Computer Vision and Image Understanding 160 (2017) 57–72

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Online hash tracking with spatio-temporal saliency auxiliary

Jianwu Fang[a,b], Hongke Xu[a], Qi Wang[c,*], Tianjun Wu[d]

[a] School of Electronic and Control Engineering, Chang'an University, Xi'an, China
[b] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China
[c] School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China
[d] College of Science, Chang'an University, Xi'an, China

ARTICLE INFO

Article history:
Received 9 April 2016
Revised 19 October 2016
Accepted 29 March 2017
Available online 5 April 2017

Keywords:
Visual tracking
Spatio-temporal saliency
Online hash-code learning
Minimum barrier distance

ABSTRACT

In this paper, we propose an online hashing tracking method with a further exploitation of spatio-temporal saliency for template sampling. Specifically, spatio-temporal saliency is firstly explored to make the sampled templates contain true object templates as much as possible. Then, different from the previous batch modes for hashing, the hashing function in this work is online learned by new pairs of collected templates received sequentially, in which the relationship between the positive templates and negative templates can be appropriately preserved that is more useful for visual tracking. With the hash coding for templates, the between-frame matching can be efficiently conducted. Besides, this work further builds a positive template pool as a memory buffer for object depiction, in which representative truly positive target templates are gathered and utilized to restrain the degradation of the appearance model due to the error accommodation in online hashing. Extensive experiments demonstrate that our tracker performs favorably against the state-of-the-art ones.

1. Introduction

Visual tracking is a classic and challenging problem in computer vision. Because of the wide range of applications, e.g., human-computer interfaces, intelligent surveillance, video editing and description, etc., lots of researchers devoted their efforts to numerous trackers' designing and progressed the tracking significantly. However, robust tracking is still challenging owing to the frequent disturbance by heavy appearance occlusion, complex background clutter, frequent rotating and scaling, abrupt motion, severe illumination changes, etc. (Li et al., 2016).

To address the above problems, most of the efforts are denoted to construct a robust object appearance model for target/background separation. The main purpose of appearance modeling is to strengthen the distinctiveness of the object representation against the background. For achieving this, the object feature encoding is the key component. As for the feature encoding, existing appearance models over the decades can be categorized as encoding the low-level visual clues (e.g., the raw pixels, texture, color, gradient, feature points) (Bao et al., 2012; Duffner and Garcia, 2013; Mei and Ling, 2009; Zhang and van der Maaten, 2014), mid-level clues (superpixel as the main one Yang et al., 2014), and high-level structural or semantical features, such as the context priori (Wen et al., 2014; Zhang et al., 2014b; Zhu et al., 2014), subspace projection (Li et al., 2013a; Ross et al., 2008), saliency maps (Li et al., 2014; Mahadevan and Vasconcelos, 2013; Su et al., 2014) and deep hierarchical features Wang et al. (2015). For different levels of features, high-level ones have the straightforward meaning for object representation but are difficult to obtain while low-level visual ones are easy to extract but cannot represent the target content robustly. Recently, some trackers (Yang et al., 2014; Yuan et al., 2014) have utilized the superpixel as a trade-off to model the object appearance. However, superpixel segmentation is time consuming Borji et al. (2014). Hence, the low-level visual features still are the main choice in trackers' designing, such as the color information in the newest works (Liang et al., 2015; Zhang et al., 2014a), and multi-feature fusion is a common strategy for improving the tracking performance (Lan et al., 2014; Wang et al., 2013; Wang et al., 2014; Zhang et al., 2013). Despite of the demonstrated success of feature integration for tracking, the feature fusion may increase the computational burden (Liang et al., 2015) mainly because of the large-dimension issue.

To this end, hash-learning recently attracts the attention of researchers significantly in many computer vision applications, such as image retrieval (Chen et al., 2014; Liu et al., 2014) as well as tracking (Du et al., 2015; Li et al., 2013c; Ma and Liu, 2015). For example, Ma and Liu (2015) constructed a two-dimensional hashing method for appearance modeling. Du et al. (2015) proposed a

* Corresponding author.
E-mail addresses: crabwq@nwpu.edu.cn, crabwq@gmail.com (Q. Wang).

http://dx.doi.org/10.1016/j.cviu.2017.03.006
1077-3142/

tracker based on discriminative supervised learning hashing. With the learned hash functions, all target templates and candidates are mapped into compact binary space. Then, SVM classifier is exploited to consider the discriminative information between samples with different labels. However, the existing hash-based trackers updates the hash functions with all the samples collected in new frame via a batch mode, which is not a real online manner claimed to be efficient for large-scale object tracking. In addition, these hashing trackers collect positive and negative templates in each frame simultaneously. Nevertheless, positive templates often are difficult to accurately gather because of matching error.

In this paper, we propose a tracking method via an online pair-wise hashing algorithm. In the first frame, we sample some positive and negative templates for hashing function initialization. Then, the newly collected pair of templates is accommodated into the hash-code based appearance model with a kernel mapping based passive-aggressive learning strategy (Crammer et al., 2006). With this pair-wise strategy, the relationship between the positive templates and negative templates can be appropriately preserved that is very useful for visual tracking. In addition, with the encoded hash-code based features, the between-frame templates can be efficiently matched. Although the online manner can adapt to the dynamic scene effectively, the error accommodation and propagation are the main issue in online learning (Grabner and Bischof, 2006). Therefore, this paper further builds a template pool to collect the truly positive template in each frame, and updates it with an effective and efficient way. With the constructed positive template pool, the object appearance model in our online hashing can be robustly corrected when degradation occurs because of heavy occlusion, appearance changes, and so on. In summary, our tracker can be ranked into the discriminative framework (Babenko et al., 2011b; Bai et al., 2013; Zhang et al., 2015), focusing on building online classifiers to distinguish the target from the background, in contrast to the generative-based (Kwon et al., 2014; Zhang et al., 2014b) focusing on learning the target appearance model and searching the most similar region according to the learned template model.

All the discriminative trackers need to face the template sampling issue. The existing sampling methods in tracking, such as window sliding (Li et al., 2013c), gradient descent (Zhou et al., 2009), stochastic sampling (Li et al., 2013a; Zhong et al., 2014), dense sampling (Zhang et al., 2014b), etc., commonly treat the target center in the previous frame as the starting point for sampling. However, the object in current frame may demonstrate large displacement relative to the previous time, e.g., abrupt motion occurrence, which may make the sampled templates with poor quality, and cause severe drift. Therefore, this paper extracts template sampling center by a multi-clue based spatio-temporal saliency model (will be explained in Section 4). By that, the salient target region in each frame is detected, and can produce greater improvement in our tracking.

To summarize, our contributions are three-fold:

(1) This work firstly proposes a real online hashing tracker. Different from previous ones, the hashing function is online learned by new pairs of collected templates received sequentially, in which the relationship between the positive templates and negative templates can be appropriately preserved that is more useful for visual tracking.

(2) We build a positive template pool as a memory buffer for object depiction where representative true positive target templates are stored and used to restrain the object appearance model from degradation in online hashing.

(3) We introduce a multi-clue based spatio-temporal saliency model to guide the template sampling. Different from the other saliency models in tracking, we can adaptively select the best clue for object saliency computation and generate a saliency map with good distinction for object and background when illumination or appearance change occurs.

We thoroughly evaluate our tracker with state-of-the-art trackers on 50 challenging sequences, and the results demonstrate that our tracker performs favorably better than the state-of-the-arts.

The remainder of this paper is as follows: Section 2 presents the related works to this paper. Section 3 gives the tracker overview. Section 4 depicts the spatio-temporal saliency model for guiding the template sampling. Section 5 describes our online hashing method for visual tracking. Experiments and discussions are given in Section 6. The conclusion is finally summarized in Section 7.

2. Related works

Based on the motivation of this work, the literatures on hashing methods and hashing trackers are presented as follows.

2.1. Hashing methods

Hashing methods recently attract attention of researchers for approximating the nearest neighbors in machine learning field, and are successfully applied in image retrieval (Chen et al., 2014; Liu et al., 2014), face indexing (He et al., 2015), and so on. The main characteristic of hashing aims to map the high-dimensional data to a compact binary code, solving the scale and dimension increasing problem with a constant time. Generally, hashing methods can be categorized as data-independent ones and data dependent ones (Chang, 2012). In the data-independent ones, locality-sensitive hashing (LSH) (Charikar, 2002), as the most representative one, constructed hash functions with random projection. Later, the variants of LSH, such as min-hash (Chum et al., 2008) and $\ell_{p,\, p \in (0,\, 2]}$-stable hashing (Datar et al., 2004), obtained good performance in pattern recognition and computer vision applications. However, data-independent hashing methods have an issue that pursuing high-accuracy needs long hash codes to guarantee, which loses the original intention of hashing. Hence, data-dependent hashing approaches have become the main kind, where the label information, distribution and similarity of data can be exploited effectively. Based on the utilization of label information, data-dependent hashing methods can be further divided into unsupervised (Weiss et al., 2008), supervised (Chang, 2012), and semi-supervised methods (Wu et al., 2013a). Among these methods, supervised and semi-supervised hashing approaches take the label information or pair-wise correlation of data into account. For example, Weiss et al. (2008) achieved a spectral hashing method which solves nonlinear functions along the principal components of the data. Liu et al. (Chang, 2012) proposed a kernel-based supervised hashing method which minimizes on similar pairs of data and simultaneously maximizes on dissimilar pairs of data, and achieved a high-quality of hashing. Wang et al. (Wu et al., 2013a) constructed a semi-supervised nonlinear hashing using bootstrap sequential projection learning. Although the data-dependent hashing methods generate a higher performance than data-independent ones, they are all batch mode learning that learns the hash functions once for all data, which is impractical for large-scale sequential data application. For overcoming the above problems, Huang et al. (2013) proposed an online hashing method, and online manner begins to attract the attention in hashing methods, such as the adaptive hashing (Cakir and Sclaroff, 2015) proposed by Cakir and Sclaroff and the online sketching hashing (Leng et al., 2015). Inspired by the efficient feature encoding for large-scale data via online hashing methods, this paper proposes a robust visual tracking method via online hashing.

2.2. Hashing trackers

Owing to the efficient feature encoding, some hash-based trackers have been constructed. For instance, Li et al. (2013c) proposed a tracking method by learning the compact binary code, in which positive and negative samples are encoded by a randomized decision trees. By growing the leaf-node of trees, the feature fusion was conducted. Although this method obtains an effective tracking performance, it needs a sample buffer to store the positive and negative samples, discarding the oldest ones when the buffer overflows, and updating the hash function with all the samples in new frame by an incremental latent-structured support vector machine (LS-SVM). Ma and Liu (2015) proposed a two-dimensional hashing method for visual tracking. By hashing each image patch into independent hash functions, the similarity of the templates is computed by matching the hashing functions of registered pair-wise patches. Apparently, the similarity measurement is not suitable for deformative objects. Although the work of Ma and Liu (2015) utilized an incremental 2DLDA-like model to update the hash functions, the updating needs to train all the samples in current frame with a batch mode. It is similar to the work of discriminative hash tracker Du et al. (2015) proposed recently. Besides, these hash-based trackers need to gather positive and negative samples in each frame simultaneously. However, the positive samples sometimes are difficult to collect accurately because of matching errors. Hence, we utilize an effective strategy that only the negative templates are used to update the hash functions, and the object representation is strengthened with a positive template memory buffer collected timely.

3. Tracker overview

In this work, we formulate the visual tracking problem within the particle filtering framework. In this context, the target state $\mathbf{x}_t$ at time t can be determined by the posteriori probability $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, denoted as:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = p(\mathbf{y}_t|\mathbf{x}_t)\int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}, \quad (1)$$

where $\mathbf{y}_{1:t}$ is the observation set of target up to time t, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the motion model, and $p(\mathbf{y}_t|\mathbf{x}_t)$ specifies the observation model evaluating the likelihood. With that, the optimal target state $\hat{\mathbf{x}}_t$ at time t is obtained by the maximum a posteriori (MAP) over a set of templates,

$$\hat{\mathbf{x}}_t = \arg\max_{\mathbf{x}_t} p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1}). \quad (2)$$

With respect to the motion model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, most of the works formulate it by a Gaussian function (Jia et al., 2012; Zhong et al., 2014) $\mathcal{N}(\mathbf{x}_t; \mathbf{x}_{t-1}, \Sigma)$ with mean $\mathbf{x}_{t-1}$ as the target state, containing six parameters which are horizontal translation, vertical translation, rotation angle, scale, aspect ratio, and skew, in the previous time and the diagonal covariance matrix Σ. However, the object between frames may demonstrate large motion displacement (see Fig. 1). Hence, the sampled templates are prone to contain rather a little portion of positive ones which are easy to cause the confusing observation. Motivated by the works (Mahadevan and Vasconcelos, 2013; Su et al., 2014), the multi-clue saliency measurement is successfully embedded into object tracking for guiding the localization.

However, the existing trackers, implanting the saliency strategy, do not evaluate the importance of different feature clues and only utilize the saliency evaluation on the current frame, which may degrade the tracking performance when encountering the background clutter and illumination changes. Therefore, in this work, we propose a spatio-temporal saliency model (explained in Section 4) to guide the template sampling and obtain an adaptive selection on different feature clues when extracting the salient region.

In terms of the observation model $p(\mathbf{y}_t|\mathbf{x}_t)$, this work formulates it with the proposed online hashing mechanism which will be depicted in Section 5 in detail. The visual flowchart of the proposed tracker which is simplifying named it as **OKH** is summarized as Fig. 2.

4. Template sampling with spatio-temporal saliency guidance

In this paper, we formulate the motion model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ similarly as a Gaussian distribution $\mathcal{N}(\mathbf{x}_t; \tilde{\mathbf{x}}_{t-1}, \Sigma)$, while $\tilde{\mathbf{x}}_{t-1}$ is differently not the target state in the previous time. Instead, we propose a spatio-temporal saliency model to search the most appropriate location for $\tilde{\mathbf{x}}_{t-1}$, so as to make $\mathbf{x}_t$ as close to the true target position in current frame as possible. Hence, our goal for effective template sampling is converted to find the important region in both inter frames and inner frame with a temporal and spatial consideration. In this work, the perspectives of color, position and gradient are incorporated.

4.1. Spatial saliency

It is well known that in the particle filter, the positive templates are drawn around the target location within a radius of a few pixels, and negative templates consist of images some pixels farther away from the target location within an annular region. In this work, we set the size of the annular region is twice the size of the target region. Therefore, for the salient region at time t, different from the work Su et al. (2014) detecting the salient region in the whole image, this work finds the important region in a local rectangle image region R_t centered in the target location of $\mathbf{x}_{t-1}$ at $(t-1)th$ frame, composing the tracked target region at time $t-1$ and a larger annular region surrounding it, as shown in Fig. 1(a). The purpose is to suppress the influence of the complex environments with less time cost. Then, two useful cues are adopted, i.e., local contrast and location heuristic (Yan et al., 2014).

4.1.1. Local contrast

Local contrast computes the salient value for each image pixel to its surrounding neighborhood. The region with higher value of local contrast is more attractive to human eyes (Borji et al., 2014). The local contrast value LC_i of pixel $\mathbf{p}_i$ in R_t is computed by comparing it with all the other pixels in R_t, denoted as:

$$LC_i = \sum_{j,j\neq i} \omega(\mathbf{p}_i, \mathbf{p}_j)\|\mathbf{f}_i - \mathbf{f}_j\|_2, \quad (3)$$

where $\omega(\mathbf{p}_i, \mathbf{p}_j)$ is the distance between $\mathbf{p}_i$ and $\mathbf{p}_j$, and is set as $e^{-D(\mathbf{p}_i,\mathbf{p}_j)/\lambda^2}$ controlling spatial distance influence between two pixels, where λ^2 is the parameter controlling the size of surrounding neighborhood. $D(\cdot)$ is the square of Euclidean distances between $\mathbf{p}_i$ and $\mathbf{p}_j$. $\mathbf{f}_i$ and $\mathbf{f}_j$ are the feature vector of $\mathbf{p}_i$ and $\mathbf{p}_j$, respectively. Eq. (3) indicates that closer pixels have larger impact than distant ones.

4.1.2. Location heuristic

Location heuristic evaluates the location impact to image regions. Based on the observation, human attention favors image center Borji et al. (2014). Hence, location heuristic value LH_i of pixel $\mathbf{p}_i$ in R_t is calculated by:

$$LH_i = \exp\{-\|\mathbf{l}_i - \mathbf{l}_c\|^2/2\gamma^2\}, \quad (4)$$

where $\mathbf{l}_i$ and $\mathbf{l}_c$ are the coordinate of $\mathbf{p}_i$ and the center of R_t, respectively, and γ^2 is the parameter controlling the window size for normalizing LH_i.

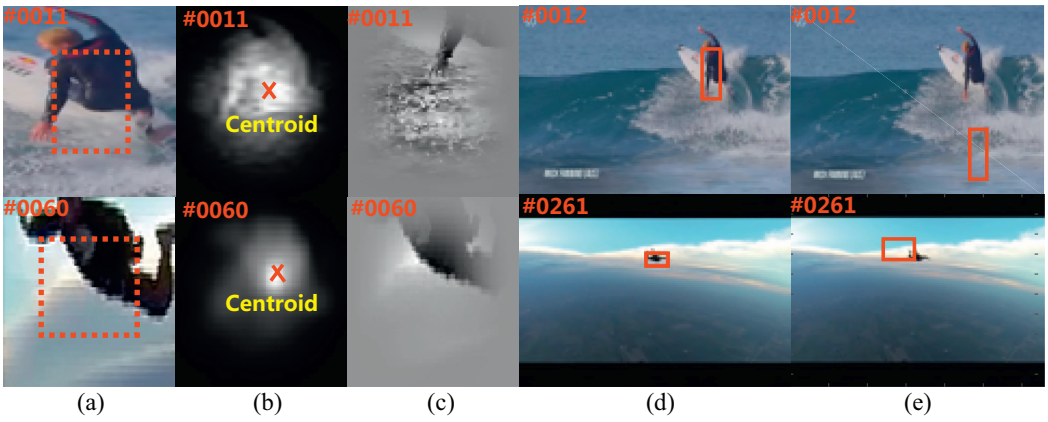

Fig. 1. Comparison of typical examples for template sampling guidance. (a) are the original color image region with an object marked by dashed line; (b) are the obtained salient regions by our spatio-temporal saliency model; (c) are the results generated by a newly proposed spatio-temporal contextual model Zhang et al. (2014b); (d) and (e) are the tracked results by our model and Zhang et al. (2014b), respectively.

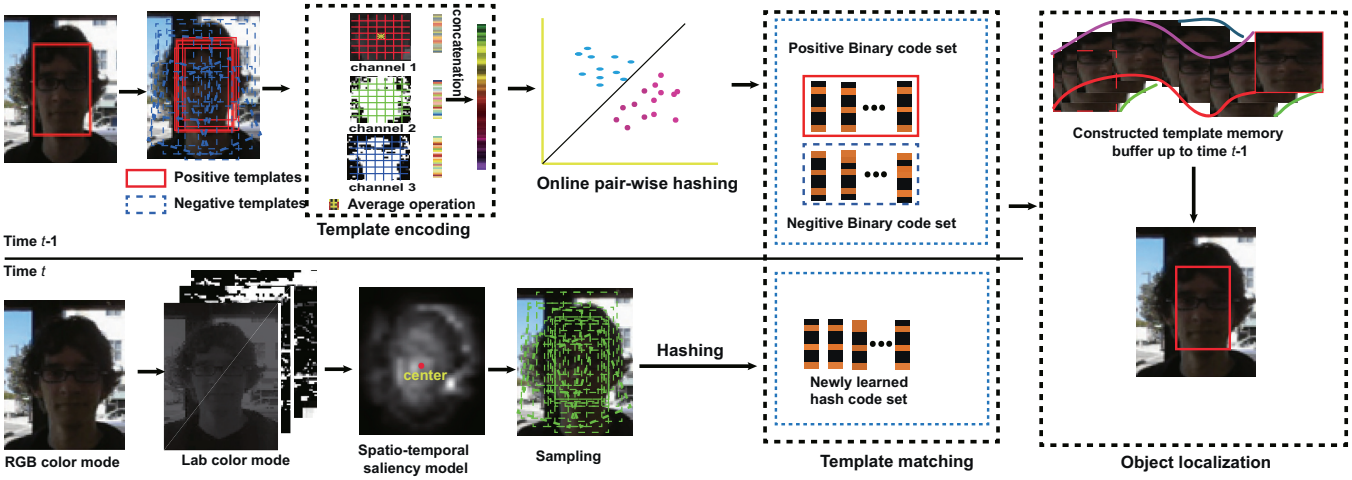

Fig. 2. Flowchart of the proposed tracker.

Consequently, the spatial saliency of pixel $\mathbf{p}_i$ is computed by combining h_i with c_i, and defined as:

$$s_i = LH_i * LC_i. \tag{5}$$

After computing the spatial saliency value for each pixel, we obtain a saliency map $S^f = \{s_i\}_{i=1}{}^N$ for one kind of feature, such as color, where f is the feature index and N is the pixel number. Assume that there are F spatial saliency maps for F kind of features are available, most of the saliency methods commonly adopt the summation (Li et al., 2013b) or inner-product (Yan et al., 2014) operation to combine them. It is obvious that these fusions are limited for object tracking, e.g., the environment with complex background clutter. Hence, we select them adaptively with a temporal consideration (see Section 4.2).

4.2. Spatio-temporal saliency

In order to integrate the temporal saliency, to be specific, we exploit the temporal statistics of the representation difference of object and background with respect to different features over time to adaptively weigh the visual features for saliency computation.

For achieving this, we first utilize the *histograms* to model the representation distribution of the object region and the annular background region (as shown in Fig. 1(a)) for the fth kind of feature (e.g., color). Second, we formulate the pair-wise *histogram* difference Δe_t^f of the object and background with respect to feature f at time t using χ^2 distance:

$$\Delta e_t^f = \chi^2(\mathbf{H}_t^{ob}, \mathbf{H}_t^{bc}), \tag{6}$$

where $\chi^2(\mathbf{b_1}, \mathbf{b_2}) = \frac{1}{2}\sum_{i=1}^{b} \frac{|\mathbf{b}_1(i)-\mathbf{b}_2(i)|^2}{\mathbf{b}_1(i)+\mathbf{b}_2(i)}$ is with b number of bins, and $\mathbf{H}_t^{ob}$ and $\mathbf{H}_t^{bc}$ are the normalized histograms for object and background representation. Δe_t^f is used to model and update the temporal statistics of the representation difference of object and background with respect to the fth kind of feature in following. Third, this paper utilizes a Gaussian distribution $\mathcal{N}(\mu_t^f, \Sigma_t^f)$ to model the temporal statistics of the representation difference of object and background to the fth kind of feature over time. With that, the weight of different visual feature in the tth frame for saliency computation is computed as:

$$w_t^f = \mu_t^f / \sum_{f=1}^{F} \mu_t^f, \tag{7}$$

where $\mu_t^f = \delta\mu_{t-1}^f + (1-\delta)\Delta e_t^f$, $\Sigma_t^f = \delta\Sigma_{t-1}^f + (1-\delta)\Sigma_t^f$, and δ is a learning rate balancing the current object-background representation difference with the historical ones (set as 0.3 for this work).

Eq. (7) is prone to select the feature with larger representation difference for object and background, which indicates that the spatial saliency map with more salient object region is favored. With the temporal consideration, the spatio-temporal saliency at time t

is written as:

$$S_t = \sum_{f=1}^{F} w_t^f S_t^f. \tag{8}$$

Taking Fig. 1 as an example, the computed saliency map can robustly find the most salient region in each frame. Then the object center for $\tilde{\mathbf{x}}_{t-1}$ is computed by the object-biased measurement, see work of Li et al. (2013b) for detail.

5. Online hashing for visual tracking

In this section, we will give the online hashing method for visual tracking. Firstly, a brief review about the traditional supervised hashing is given. Next, the tracking method via online hashing is presented. Specially, in the online hashing tracking, we construct a true positive template pool collected in historical time to restrain the appearance degradation of online hashing, and make the tracking performance improved.

5.1. Supervised hashing

The goal of supervised hashing is to generate a R-bit binary code $h \in \{-1, 1\}^R$ for each training sample whose label information is exploited, i.e., obtaining the hash functions $\{h_r(\mathbf{y})\}_{r=1}^R$ by learning the hash projection vector $\mathbf{w}_r \in \mathbb{R}^d$, where $\mathbf{y} \in \mathbb{R}^D$ is the observed feature vector, and D is the feature dimension. It is worth noting that in order to make the sample more separable, d samples are usually selected randomly as the references, and kernel trick (Chang, 2012) is commonly utilized to map the original feature set to references, e.g., mapping $\mathbf{y} \in \mathbb{R}^{1\times D}$ to $\mathbf{z} \in \mathbb{R}^{1\times d}$. Generally, the linear projection hashing is utilized owing to its efficiency and simplicity, and defined as:

$$h_r(\mathbf{z}) = sgn(\mathbf{w}_r^T \mathbf{z} + b_r), 1 < r < R, \tag{9}$$

where b_r is a threshold, and sgn is the sign function. For clearer description, we define $\mathbf{w}_r$ as $[\mathbf{w}_r^T, b_r]^T$, and redefine $\mathbf{z}$ to be $[\mathbf{z}^T, 1]^T$, and the hash function becomes:

$$\mathbf{h}(\mathbf{z}) = sgn(\mathbf{W}^T \mathbf{z}), \tag{10}$$

where $\mathbf{W}^T \in \mathbb{R}^{R\times(d+1)}$ is the hashing projection matrix and $\mathbf{h} = [h_1(\mathbf{z}), h_2(\mathbf{z}), \ldots, h_R(\mathbf{z})]^T$. Because Eq. (10) is not differentiable, its optimization is hard to conduct. Many hashing methods utilize the structured SVMs (Finley and Joachims, 2008) to solve $\mathbf{W}^T$, i.e.:

$$\mathbf{f}(\mathbf{z}) = \arg \max_{\mathbf{h}\in\{-1,1\}^R} \mathbf{h}^T \mathbf{W}^T \mathbf{z}. \tag{11}$$

In general, the label information of training samples is obtained by evaluating the similarity of their binary codes, and Hamming distance is widely adopted. If the samples are more similar, Hamming distance should be smaller.

With the label information and projection strategy, most of the hashing methods learn the hashing functions once for all samples, e.g., random trees (Charikar, 2002), which are impractical for large-scale data. Fortunately, Huang et al. (2013) proposed an online hashing method recently. In each learning rounds, there is only one pair of samples received, and is in sense of a passive aggressive strategy (Grabner and Bischof, 2006). Because of the pair-wise passive aggressive strategy, the relationship of different kind of samples are appropriately preserved. Motivated by that, this paper for the first time introduces the online hashing algorithm (Huang et al., 2013) into object tracking.

5.2. Online hash tracking

In this subsection, we will give the whole tracking procedure. Firstly, we sample some positive and negative templates by the predefined Gaussian motion model assisted by a guidance of the spatio-temporal saliency aforementioned in Section 4, and encode them with an exploitation of color information. Secondly, the core online hashing code learning for sequentially arrived template features is given. Thirdly, we construct an efficient and effective strategy for between-frame template matching, and select the candidates with top matching scores for true target approximation. Finally, we further build true template memory buffer for object description to select the best template candidate as target.

5.2.1. Feature encoding

As claimed that local representations for template are superior to holistic ones, and motivated by the work (Liang et al., 2015) which proves Lab color space almost obtained the largest gain for boosting the tracking performance, we utilize a patch-based color template representation. For a color template $T \in \mathbb{R}^{c\times a\times b}$ after resizing, where c is the channel number of Lab space, and $[a, b]$ is the size of the template. In order to obtain a pure representation, we slide each color template with a 4×4 window to filter the inevitable noise pixels because of imaging condition. Then, the average of each sliding window is computed, and all of them of a template in each channel are stacked into a feature vector. The color template is represented by concatenating the vectors obtained in three color channels. Here, we denote the feature vector for each color template as $\mathbf{y} \in \mathbb{R}^{1\times D}$, where D is the dimension of each feature vector. Specially, $\mathbf{y}$ is normalized into [0, 1] by ℓ_2-norm.

5.2.2. Online hash code learning

In the first frame, we draw m positive templates $\{\mathbf{y}_i\}_1^m \in \mathbb{R}^{m\times D}$ around the target location within a radius of a few pixels, and n negative templates $\{\mathbf{y}_j\}_1^n \in \mathbb{R}^{n\times D}$ some pixels further away from the target location within an annular region, where D is the feature dimension. Then, for a pair of feature vector, i.e., $\mathbf{y}_i$ and $\mathbf{y}_j$, this paper, different from Huang et al. (2013), incorporates both the location and appearance of them into the similarity evaluation, which is defined as:

$$s(\mathbf{y}_i, \mathbf{y}_j) = d(\mathbf{y}_i, \mathbf{y}_j) \exp\{-\|\mathbf{l}_i - \mathbf{l}_j\|^2/\sigma\}, \tag{12}$$

where $d(\mathbf{y}_i, \mathbf{y}_j) = \frac{1}{Z}\|\mathbf{y}_i - \mathbf{y}_j\|^2$ is the standard Euclidean distance, where Z is the factor for normalization, and $\mathbf{l}_i$ and $\mathbf{l}_j$ are the position of $\mathbf{y}_i$ and $\mathbf{y}_j$. With a normalization of $s(\mathbf{y}_i, \mathbf{y}_j) \in [0, 1]$, the similarity label of $\mathbf{y}_i$ and $\mathbf{y}_j$ are given as: $s = 1$, if the normalized $s(\mathbf{y}_i, \mathbf{y}_j)$ is less than 0.01, and $s = -1$ for vice versa. With the similarity labels, the loss of the learned hash codes in this work is defined as $loss(\mathbf{H}, s) = H(\mathbf{h}_i, \mathbf{h}_j)$, if $s = 1$, and $loss(\mathbf{H}, s) = R - H(\mathbf{h}_i, \mathbf{h}_j)$, if $s = -1$, where $H(\cdot)$ is the Hamming distance, and $\mathbf{H} = [\mathbf{h}_i, \mathbf{h}_j]$ is the learned pair-wise hash code of feature pair $[\mathbf{y}_i, \mathbf{y}_j]$. Note that compared with the appearance measurement, Eq. (12) also exploits the smoothness of the target state between frames.

With the computed similarity labels of templates, the next work is to learn the projection matrix $\mathbf{W}$ for template hashing. In this work, $\mathbf{W}$ is initialized as $\mathbf{W}_{LSH} \in \mathbb{R}^{(d+1)\times R}$ sampled from a zero-mean multivariate Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$, where $d < (m + n)$ is the number of templates that are randomly selected as the references for feature mapping. In this work, we follow Chang (2012) to map the feature space. Assume the learned projection matrix in lth rounds to be $\mathbf{W}_l$. Meanwhile, we suppose that the newly received feature pair $\mathbf{Y}_l = [\mathbf{y}_i^l, \mathbf{y}_j^l]$ is mapped into $\mathbf{Z}_l = [\mathbf{z}_i^l, \mathbf{z}_j^l]$, and we can use the labeled information which tells the pair is dissimilar or not to generate an optimal hash code $\mathbf{G}_l = [\mathbf{g}_i^l, \mathbf{g}_j^l]$. Here, the pair-wise hash code $\mathbf{H}_l = [\mathbf{h}_i^l, \mathbf{h}_j^l]$ is obtained by using Eq. (11). In order to preserve the information already learned, the newly learned $\mathbf{W}_{l+1}$ should stay as close to $\mathbf{W}_l$ as possible (Huang et al., 2013). Hence, the objective function for learning $\mathbf{W}$ is denoted as:

$$\mathbf{W}_{l+1} = \arg\min_{\mathbf{W}} \frac{1}{2}\|\mathbf{W} - \mathbf{W}_l\|^2 + C\xi, \ \xi > 0, \tag{13}$$

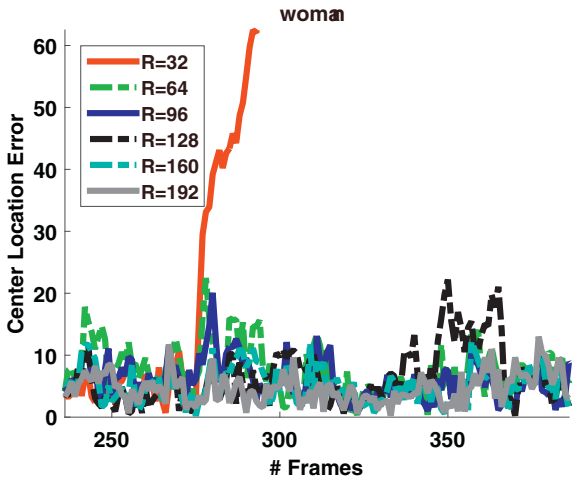

Fig. 3. Tracking performance on "woman" sequence with different hash bits.

where C is a positive constant named as aggressiveness parameter, ξ is a non-negative variable for minimizing the prediction loss. In order to get a better solution of $\mathbf{W}_{l+1}$, the optimization of Eq. (13) forces the learned hash codes of $\mathbf{Z}_l = [\mathbf{z}_i^l, \mathbf{z}_j^l]$ to approximate the optimal hash code $\mathbf{G}_l$ as close as possible. Hence, the prediction-based loss function $\ell(\mathbf{W}, \mathbf{H}_l, \mathbf{G}_l, s)$ (Crammer et al., 2006) is introduced, written as:

$$\begin{aligned}\ell(\mathbf{W}, \mathbf{H}_l, \mathbf{G}_l, s) &= H_l(\mathbf{W}) - G_l(\mathbf{W}) + \sqrt{loss(\mathbf{H}_l, s)} \\ &= (\mathbf{h}_i^l - \mathbf{g}_i^l)\mathbf{W}^T\mathbf{z}_i^l + (\mathbf{h}_j^l - \mathbf{g}_j^l)\mathbf{W}^T\mathbf{z}_j^l + \sqrt{loss(\mathbf{H}_l, s)}.\end{aligned} \tag{14}$$

It is clear that if $\ell(\mathbf{W}, \mathbf{H}_l, \mathbf{G}_l, s)=0$, $G_l(\mathbf{W}_{l+1}) = H_l(\mathbf{W}_{l+1}) + \sqrt{loss(\mathbf{H}_l, s)} > H_l(\mathbf{W}_{l+1})$. It indicates that the temporary value of $\hat{\mathbf{g}}_l$ in optimization is prone to the optimal hash code set $\mathbf{G}_l$ rather than $\mathbf{H}_l$. By adding the constraint $\ell(\mathbf{W}, \mathbf{H}_l, \mathbf{G}_l, s) > \xi$ to Eq. (13), Eq. (13) can be solved by Lagrangian optimization (Huang et al., 2013). Different from the previous hash trackers, the learning processes each feature pair sequentially, which can preserve the relationship of samples than the batch-mode strategy, and is more adequate for tracking.

To make a concise description, this work only discusses the case of dissimilar pairs and the inference for similar pairs can be similarly developed. For the optimal hash code set $\mathbf{G}_l$ inference, we derive a difference $\pi = H(\mathbf{g}_i^l, \mathbf{g}_j^l) - H(\mathbf{h}_i^l, \mathbf{h}_j^l)$, where $H(\cdot)$ is the Hamming distance. In the inference, aiming at updating $\mathbf{W}$ as few as possible, and preserving the learned information as more as possible, this work picks up the bits whose indexes satisfy $H(\mathbf{h}_{i(r)}^l, \mathbf{h}_{j(r)}^l) = 1$, and set π as $R - H(\mathbf{h}_i^l, \mathbf{h}_j^l)$. Then, the contribution of each bit of hash code is determined by $\rho_r = \max(\mathbf{h}_{i(r)}^l \mathbf{w}_r^{l\,T} \mathbf{z}_i^l, \mathbf{h}_{j(r)}^l \mathbf{w}_r^{l\,T} \mathbf{z}_j^l)$. Sort ρ_r and select the corresponding hash bits of the π largest ρ_r to be different from $\mathbf{H}_l$ and others to the same as $\mathbf{H}_l$. The resulted pair-wise hash code is denoted as $\mathbf{G}_l$.

As for the hash bits R in this work, we run the "*woman*" sequence with different hash bits (Set as 32, 64, 96, 128, 160, 192, respectively). The tracking results are shown in Fig. 3. We notice that too short hash bit is not reasonable because of the poor ability of information preservation, and longer hash bit upgrades the performance finitely. Considering the performance and the efficiency, this work fixes the hash bits as 128 in all the experiments.

5.2.3. Template matching

In this work, given the hash projection matrix $\mathbf{W}_{t-1}$ learned at $(t-1)$th frame, we can derive the hash code sets for positive and negative templates by Eq. 10, specified as $\mathbf{H}_{t-1}^+ = [\mathbf{h}_1^+, \ldots, \mathbf{h}_m^+]$ and $\mathbf{H}_{t-1}^- = [\mathbf{h}_1^-, \ldots, \mathbf{h}_n^-]$, where $\mathbf{h} \in \{-1, 1\}^R$ is the hash code of each template. For the new template set sampled at the newest frame, this work first hashes them into the hash codes $\mathbf{H}_t$, and then presents an efficient and effective object matching method, and the matching score $\mathbf{S}$ of $\mathbf{H}_t$ is computed by:

$$\mathbf{S} = \overline{\mathbf{H}_t^T \mathbf{H}_{t-1}^+} - \overline{\mathbf{H}_t^T \mathbf{H}_{t-1}^-}, \tag{15}$$

where $\bar{\cdot}$ is the average operation in row direction. The hidden meaning of Eq. (15) is to make the matched target template be more similar to the ones of the target and dissimilar to the ones of the background. Although online hashing is more efficient than the batch-mode ones, it also has the issue of the error accommodation existing in online learning. Hence, after template matching, in order to obtain a more accurate result, we sort $\mathbf{S}$ in a descender order, and select template candidates with top Q matching score for a further examination. For this purpose, we build a true positive template memory buffer collected timely in following, and select the best target template as the tracked result.

5.3. Object localization

Assume the constructed true positive template pool is $\mathcal{T}_{t-1}(O_p, \alpha_p), p = 1, 2, \ldots, P$ at time $t-1$, where $\{O_p\}_1^P$ is the positive template set collected timely, $\{\alpha_p\}_1^P$ is the related weight set for $\{O_p\}_1^P$, and P is the size of the template pool. Given the template candidates $\{C_q\}_1^Q$ with top Q matching scores at time t, this paper goes back to compute the similarity of their original color-based feature vector to the ones of the $\mathcal{T}_{t-1}(O_p, \alpha_p)$. That is because original feature vector has richer information than the hash codes. For the similarity measurement, this paper proposes a distance transform:

$$\beta_q = \sum_{p=1}^{P} \alpha_p \left(\max_{i=1}^{D} |O_{p(i)} - C_{q(i)}| - \min_{i=1}^{D} |O_{p(i)} - C_{q(i)}| \right) \tag{16}$$

where β_q is the confidence of C_q, D is the feature dimension of original feature vector, and $|\cdot|$ is the absolute operation. The proposed distance is similar to the minimum barrier distance (MBD) (Ciesielski et al., 2014) which computes the path cost of consecutive pairs of pixels in image, and is more robust to illumination changes and blur than geodesic distance and Euclidean distance (Ciesielski et al., 2014). That is because the operation of $\max(\cdot) - \min(\cdot)$ is prone to measure the difference of distributions. Differently, the distance transform in Eq. (16) is a weighted MBD (WMBD) which can better consider the importance of collected templates in localization. With that, the optimal target state at time t is determined by:

$$\hat{\mathbf{x}}_t = \arg \min_{\mathbf{x}_q \in \{\mathbf{x}_q\}_1^Q} \beta_q. \tag{17}$$

As for Eq. (16), another key component for the success of $\mathcal{T}_{t-1}(O_p, \alpha_p)$ is to determine the weight set $\{\alpha_p\}_1^P$. To this end, this paper proposes an adaptive strategy for updating $\{\alpha_p\}_1^P$. To be specific, we introduce a vote set $\{v_p\}_1^P$ for $\{O_p\}_1^P$ to represent the template importance, where the larger the value of v_p is, the more important the O_p is.

Weight set updating. At the tth $(t < P)$ frame, set the vote value of each collected template as 1, and their weight $\alpha_{1:t} = \frac{1}{\sum_{p=1:t} v_p}$. Meanwhile, put the newly collected template C_t and its weight α_t at the end of the template pool.

For the tth $(t > P)$ frame, measure the similarity of the templates in $\mathcal{T}_t(O_p, \alpha_p)$ with the target template C_T of the optimal target state $\hat{\mathbf{x}}_t$. Then, we re-compute WMBD between C_T and each

template in $\mathcal{T}_t(O_p, \alpha_p)$, and denoted as β_p. If $\beta_p < \check{\beta}$, $v_p = v_p + 1$, and reweigh α_p as $\alpha_p = \frac{v_p}{\sum_{p=1:P} v_p}$.

After that, the template pool is updated by removing the template with least weight, and putting the template sample of $\hat{\mathbf{x}}_t$ into its end. In order to prevent the newly collected template from discarding early, we set the vote of the newly collected true positive template as the mid-value of $\{v_p\}_1^P$.

Highlights. With this strategy for template updating, the target regions at the beginning frames are favored by the template pool. Therefore, we can find that the updating strategy on one hand can restrain the appearance degradation owing to the error accommodation in online hashing, and on the other hand can help to refind the target when heavy occlusion occurs. Although this strategy somewhat seems ad-hoc, it can boost the performance significantly.

5.4. Template updating

In this work, we only use the negative templates sampled some pixels farther away from the target location within an annular region to update the online hashing model. That is because the tracked target is easy to change, and may degrade the appearance model. Besides, the online hashing model is updated every 5 frames. Therefore, the object tracking in each frame can be achieved by merely conducting Eq. (15), and can boost the efficiency significantly.

6. Experiments and discussions

Because colorful videos are more common in practical applications, this paper evaluates the proposed tracker on 50 color sequences (named as OKH-Color50 (with 22190 frames)). They are collected from the widely used benchmark (Wu et al., 2013b) for online tracking and the newly proposed video benchmark (Liang et al., 2015). The frame initialization of OKH-Color50 dataset is shown in Fig. 4, and the confusion matrix of 10 challenging attributes and 7 object categories are demonstrated in Fig. 5(a) and (b), respectively. The challenging attributes (simplified as "*Attr*") concerned in this work are explained as that "*IV*", "*SV*", "*OCC*", "*DEF*", "*MB*", "*FM*", "*IPR*", "*OPR*", "*BC*", "*LR*" represent that the targets in the sequences are with illumination variation, scale variation, occlusion, non-rigid deformation, motion blur, fast motion, in-plane-rotation, out-plane-rotation, background clutter or low resolution challenging attributes, respectively. In all the sequences, the ground-truth of the target is labeled manually in each frame.

For evaluating the superiority of the proposed tracker, twelve state-of-the-art trackers are selected as the competitors. They are Frag (Adam et al., 2006), DFT (Sevilla-Lara, 2012), ASLA (Jia et al., 2012), MIL (Babenko et al., 2011a), OAB (Grabner and Bischof, 2006), VTD (Kwon and Lee, 2010), Struck (Hare et al., 2011), CT (Zhang et al., 2012), TGPR (Gao et al., 2014), MEEM (Zhang et al., 2014a) and the newly proposed HSKCF (high speed tracking with kernel correlation filters) (Henriques et al., 2015) and MUSTer (Hong et al., 2015). In order to obtain a fair comparison, the optimal parameters in all these trackers remain unchanged. In the meantime, this paper runs each tracker for five trials, and obtains the average performance as the baseline. For the performance evaluation, quantitative and qualitative ones are provided. Among them, quantitative evaluation utilizes two metrics:

1) Center location error (CLE) that measures the distance between the centers of the tracked target and the ground-truth.

2) Success rate (SR) that computes the percentage of frames where the overlapping rate $\frac{B_T \cap B_G}{B_T \cup B_G}$ is greater than a threshold (set as 50% empirically), where B_T and B_G are the bounding boxes of predicted target and related ground-truth, respectively, and $\cap$ and $\cup$ are the intersection and union of two regions, respectively.

Qualitative evaluation is given by demonstrating the snapshot frames of the sequence where the tracking results of the target are marked by different colorful bounding boxes.

6.1. Implementation setups

Our tracker runs at 9 fps on a platform with 3.60 GHz CPU and 4GB RAM. As mentioned before, the proposed tracker in this paper is ranked as the discriminate framework. The number of the positive and negative tracking templates in the first frame are set as 50 and 150, respectively. Besides, this work similarly utilizes the *affsig* parameters like (Jia et al., 2012) yet a less number of particles, (i.e., 300 is enough), because of the effectiveness of the proposed spatio-temporal saliency model. Besides, this work utilizes Lab color space and histogram of oriented gradient (HOG) to represent the color and gradient information of pixels when evaluating the spatio-temporal saliency, where the cell size of HOG operator is 8×8. For the templates, this paper resizes them as 32×32. Consequently, with the sliding window with the size of 4×4 in template encoding, the feature dimension of a color-based template is $225 \times 3 = 675$. In addition, the target candidate number Q in object localization is set as 20. The size of the constructed template memory buffer in this work is fixed to 10. In the online hashing, we need to determine the kernel function κ and the number of samples d for feature set mapping. In this work, we feed the online hashing the Gaussian RBF kernel ($\kappa(x, y) = \exp(-\|x - y\|^2/2\sigma^2)$) and d as 10 and 30 for positive templates and negative templates, respectively. The aggressiveness parameter C is set as 0.001. To make a fairer comparison, all the parameters are fixed in the experiments.

6.2. Evaluations

In order to demonstrate the superiority of the proposed tracker, we firstly give the overall performance evaluation on the 50 video sequences. Specifically, the precision plots and the success rate plots are shown in Fig. 6, where the precision plots are ranked by the result at the CLE threshold of 20 pixels and the success rate plots are ranked by the value of area under curve (AUC). Besides, we also demonstrate the performance in Table 1 and Table 2 to show the ability for tackling different challenging attributes by distinct trackers. It can be observed that MUSTer, MEEM and OKH can be generally ranked as the top three ones. Besides, one main advantage of hashing methods is that they can integrate more robust features flexibly without increasing the computation burden of training because of the fixed hash bits. Therefore, OKH is the best with overall evaluation.

To give a more thorough comparison, the detailed in-depth analysis of the comparison between OKH and the state-of-the-arts is illustrated as follows.

6.2.1. Background clutter

In this situation, the surrounding background of the target always distracts the tracker and causes drift easily. From the results in Fig. 7, we can observe that MUSTer and our tracker are manifestly superior to others. This benefits from: 1) the spatio-temporal saliency model in this work which owns the important role in this situation because of the adaptive selection for the visual clue (i.e., color or gradient) utilized for salient object extraction in each frame; 2) the pair-wise online hashing which can better preserve the relationship of positive and negative samples generating a robust target/background separation, and 3) the most important reason that MUSTer and OKH both have the long-term memory to store the historical template knowledge.

Fig. 4. Frame initialization of the OKH-Color50 dataset, where "*IV*", "*SV*", "*OCC*", "*DEF*", "*MB*", "*FM*", "*IPR*", "*OPR*", "*BC*", "*LR*" represent that the targets in the sequences are with *illumination variation, scale variation, occlusion, non-rigid deformation, motion blur, fast motion, in-plane-rotation, out-plane-rotation, background clutter* or *low resolution* challenging attributes, respectively.

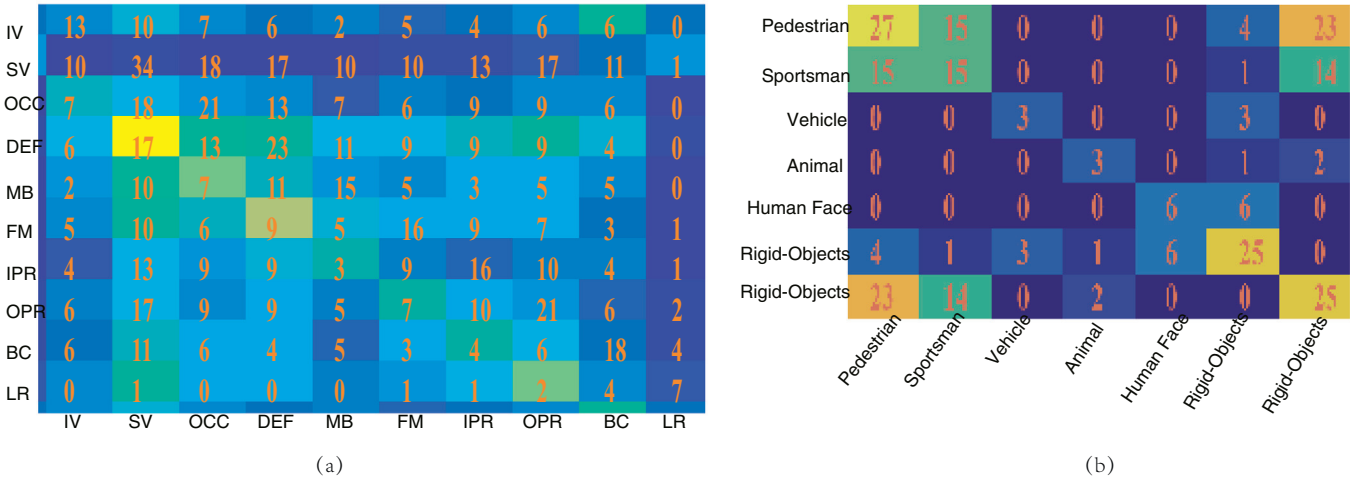

Fig. 5. The statistics of the video sequences. (a) The confusion matrix of the challenging attributes in OKH-Color50 dataset, and (b) the confusion matrix of the object categories.

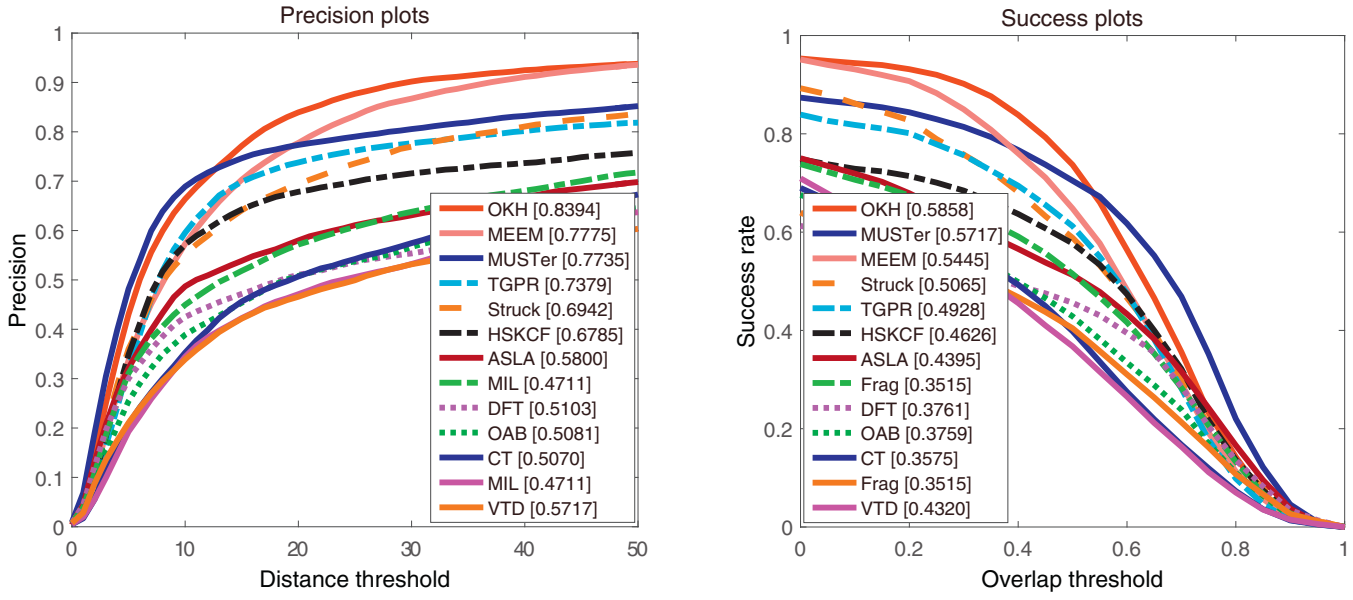

Fig. 6. Overall performance evaluation on OKH-Color50 dataset.

Table 1
The precision rate comparison corresponding to different challenging attributes for the competing trackers. For a clearer demonstration, the best tracker is masked by Red color, the second and the third are respectively tagged by Green and Blue color.

Attr	Frag	MIL	CT	OAB	DFT	VTD	ASLA	Struck	TGPR	MEEM	HSKCF	MUSTer	OKH
IV	0.357	0.423	0.451	0.283	0.419	0.542	0.475	0.584	**0.849**	0.714	0.599	**0.798**	**0.731**
SV	0.407	0.473	0.532	0.539	0.476	0.604	0.638	0.643	**0.756**	**0.748**	0.665	0.709	**0.754**
OCC	0.463	0.350	0.455	0.417	0.472	0.510	0.540	0.572	0.627	**0.780**	0.598	**0.785**	**0.810**
DEF	0.551	0.397	0.463	0.462	0.392	0.464	0.508	0.673	0.731	**0.762**	0.731	**0.817**	**0.868**
MB	0.601	0.494	0.613	0.592	0.591	0.715	0.588	0.776	**0.914**	**0.810**	0.796	0.796	**0.843**
FM	0.399	0.457	0.366	0.436	0.479	0.402	0.394	0.579	**0.687**	**0.600**	0.553	0.593	**0.695**
IPR	0.300	0.341	0.299	0.478	0.376	0.457	0.423	0.550	**0.663**	0.699	0.601	**0.651**	**0.718**
OPR	0.407	0.408	0.431	0.461	0.497	0.559	0.558	0.672	0.753	**0.834**	0.708	**0.791**	**0.773**
BC	0.490	0.496	0.573	0.592	0.588	0.628	0.593	0.759	0.832	**0.882**	0.745	**0.974**	**0.958**
LR	0.701	0.754	0.702	0.683	0.611	0.763	0.821	**1.00**	0.741	**1.00**	0.768	0.863	**0.960**

6.2.2. Non-rigid deformation and in-plane-rotation/out-plane-rotation

For non-rigid deformation challenge, the sequences usually have a non-rigid object which is likely to demonstrate deformation, such as the sportsman. The main difficulty in this situation is how to suppress the influence from the deformable part of the object as much as possible. For this challenge, Fig. 8 shows that our tracker is competitive to the top trackers, such as MEEM and MUSTer. One main reason is because of the robust template sampling with the spatio-temporal saliency guidance. Another important element is the hashing method that can preserve the main structure of the template which utilizes a patch-based encoding procedure.

Table 2
The success rate comparison corresponding to different challenging attributes for the competing trackers. For a clearer demonstration, the best tracker is masked by Red color, the second and the third are respectively tagged by Green and Blue color.

Attr	Frag	MIL	CT	OAB	DFT	VTD	ASLA	Struck	TGPR	MEEM	HSKCF	MUSTer	OKH
IV	0.292	0.336	0.345	0.272	0.331	0.428	0.415	0.453	0.525	**0.578**	0.430	**0.619**	**0.536**
SV	0.295	0.330	0.345	0.369	0.332	0.443	0.476	0.446	0.477	**0.522**	0.410	**0.507**	**0.491**
OCC	0.361	0.293	0.333	0.318	0.344	0.391	0.436	0.439	0.455	**0.521**	0.416	**0.552**	**0.558**
DEF	0.376	0.312	0.313	0.323	0.264	0.331	0.404	0.477	0.482	**0.516**	0.486	**0.583**	**0.566**
MB	0.422	0.366	0.406	0.405	0.414	0.512	0.398	0.565	0.597	**0.615**	0.531	**0.556**	**0.558**
FM	0.361	0.417	0.392	0.396	0.391	0.374	0.379	0.508	0.520	**0.521**	0.423	**0.532**	**0.576**
IPR	0.267	0.348	0.290	0.405	0.312	0.429	0.399	0.468	0.501	**0.563**	0.430	**0.528**	**0.543**
OPR	0.315	0.311	0.328	0.345	0.369	0.421	0.429	0.477	0.496	**0.534**	0.467	**0.588**	**0.566**
BC	0.356	0.335	0.384	0.405	0.421	0.452	0.405	0.540	0.538	**0.640**	0.520	**0.683**	**0.611**
LR	0.450	0.354	0.326	0.415	0.402	0.472	0.473	**0.610**	0.343	0.573	0.456	**0.571**	**0.638**

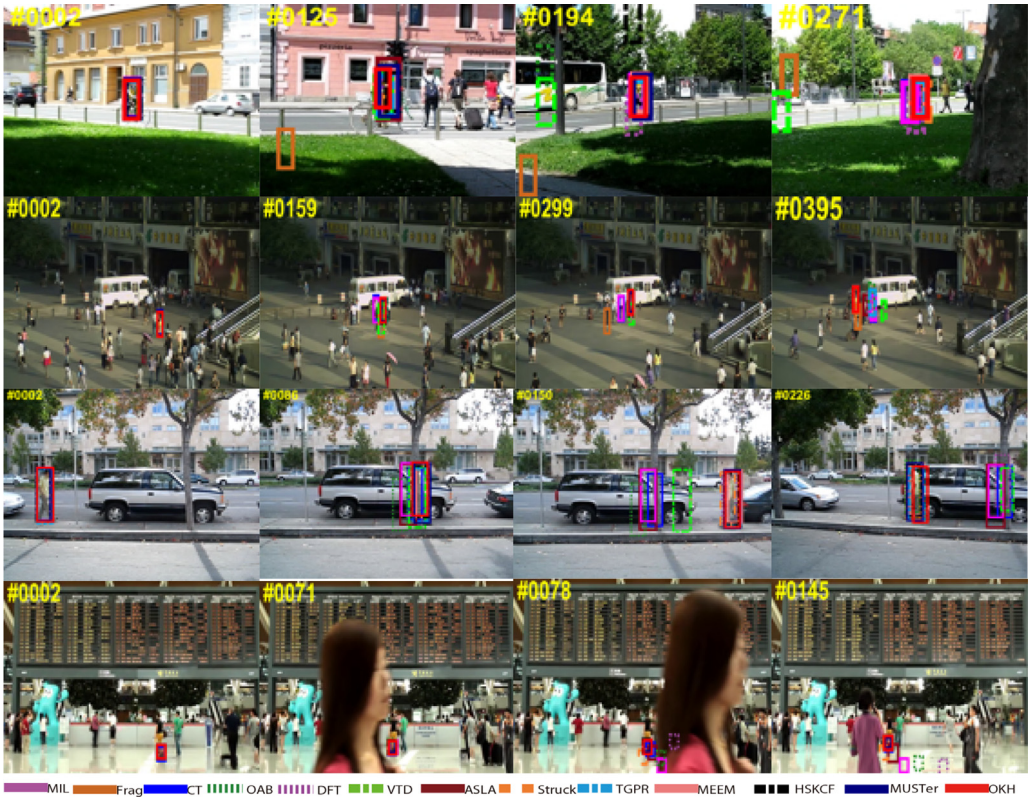

Fig. 7. Qualitative performance evaluation on cluttered background situation. The typical sequences in this figure are "*Bicycle*", "*Busstation*", "*David3*", and "*Suitcase*", respectively.

As for the In-plane-rotation/Out-plane-rotation challenges, the targets are difficult to track because of the significant rotations (e.g., hand slanting or shaking) which usually make the object appearance change. For tackling this kind of challenges, OKH, MEEM (Zhang et al., 2014a) and MUSTer (Hong et al., 2015) show competitive tracking performance, and are better than the others. For example, OKH, MEEM and MUSTer can track the target robustly, proved by the 313rd frame of "*Bolt*" and the 298th frame of "*Hurdle2*". One main reason is OKH and MEEM are all based color template encoding. In addition, OKH, MEEM and MUSTer all have the re-targeting ability when the appearance of object varying. To make a further observation from Table 1 and Table 2, OKH is better than MEEM. That is because the spatio-temporal saliency model to some degree can get rid of a large part of the background. Hence, when the diverse rotations occur, this tracker can resist the influence of the appearance change caused by rotation.

6.2.3. Occlusion

Occlusion is one of the most important challenges to be addressed in tracking. In general, the object in this situation is partially or fully occluded by other scenes which always disturb the consistency of the target appearance. Fig. 9 demonstrates the qualitative comparison of the competing trackers. In this work, a memory buffer for storing the positive templates is constructed, which can restrain the appearance degradation in online hashing, and the historical and the newest target information can be effectively preserved. Besides, owing to the long-term memory mechanism by point corresponding in MUSTer, our tracker and MUSTer show better ability for tackling occlusion. Specially, the tracking results in "*Motorbike*" sequence verify the superiority of OKH and MUSTer manifestly, taking the 537th frame for an example. However, OKH is the best in the competitors, shown by Table 1 and Table 2. Additionally, the patch-based color template encoding makes the proposed tracker in this work have strong robustness for occlusion handling.

6.2.4. Illumination variation

Because the imaging scenarios alter dynamically, the illumination condition of objects changes frequently because of the shadow disturbance or the strong light twinkling. The difficulty of the se-

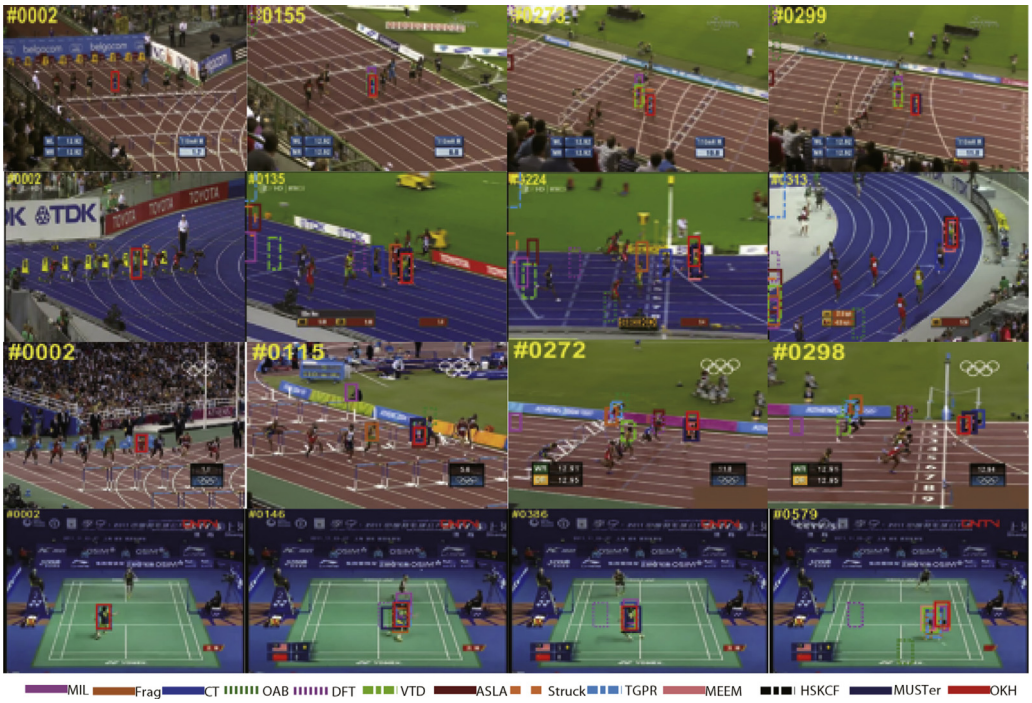

Fig. 8. Qualitative performance evaluation on deformation or in-plane-rotation/out-plane-rotation challenges. The typical sequences are "*Hurdle1*", "*Bolt*", "*Hurdle2*", and "*Badminton*", respectively.

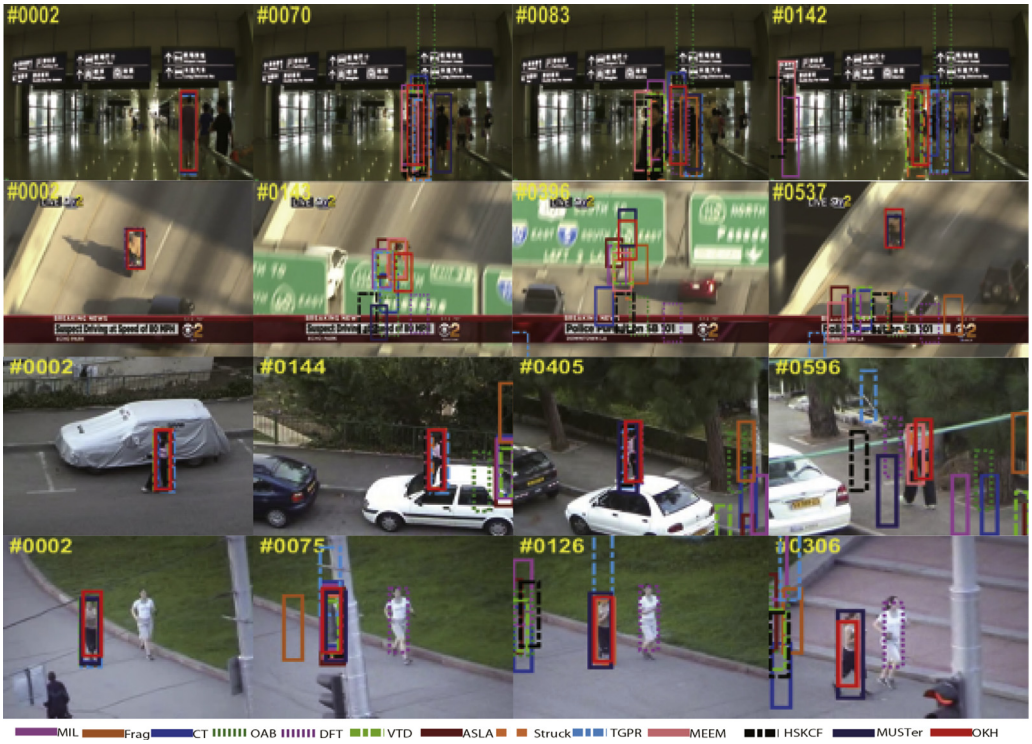

Fig. 9. Qualitative performance evaluation on occlusion situations. This figure shows tracking results of "*Airport*", "*Motorbike*", "*Woman*", and "*Jogging1*" sequences, respectively.

quences in this situation is how to boost the robustness of the object appearance. From Table 1, we can see that TGPR (Gao et al., 2014) performs the best in target locating, and MUSTer (Hong et al., 2015) matches the target region the best, shown by Table 2. OKH generates a comparative performance with the top ones in overall performance. However, all the trackers cannot tackle the "*Matrix*" successfully. The 78th frame and 84th frame in Fig. 10 of "*Matrix*" bore out this observation. That is because the current trackers are based on some spatial or temporal assumption, such as motion smoothness or appearance consistency. The human face in "*Matrix*" not only undergoes varying illumination, but also has dramatic movement, which makes the assumptions be not valid at all, even if for the spatio-temporal saliency in this work. This makes us ponder in future.

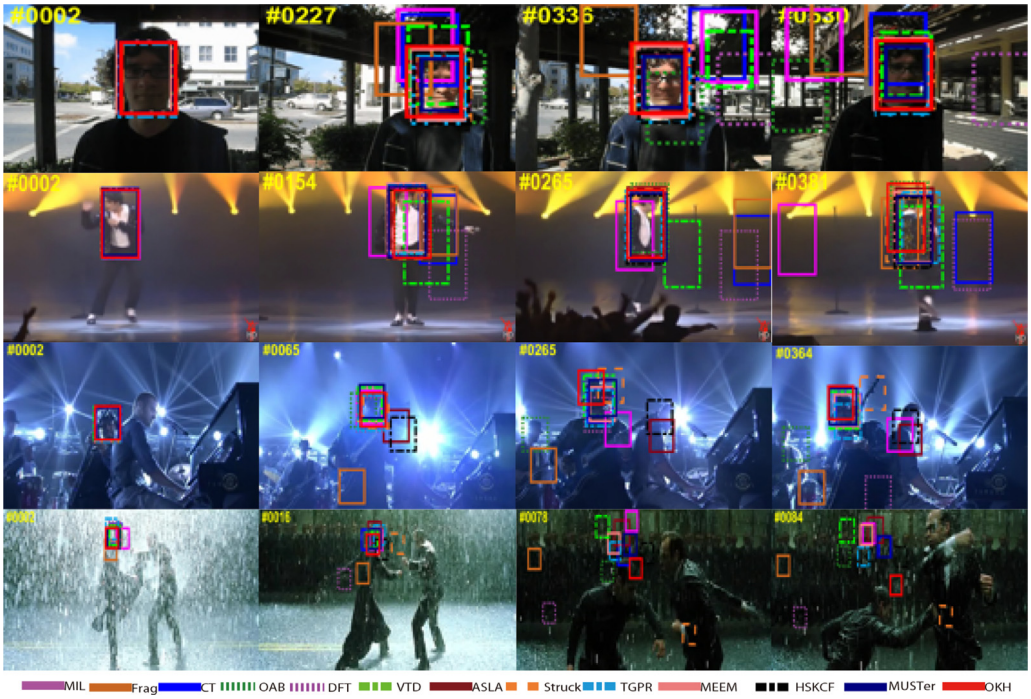

Fig. 10. Qualitative performance evaluation on illumination variation challenge. This figure demonstrates tracking results of "*Trellis*", "*Michaeljackson*", "*Shaking*", and "*Matrix*" sequences, respectively.

6.2.5. Fast motion/motion blur

The sequences in this situation are difficult because the distinctive texture or structure information of target is difficult to capture owing to the fast movement or motion blur. From Table 1 and Table 2, we can see that OKH, MEEM (Zhang et al., 2014a) and TGPR (Gao et al., 2014) outperform other trackers in localization, and OKH, MEEM (Zhang et al., 2014a) and MUSTer (Hong et al., 2015) match the target region better. However, MEEM and MUSTer drift away in the 110th frame in "*Plane*" sequence, and TGPR and MUSTer cannot track the "*fish*" from the 225th frame on in "*Fish*" sequence, seeing Fig. 11 for a demonstration. Hence, OKH performs the best. This benefits from the spatio-temporal saliency model in this work. If the target moves with a large displacement in adjacent frames, the spatio-temporal saliency model can guide the motion model to approximate the true target center in newly observed frame as close as possible.

From the above thorough analysis with extensive experiments, we can conclude that our tracker has better effectiveness and robustness when facing variously difficult challenges.

6.3. Discussions

6.3.1. Component investigation

As described before, this work contributes an online hashing tracker with a spatio-temporal saliency auxiliary, where a spatio-temporal saliency model (abbreviated as S), online pair-wise hashing and positive template memory buffer (abbreviated as T) are the central components. For a clearer evaluation, we take the tracker only with online pair-wise hashing component as a baseline and named as OKH\TS. Then, we add other components, the spatio-temporal saliency model (S) and the positive template memory buffer (T) to OKH\TS, and construct OKH\T and OKH\S, respectively. Specially, we denote the tracking method with all the three components as OKH. Fig. 12 demonstrates the precision rate and success rate curves on OKH-Color50 dataset. We can see that combining all the component generates the best performance, followed by OKH\S, OKH\T, and OKH\TS. It can be observed that the template buffer can improve the precision of target localization manifestly. Spatio-temporal saliency model is the most efficient component in tackling challenging attributes. Online pair-wise hashing can capture the main structure information. In summary, tracking with all the three components together is the best.

6.3.2. Analysis w.r.t. object categories

In this section, we present how the tracker performs for different object categories on OKH-Color50 dataset. Fig. 13 demonstrates the average precision rate and average success rate for different trackers w.r.t. distinct object categories. We can observe that MEEM (Zhang et al., 2014a) , MUSTer (Hong et al., 2015) and our OKH perform well in most of the categories. This is in consistent with the performance on different challenging attributes. Through observation in Fig. 13, *sportsman* and *animal* are the most challenging among 7 main object types. The main reason is that appearance deformation and shape variation occur frequently in *sportsman* category, and the motion of the *animal* is uncontrollable.

6.3.3. Difference with the deep learning models

We have recognized that deep learning models perform powerful in many computer vision tasks, including more recent works on tracking. In this subsection, we make a description to expound the difference between our work and the more recent deep learning models in tracking. The main goal of deep learning models is to obtain strong feature by large-scale data training or to find the intrinsic structure of data by designing complex neural networks. For visual tracking, the goal is similar to a large extent (Nam and Han, 2016; Tao et al., 2016). Specially, RNN is recently used to model the complex structure of object and its contextual background in tracking task (Cui et al., 2016). Our framework is different from deep learning models from three points: 1) We want to find an efficient and effective projective mechanism to map the high-dimensional feature into a lower dimensional one with a preservation of feature structure, not a new kind of feature; 2) Most of the deep learning models rely on the external large-scale dataset for feature training. Our framework aims to use pair-wise relationship

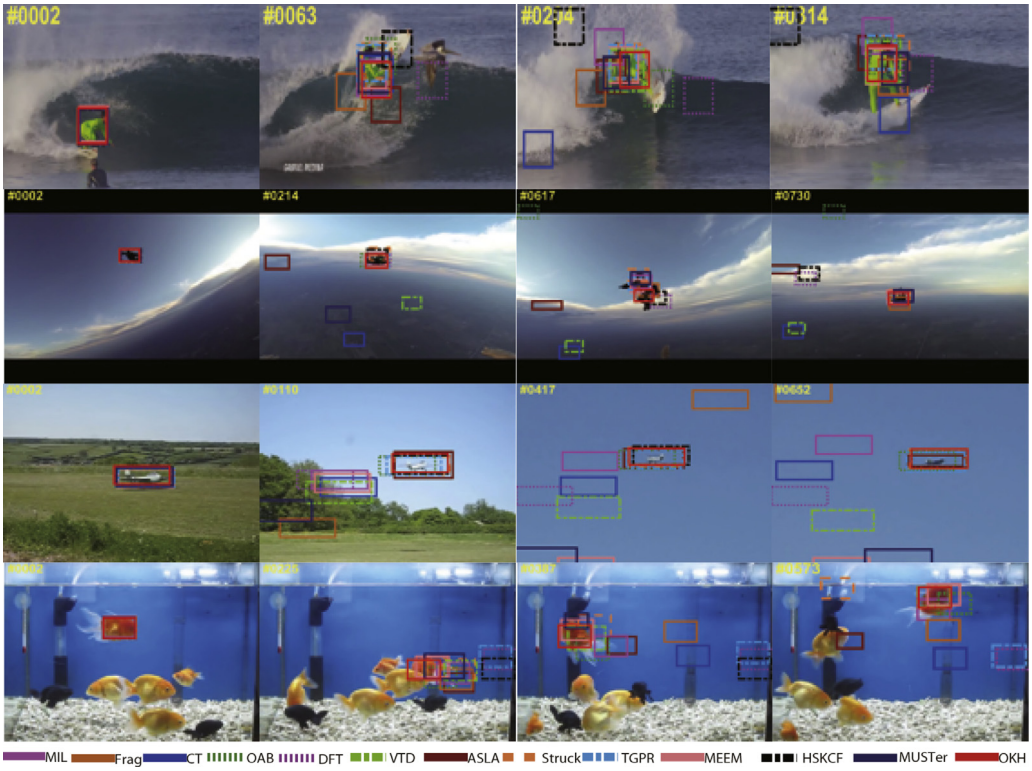

Fig. 11. Performance evaluation on fast motion situations. The sequences in this figure are "*Surf1*", "*Skyjumping*", "*Plane*", "*Fish*", respectively.

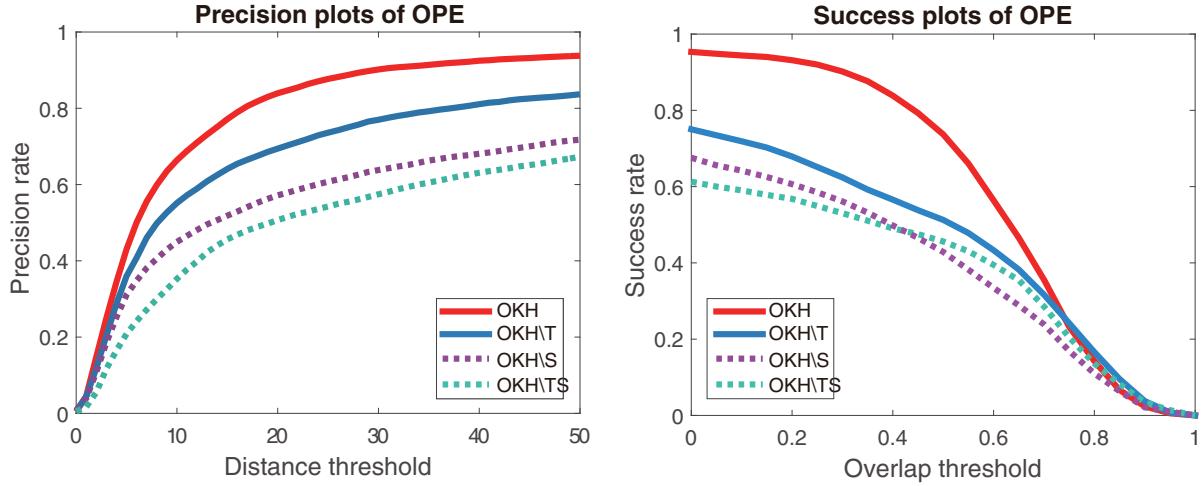

Fig. 12. Performance comparison with respect to the utilization of different components of the proposed tracker on all 50 the sequences.

of positive and negative templates with an online hashing manner, where only the first frame of video is labeled for training, by which the templates between frames can be efficiently matched; 3) This work integrates the saliency insight and template memory buffer to explicitly tackle the spatio-temporal influence caused by dynamic scenes. There is little parameter in this work, which is different from the large parameter space in the common deep models.

6.3.4. Failure situation demonstration

In this section, we give a discussion on some tracking failures. As shown in Fig. 14, there are some challenging sequences, in which all the existing trackers failed. Among them, the sequences in the left column demonstrate a long-term fully occlusion, the *suv* scales up rapidly in "CarScale" sequence, and the left-bottom sequence arises a fast motion accompanied by a fast scale shrinkage. Despite the fact that many efforts are devoted to tackle the scale variation and occlusion, we still need a further pondering. Making a general observation, the trackers in the literature tackle the occlusion with a predefined motion model, such as Gaussian, to restrain the template sampling. The failures for the long-term full occlusion is caused by the fact that trackers cannot re-detect or re-localize the target when it reappears at a location which is not in a predefined range. Therefore, for a further research for tracking, more appropriate sampling methods are needed. Besides, for the rapid scale variation, this work suggests objectness (Cheng et al., 2014) may be a promising alternative, which can provide many object candidates in each frame, and the best candidate might be selected by favorable optimization.

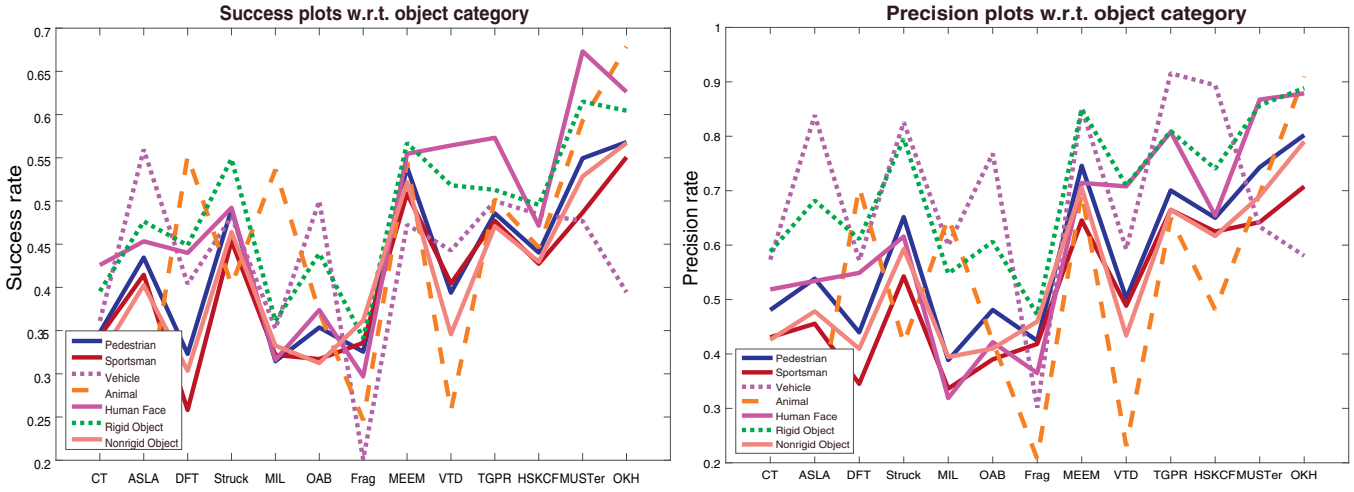

Fig. 13. Tracking performance w.r.t. object category.

Fig. 14. Typical failures on tracking. "*CarScale*" shows rapid scale variation; "*Suitcase*" demonstrates long-term fully occlusion; "*Matrix*" has extremely fast motion.

7. Conclusions

This work proposes a novel online hashing tracking method with a spatio-temporal saliency auxiliary. Through online hashing each pair of templates received sequentially, and the relationship of positive and negative samples can be persevered more effectively. Consequently, the tracking problem is formulated as an efficient binary code matching. The spatio-temporal saliency model makes the sampled candidates approximate the true target as much as possible. Additionally, the memory buffer, collecting the truly positive templates, can address the potential degradation of appearance model owing to the error accommodation in online hashing, and give a better localization. Extensive experiments proved that the proposed tracker can deal with different challenges better than the other state-of-the-arts. In this work, only the Lab color space is utilized for template encoding. Hence, future works concentrate on enriching the image representation by utilizing more features as well as powerful template sampling. In addition, because of the powerful ability of features obtained by deep learning models. This also may be investigated in our future work.

Acknowledgments

This work was supported by the National Key R&D Program Project under grant 2016YFB1001004, and also was supported by National Natural Science Foundation of China under grants 61603057, 61379094, 41601437 and 61503235.

References

Adam, A., Rivlin, E., Shimshoni, I., 2006. Robust fragments-based tracking using the integral histogram. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 798–805.
Babenko, B., Yang, M., Belongie, S., 2011. Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. 33 (8), 1619–1632.
Babenko, B., Yang, M.-H., Belongie, S., 2011. Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. 33 (8), 1619–1632.
Bai, Q., Wu, Z., Sclaroff, S., Betke, M., Monnier, C., 2013. Randomized ensemble tracking. In: Proc. IEEE Int'l. Conf. Computer Vision, pp. 2040–2047.
Bao, C., Wu, Y., Ling, H., Ji, H., 2012. Real-time robust ℓ_1 tracker using accelerated proximal gradient approach. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1830–1837.
Borji, A., Cheng, M.-M., Jiang, H., Li, J., 2014. Salient object detection: a survey. arXiv preprint arXiv:1411.5878.
Cakir, F., Sclaroff, S., 2015. Adaptive hashing for fast similarity search. In: IEEE Int'l. Conf. Computer Vision, pp. 1044–1052.
Chang, S.F., 2012. Supervised hashing with kernels. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 2074–2081.
Charikar, M.S., 2002. Similarity estimation techniques from rounding algorithms. In: Proc. ACM Symposium on Theory of Computing, pp. 380–388.
Chen, L., Xu, D., Tsang, I.W., Li, X., 2014. Spectral embedded hashing for scalable image retrieval. IEEE Trans. Cybern. 44 (7), 1180–1190.
Cheng, M.M., Zhang, Z., Lin, W.Y., Torr, P., 2014. Bing: Binarized normed gradients for objectness estimation at 300fps. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 3286–3293.
Chum, O., Philbin, J., Zisserman, A., 2008. Near duplicate image detection: Min-hash and tf-idf weighting, pp. 812–815.
Ciesielski, K.C., Strand, R., Malmberg, F., Saha, P.K., 2014. Efficient algorithm for finding the exact minimum barrier distance. Comput. Vision Image Understanding 123 (2), 53–64.
Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y., 2006. Online passive-aggressive algorithms. J. Mach. Learn. Res. 7 (3), 551–585.
Cui, Z., Xiao, S., Feng, J., Yan, S., 2016. Recurrently target-attending tracking. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1449–1458.
Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.S., 2004. Locality-sensitive hashing scheme based on p-stable distributions. In: Proc. Twentieth Symposium on Computational Geometry, pp. 253–262.
Du, D., Zhang, L., Lu, H., Mei, X., Li, X., 2015. Discriminative hash tracking with group sparsity. IEEE Trans. Cybernetics 46 (8), 1914–1925.
Duffner, S., Garcia, C., 2013. Pixeltrack: a fast adaptive algorithm for tracking non-rigid objects. In: Proc. IEEE Int'l. Conf. Computer Vision, pp. 2480–2487.
Finley, T., Joachims, T., 2008. Training structural SVMs when exact inference is intractable. In: Proc. Int'l. conf. Machine learning, pp. 304–311.
Gao, J., Ling, H., Hu, W., Xing, J., 2014. Transfer learning based visual tracking with gaussian processes regression. In: Proc. Eur. Conf. Computer Vision, pp. 188–203.
Grabner, H., Bischof, H., 2006. Online boosting and vision. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 260–267.
Hare, S., Saffari, A., Torr, P.H.S., 2011. Struck: Structured output tracking with kernels. In: Proc. Int'l. Conf. Computer Vision, pp. 263–270.
He, R., Cai, Y., Tan, T., Davis, L., 2015. Learning predictable binary codes for face indexing. Pattern Recognit 48 (10), 3160–3168.

Henriques, J.F., Caseiro, R., Martins, P., Batista, J., 2015. High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. 37 (3), 583–596.
Hong, Z., Chen, Z., Wang, C., Mei, X., 2015. Multi-store tracker (muster): a cognitive psychology inspired approach to object tracking. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 749–758.
Huang, L.-K., Yang, Q., Zheng, W.-S., 2013. Online hashing. In: Proc. Int'l. Joint Conf. Artificial Intelligence, pp. 1422–1428.
Jia, X., Lu, H., Yang, M.H., 2012. Visual tracking via adaptive structural local sparse appearance model. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1822–1829.
Kwon, J., Lee, H.S., Park, F.C., Lee, K.M., 2014. A geometric particle filter for template-based visual tracking. IEEE Trans. Pattern Anal. Mach. Intell. 36 (4), 625–643.
Kwon, J., Lee, K.M., 2010. Visual tracking decomposition. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1269–1276.
Lan, X., Ma, A.J., Yuen, P.C., 2014. Multi-cue visual tracking using robust feature-level fusion based on joint sparse representation. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1194–1201.
Leng, C., Wu, J., Cheng, J., Bai, X., Lu, H., 2015. Online sketching hashing. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 2503–2511.
Li, A., Lin, M., Wu, Y., Yang, M., 2016. Nus-pro: a new visual tracking challenge. IEEE Trans. Pattern Anal. Mach. Intell. 38 (2), 335–349.
Li, W., Wang, P., Qiao, H., 2014. Visual tracking via saliency weighted sparse coding appearance model. In: Proc. Int'l. Conf. Pattern Recognition, pp. 4092–4097.
Li, X., Dick, A.R., Shen, C., van den Hengel, A., Wang, H., 2013. Incremental learning of 3D-DCT compact representations for robust visual tracking. IEEE Trans. Pattern Anal. Mach. Intell. 35 (4), 863–881.
Li, X., Lu, H., Zhang, L., Ruan, X., Yang, M.H., 2013. Saliency detection via dense and sparse reconstruction. In: Proc. IEEE Int'l. Conf. Computer Vision, pp. 2976–2983.
Li, X., Shen, C., Dick, A.R., van den Hengel, A., 2013. Learning compact binary codes for visual tracking. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 2419–2426.
Liang, P., Blasch, E., Ling, H., 2015. Encoding color information for visual tracking: algorithms and bechmark. IEEE Trans. Image Processing 24 (12), 5630–5643.
Liu, Z., Li, H., Zhou, W., Zhao, R., Tian, Q., 2014. Contextual hashing for large-scale image search.. IEEE Trans. Image Processing 23 (4), 1606–1614.
Ma, C., Liu, C., 2015. Two dimensional hashing for visual tracking. Comput. Vision Image Understanding 135, 83–94.
Mahadevan, V., Vasconcelos, N., 2013. Biologically inspired object tracking using center-surround saliency mechanisms. IEEE Trans. Pattern Anal. Mach. Intell. 35 (3), 541–554.
Mei, X., Ling, H., 2009. Robust visual tracking using ℓ_1 minimization. In: Proc. IEEE Int'l. Conf. Computer Vision, pp. 1436–1443.
Nam, H., Han, B., 2016. Learning multi-domain convolutional neural networks for visual tracking. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 4293–4302.
Ross, D.A., Lim, J., Lin, R.-S., Yang, M.-H., 2008. Incremental learning for robust visual tracking. Int. J. Comput. Vis. 77 (1–3), 125–141.
Sevilla-Lara, L., 2012. Distribution fields for tracking. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1910–1917.
Su, Y., Zhao, Q., Zhao, L., Gu, D., 2014. Abrupt motion tracking using a visual saliency embedded particle filter. Pattern Recognit. 47 (5), 1826–1834.
Tao, R., Gavves, E., Smeulders, A.W.M., 2016. Siamese instance search for tracking. CoRR abs/1605.05863.
Wang, D., Lu, H., Yang, M.-H., 2013. Online object tracking with sparse prototypes. IEEE Trans. Image Processing 22 (1), 314–325.
Wang, L., Liu, T., Wang, G., Chan, K.L., Yang, Q., 2015. Video tracking using learned hierarchical features. IEEE Trans. Image Processing 24 (4), 1424–1435.
Wang, Q., Fang, J., Yuan, Y., 2014. Multi-cue based tracking. Neurocomuting 131, 227–236.
Weiss, Y., Torralba, A., Fergus, R., 2008. Spectral hashing. Proc. Adv. Neural Inf. Process. Syst. 282 (3), 1753–1760.
Wen, L., Cai, Z., Lei, Z., Yi, D., Li, S.Z., 2014. Robust online learned spatio-temporal context model for visual tracking. IEEE Trans. Image Processing 23 (2), 785–796.
Wu, C., Zhu, J., Cai, D., Chen, C., Bu, J., 2013. Semi-supervised nonlinear hashing using bootstrap sequential projection learning. IEEE Trans. Knowl. Data Eng. 25 (6), 1380–1393.
Wu, Y., Lim, J., Yang, M.-H., 2013. Online object tracking: A benchmark. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 2411–2418.
Yan, Q., Shi, J., Xu, L., Jia, J., 2014. Hierarchical saliency detection on extended CSSD. IEEE Trans. Pattern Analy. Mach. Intell. 9 (4), 717–729.
Yang, F., Lu, H., Yang, M.-H., 2014. Robust superpixel tracking. IEEE Trans. Image Processing 23 (4), 1639–1651.
Yuan, Y., Fang, J., Wang, Q., 2014. Robust superpixel tracking via depth fusion. IEEE Trans. Circuits Syst. Video Technol. 24 (1), 15–26.
Zhang, J., Ma, S., Sclaroff, S., 2014. MEEM: robust tracking via multiple experts using entropy minimization. In: Proc. Eur. Conf. Computer Vision, pp. 188–203.
Zhang, K., Zhang, L., Liu, Q., Zhang, D., Yang, M.-H., 2014. Fast visual tracking via dense spatio-temporal context learning. In: Proc. Eur. Conf. Computer Vision, pp. 127–141.
Zhang, K., Zhang, L., Yang, M.H., 2012. Real-time compressive tracking. In: Eur. Conf. Computer Vision, pp. 864–877.
Zhang, L., van der Maaten, L., 2014. Preserving structure in model-free tracking. IEEE Trans. Pattern Anal. Mach. Intell. 36 (4), 756–769.
Zhang, T., Liu, S., Ahuja, N., Yang, M.-H., Ghanem, B., 2015. Robust visual tracking via consistent low-rank sparse learning. Int. J. Comput. Vis. 111 (2), 171–190.
Zhang, X., Hu, W., Bao, H., Maybank, S.J., 2013. Robust head tracking based on multiple cues fusion in the Kernel–Bayesian framework. IEEE Trans. Circuits Syst. Video Technol. 23 (7), 1197–1208.
Zhong, W., Lu, H., Yang, M.-H., 2014. Robust object tracking via sparse collaborative appearance model. IEEE Trans. Image Processing 23 (5), 2356–2368.
Zhou, H., Yuan, Y., Shi, C., 2009. Object tracking using SIFT features and mean shift. Comput. Vision Image Understanding 113 (1), 345–352.
Zhu, G., Wang, J., Zhao, C., Lu, H., 2014. Part context learning for visual tracking. In: Proc. British Machine Vision Conference, pp. 26–33.

Jianwu Fang received the Ph.D. degree in SIP (signal and information processing) from the Center for Optical Imagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China in 2015. He is currently a lecturer in the School of Electronic & Control Engineering, Chang'an University, Xi'an, China, and is also a postdoctoral researcher in the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China. His research interests include computer vision and pattern recognition.

Hongke Xu received his PhD degree from Changan University, China, in 2006. He is currently a full professor of School of Electronic & Control Engineering, Chang'an University, Xi'an, China. His research interests include signal processing and traffic intelligent control engineering.

Qi Wang received the B.E. degree in automation and Ph.D. degree in pattern recognition and intelligent system from the University of Science and Technology of China, Hefei, China, in 2005 and 2010 respectively. He is currently an associate professor with the School of Computer Science and the Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision and pattern recognition.

Tianjun Wu received the B.S. degree in mathematics and information science, the M.S. degree in applied mathematics from Changan University, Xi'an, China, in 2009 and 2012 respectively; and the Ph.D. degree in cartography and geographical information system from the State Key Laboratory of Remote Sensing Sciences, Institute of Remote Sensing and Digital Earth (RADI), Chinese Academy of Sciences, Beijing, China, in 2015. He is currently a lecturer in the Department of Mathematics and Information Science, College of Science, Chang'an University. His research interests include remote sensing image processing and information analysis.