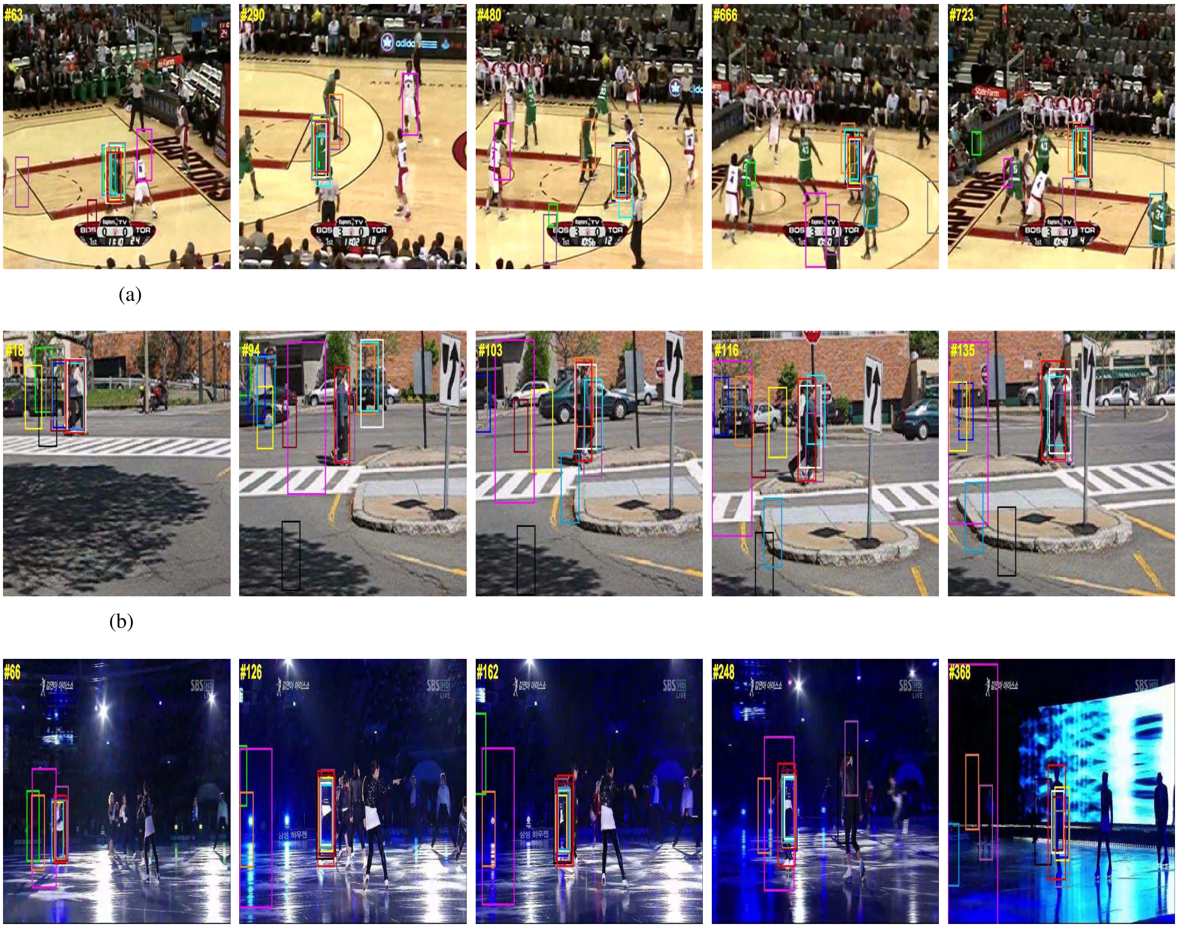
(c)

FIGURE 11. Qualitative comparison results of our approach with 12 state-of-the-art trackers in challenging situations. The example frames are the bird2, woman and tiger2 sequences respectively, in which the targets undergo occlusion and rotations. Best viewed on color display. The results of IVT [12], VTD [14], CCT [2], VTS [21], CXT [22], CMT [4], CT [23], Struck [24], MEEM [7], MUSTer [7], CNT [26], SCM [27] and our RAFT are represented by green, blue, black, yellow, carmine, ultramarine, orange, purple, turquoise, white, crimson, grey and red boxes respectively. (a) bird2. (b). woman. (c) tiger2.

To gain more insights, we evaluate the performance of RAFT for individual attributes and compare with CCT [2], MEEM [25] and CNT [26]. Fig. 8 shows the plot for AUC. It is observed that RAFT is better mainly in LR, MB, OCC, DEF and IPR. Especially on the image sequences with the low resolution attribute, the RAFT algorithm ranks first among all evaluated trackers. The low resolution in the sequences makes it hard to extract effective features from the image to represent the target. In contrast, not only local representations and the context information with the surronding region are exploited, but also holistic templates information is extracted in our algorithm. Such a fully collaborative model makes it robust to separate the target from the background. For the videos with attributes such as in-plane rotation, out-of-plane rotation, deformation, motion blur and occlusion, the RAFT algorithm ranks first (except MUSTer) among all evaluated algorithms, such as MEEM, CNT, and CCT. All these methods use local image features as image representations. The MEEM method utilizes HOG features to describe the target and proposes multi-expert restoration scheme to avoid drifting. Furthermore, both CNT and CCT algorithms employ local features extracted from the normalized local image patches. The CNT algorithm exploits useful local features across the target object via filtering while the CCT method extract the information between the cells which are located in the target and background with color histogram. The proposed RAFT algorithm draw on these advantages and improve the update strategy to handle the drift problem.

2) QUALITATIVE COMPARISONS

- **Overall:** Fig. 2, Fig. 11, Fig. 12 and Fig. 13 illustrate some examples of qualitative tracking results over the challenging sequences. Our tracker can successfully track the object, since our observation models combine the keypints tracking with the template matching and evolve themselves by online updating. Our tracker efficiently overcomes the occlusion, rotation, scale viriation, motion blur and low resolutio. Additionally, the



(a)

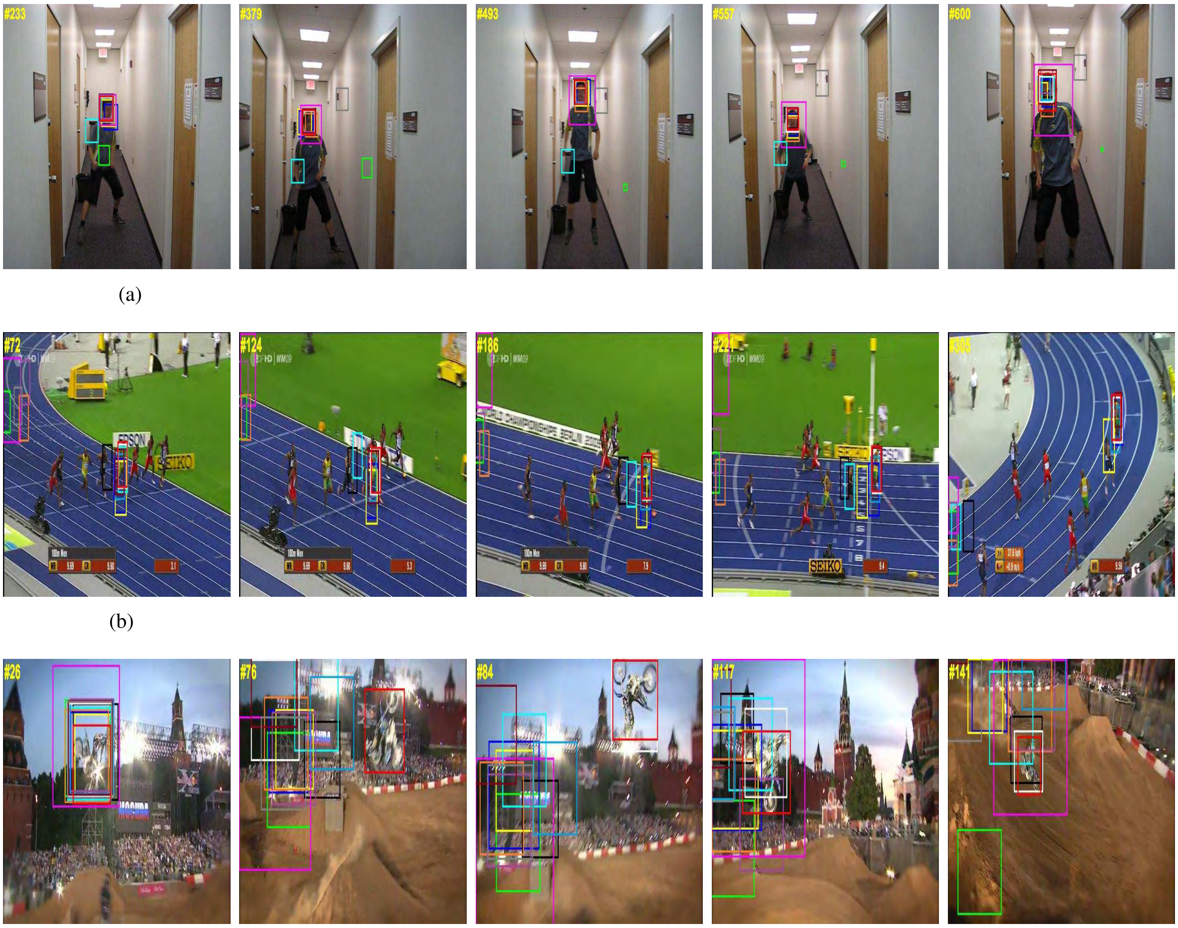
(b)

(c)

FIGURE 12. Qualitative comparison results of our approach with 12 state-of-the-art trackers in challenging situations. The example frames are the basketball, couple and skating1 sequences respectively, in which the targets undergo deformation and complex background. Best viewed on color display. The results of IVT [12], VTD [14], CCT [2], VTS [21], CXT [22], CMT [4], CT [23], Struck [24], MEEM [7], MUSTer [7], CNT [26], SCM [27] and our RAFT are represented by green, blue, black, yellow, carmine, ultramarine, orange, purple, turquoise, white, crimson, grey and red boxes respectively.(a) basketball. (b) couple. (c) skating1.

proposed algorithm is also robust to deformation and background clutter because the observation models utilize a two-layer templates to represent the object and background respectively. It is conductive to enhance the robustness and distinctiveness of the template. Specific analysis is as follows.

- **Occlusion and Rotation:** Fig.11 shows sampled results of three sequences where the objects undergo heavy occlusions and rotations. Occlusion is one of the most important problems in object tracking. In fact, several trackers including the SCM method [27], the CNT algorithm [26], the CCT method [2], MEEM method [7], MUSTer approach [7] and our tracker are developed to solve this problem. In contrast, the IVT tracking method [12], the CMT tracking method [4] and the VTD tracking system [14] are less effective in handling occlusions as shown in Fig. 11. In the woman sequence, a woman is almost occluded by the car (e.g., #130, #138, #200, #221 and #307). Only the MUSTer, CXT, CNT, MEEM and our RAFT algorithm are able to track the obscured object. In the tiger2 sequence, the target is frequently occluded by dense leaf (e.g., #185, #236 and #319). In addition, there are several illumination changes in this video. The CCT, CXT, CNT, CT, IVT, VTS and MEEM methods do not perform well. In our coarsely searching process, we estimate the possible occluded patches and develop the keypoints by the optical flow which only finds the region that are not occluded. Thus, the occlusion handling scheme based on the inner cells' matching effectively alleviates the pernicious influence of occlusions. Model drift occurs because factors like tracking failure, occlusions and misalignment of training samples can lead to bad model updates. Aside from tracking a obscured target object, our tracker updates the appearance model online. The updating is deponding on the results of cells' matching and controlling the learn rate parameter to aviod bringing noises especially when heavy occlusions occur.



(a)

(b)

(c)

FIGURE 13. **Qualitative comparison results of our approach with 12 state-of-the-art trackers in challenging situations. The example frames are the boy, bolt and motorrolling sequences respectively, in which the targets undergo motion blur and low resolution. Best viewed on color display. The results of IVT [12], VTD [14], CCT [2], VTS [21], CXT [22], CMT [4], CT [23], Struck [24], MEEM [7], MUSTer [7], CNT [26], SCM [27] and our RAFT are represented by green, blue, black, yellow, carmine, ultramarine, orange, purple, turquoise, white, crimson, grey and red boxes respectively. (a) boy. (b) bolt. (c) motorrolling.**

In addition, our tracker is able to deal with rotation when the target is occluded at bird2 (e.g., #41, #56 and #73) and tiger2 (e.g., #65, #137 and #80) video. Bird2 is a virtual animation video, the bird's posture change is very exaggerated. Owing to the appearance model we employ, our tracker can accurately locate the target object as our generated histogram takes not only the spatial information of local patches, but also the context information between the surronding region with the traget into consideration.

- **Deformation and Complex background:** Fig.12 shows some examples of the tracking results of three challenging sequences where the objects undergo large shape deformation and complex background. In the basketball sequence, the hoopster appears in the sequences with rapid appearance variations due to shape deformation and fast motion. Only the MUSTer, MEEM and RAFT algorithms can track the targets well. The SCM, CXT and VTS methods miss the target and drift at the beginning of the sequence (#63). The CT approach drifts to the background at frame 290. Meanwhile, the the background is cluttered and the target moves fast. In frame #666, the CNT tracker fails as jumping from the target to another as it only looks for the maximum similarity with the target in the first frame. In the couple sequence, the woman brings appearance variations due to non-rigid body deformation when walking and the color of cars in the background is very similar with the woman's cloth. Most trackers undergo large drift. All these methods use local features that are not robust to deformation. The IVT, VTS, CCT, CT and CMT lose the goal in frame #18. Even the MUSTer, MEEM and CNT are confused by the cars behind the woman and regard it as the traget in frame #94. While they all re-dectect the traget in frame #103, #116 and #135. The MUSTer profits from the short-and long-term memory stores to process target appearance changes. The MEEM propose a multi-expert restoration scheme to address the model drift problem in online tracking. The CNT algorithm utilizes the target template from the first frame to handle the drift problem. The target object in the skating sequence undergoes heavy appearance variations when skating. Furthermore, the sequence also undergos the complex background and illumination changes. Those make the tracking more difficult. The IVT, CT and Struck methods fail to track of the target object from frame #66. The CXT method locks on other skater which is very similar with the target(#248).

- **Motion blur and Low resolution:** Fig.13 shows some examples of the tracking results of three challenging sequences where the objects undergo large fast motion, motion blur and low resolution. Fast motion of the target object or the camera leads to blurred image appearance which make tracking difficult. The boy sequence presents the tracking results in which the appearance of the boy is almost indistinguishable due to the motion blur. The IVT and CMT algorithms fail to follow the target right as shown in frame #233, #379, #493 and #557. The reason is that the true target is blurred and it is difficult for the detector of IVT to distinguish it from the background. The CMT only relies on the keypoints matching while it can not extract effective keypoints in the blur image. In the bolt sequence, the object appears in the scenes with blur due to fast motion. Only the MUSTer, CNT and our RAFT algorithms can track the targets well. The IVT, Struck, SCM, VTD, CT, CMT, CXT, MEEM and CCT methods undergo large drift at the beginning of the frame (e.g., #72, #124). The VTD and VTS methods drift to the background at frame #222. The motorrolling sequence undergo low resulution, illumination change as well as scale variations when moving from the peak to the valley of the pathway. Most tracking algorithms fail to track the obect in this sequence. Even the MUSTer approach loses the object in the frame #76 and re-dectects and track the object as shown in frame #84, #117 and #141. The proposed RAFT algorithm well handles the situation as the optical flow narrows the searching region and the tracker exploits both holistic templates and local representations to better separate the target from the background. Meanwhile, by updating the negative and positive templates online, the proposed algorithm successfully tracks the target object throughout the sequence.

V. CONCLUSION

In this paper, we have proposed a robust adaptive fusion tracking algorithm based on complex cells and keypoints to balance keypoints, local descriptors and global representations. The coarse and precise tracking have played an important role, respectively, based on the keypoints and complex cells in our framework. Finally, measurement of appearance variation has been measured by matching the current inner cells with template's individualistically. In the bases of the measurement, an adaptive learning rate parameter has been estimated for updating the object appearance model while avoiding noises. We have demonstrated in an extensive evaluation that our approach achieves prossing results on a large number of challenging sequences both in quantitative comparisons and qualitative comparisons. Compared to other approaches, our approach was better conductive to handle appearance variations and recover from drifts. Our fusion tracking framework is flexible so that it will be easy to transplant for handling with other challenge such as camera motion in the future.

REFERENCES

[1] M. E. Munich, P. Pirjanian, E. D. Bernardo, L. Goncalves, N. Karlsson, and D. Lowe, "SIFT-ing through features with ViPR," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 72–77, Sep. 2006.

[2] D. Chen, Z. Yuan, Y. Wu, G. Zhang, and N. Zheng, "Constructing adaptive complex cells for robust visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 2013, pp. 1113–1120.

[3] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. Van Den Hengel, "Part-based visual tracking with online latent structural learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2013, pp. 2363–2370.

[4] G. Nebehay and R. Pflugfelder, "Consensus-based matching and tracking of keypoints for object tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jun. 2014, pp. 862–869.

[5] X. Zhou, Y. Li, B. He, and T. Bai, "GM-PHD-based multi-target visual tracking using entropy distribution and game theory," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1064–1076, Apr. 2014.

[6] X. Zhou, H. Yu, H. Liu, and Y. Li, "Tracking multiple video targets with an improved GM-PHD tracker," *Sensors*, vol. 15, no. 12, pp. 30240–30260, 2015.

[7] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2015, pp. 749–758.

[8] S. Gu, Z. Ma, M. Xie, and Z. Chen, "Online learning of mixture experts for real-time tracking," *IET Comput. Vis.*, vol. 10, no. 6, pp. 585–592, 2016.

[9] S. Li, Z. Qin, and H. Song, "A temporal-spatial method for group detection, locating and tracking," *IEEE Access*, vol. 4, pp. 4484–4494, Sep. 2016.

[10] L. Zhao, X. Li, J. Xiao, F. Wu, and Y. Zhuang. (Dec. 2014). "Metric learning driven multi-task structured output optimization for robust keypoint tracking." [Online]. Available: https://arxiv.org/abs/1412.1574

[11] S. Hare, A. Saffari, and P. H. Torr, "Efficient online structured output learning for keypoint-based object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Apr. 2012, pp. 1894–1901.

[12] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008.

[13] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2006, pp. 798–805.

[14] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2010, pp. 1269–1276.

[15] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1064–1072, Sep. 2004.

[16] S. Chan, X. Zhou, and S. Chen, "Online learning for classification and object tracking with superpixel," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Sep. 2015, pp. 1758–1763.

[17] T. Lee and S. Soatto, "Learning and matching multiscale template descriptors for real-time detection, localization and tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2011, pp. 1457–1464.

[18] X. Mei and H. Ling, "Robust visual tracking using ℓ_1 minimization," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1436–1443.

[19] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, 1981, pp. 285–289.

[20] J. Ning, L. Zhang, D. Zhang, and C. Wu, "Robust mean-shift tracking with corrected background-weighted histogram," *IET Comput. Vis.*, vol. 6, no. 1, pp. 62–69, 2012.

[21] J. Kwon and K. M. Lee, "Tracking by sampling trackers," in *Proc. Int. Conf. Comput. Vis.*, Sep. 2011, pp. 1195–1202.

[22] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Sep. 2011, pp. 1177–1184.

[23] K. Zhang, L. Zhang, and M. H. Yang, "Real-time compressive tracking," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2012, pp. 864–877.

[24] S. Hare, A. Saffari, and P. H. Torr, "Struck: Structured output tracking with kernels," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 2011, pp. 263–270.

[25] J. Zhang, S. Ma, and S. Sclaroff, "Meem: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis.*, Apr. 2014, pp. 188–203.

[26] K. Zhang, Q. Liu, Y. Wu, and M.-H. Yang, "Robust visual tracking via convolutional networks without training," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1779–1792, Apr. 2016.

[27] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2012, pp. 1838–1845.

[28] R. Cappelli, D. Maio, and D. Maltoni, "Combining fingerprint classifiers," in *Proc. Int. Workshop Multiple Classifier Syst.*, Sep. 2000, pp. 351–361.

[29] G. Hua and Y. Wu, "Measurement integration under inconsistency for robust tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2006, pp. 650–657.

[30] S. S. Nejhum, J. Ho, and M.-H. Yang, "Visual tracking with histograms and articulating blocks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Sep. 2008, pp. 1–8.

[31] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.

[32] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," in *Proc. IEEE 20th Int. Conf. Pattern Recognit. (ICPR)*, Sep. 2010, pp. 2756–2759.

[33] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.

[34] T. Liu *et al.*, "Learning to detect a salient object," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 353–367, Feb. 2011.

[35] Y. Wu, J. Lim, and M. H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.

[36] F. Yang, H. Lu, and M.-H. Yang, "Robust superpixel tracking," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1639–1651, Apr. 2014.

SIXIAN CHAN received the bachelor's degree from the Anhui University of Architecture in 2012. He is currently pursuing the Ph.D. degree in computer science and technology from the Zhejiang University of Technology. His research interest covers image processing, machine learning, and video tracking.

XIAOLONG ZHOU (M'15) received the Ph.D. degree in mechanical engineering from the Department of Mechanical and Biomedical Engineering, City University of Hong Kong, Hong Kong, in 2013. From 2015 to 2016, he was a Senior Research Fellow with the School of Computing, University of Portsmouth, Portsmouth, U.K. He joined the Zhejiang University of Technology, Zhejiang, China, in 2014, where he currently serves as an Associate Professor with the College of Computer Science. He has authored over 50 peer-reviewed international journals and conference papers. His research interests include visual tracking, gaze estimation, 3-D reconstruction, and their applications in various fields. He serves as an ACM member. He received the T.J. Tarn Best Paper Award on ROBIO2012 and the ICRA2016 CEB award for Best Reviewers. He has served as a Program Committee Member on ROBIO2015, ICIRA2015, SMC2015, HSI2016, ICIA2016, and ROBIO2016.

SHENGYONG CHEN (M'01–SM'10) received the Ph.D. degree in computer vision from the City University of Hong Kong, Hong Kong, in 2003. He was with the University of Hamburg from 2006 to 2007. He is currently a Professor with the Tianjin University of Technology and the Zhejiang University of Technology, China. He has authored over 100 scientific papers in international journals. His research interests include computer vision, robotics, and image analysis. He is a fellow of IET and a Senior Member of CCF. He received the Fellowship from the Alexander von Humboldt Foundation of Germany. He received the National Outstanding Youth Foundation Award of China in 2013.