Neural Computing and Applications manuscript No.
(will be inserted by the editor)

Object tracking in the presence of shaking motions

Manna Dai · Shuying Cheng · Xiangjian He · Dadong Wang

Received: date / Accepted: date

Abstract Visual tracking can be particularly interpreted as a process of searching for targets and optimizing the searching. In this paper, we present a novel tracker framework for tracking shaking targets. We formulate the underlying geometrical relevance between a search scope and a target displacement. A uniform sampling among the search scopes is implemented by sliding windows. To alleviate any possible redundant matching, we propose a double-template structure comprising of initial and previous tracking results. The element-wise similarities between a template and its candidates are calculated by jointly using kernel functions which provide a better outlier rejection property. The STC algorithm is used to improve the tracking results by maximizing a confidence map incorporating temporal and spatial context cues about the tracked targets. For better adaptation to appearance variations, we employ a linear interpolation to update the context prior probability of the STC method. Both qualitative and quantitative evaluations are performed on all sequences that contain shaking motions and are selected from the OTB-50 challenging benchmark. The proposed approach is compared with and outperforms 12 state-of-the-art tracking methods on the selected sequences while running on MATLAB without code optimization. We have also performed further experiments on the whole OTB-50 and VOT 2015 datasets. Although the most of sequences in these two datasets do not contain motion blur that this paper is focusing on, the results of our method are still favorable compared with all of the state-of-the-art approaches.

Keywords shaking targets · uniform sampling · kernel · temporal and spatial context

M. Dai
Fuzhou University, China
University of Technology Sydney, Australia
Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia
E-mail: Manna.Dai@student.uts.edu.au

S. Cheng (Corresponding author)
Fuzhou University, China
Jiangsu Collaborative Innovation Center of Photovolatic Science and Engineering, China
E-mail: sycheng@fzu.edu.cn

X. He (Corresponding author)
University of Technology Sydney, Australia
E-mail: Xiangjian.He@uts.edu.au

D. Wang
Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia
E-mail: Dadong.Wang@data61.csiro.au

1 Introduction

Visual object tracking tries to locate a target in an image sequence. It has been a long standing research topic due to the proliferation of applications such as security warning, medical image analysis, sport analysis and so on [43]. However, visual tracking is challenging because of deformation, fast motion, motion blur and background clutters. Although a significant progress has been made to overcome these challenges, most state-of-the-art trackers fail in the presence of shaking motion.

The existing visual tracking approaches can be categorized into generative [23,24,32,5,18,34,1] and discriminative [48,16,46,13,2,20,49] methods. The generative tracking methods search for image regions that are most similar to a given template, while discriminative methods aim at differentiating a target from its background.

Generative tracking methods utilize varieties of Distance Measures to select the most similar patch. Euclidean Distance [9], the most easily calculated distance, is sensitive to image deformation due to the neglect of image extensibility. Mahalanobis Distance [28] computes the similarity between

two unknown samples by introducing every association among the features. However, Mahalanobis Distance calculation is not reliable because the inverse covariance matrix may not exist. Bhattacharyya Distance [4] calculates the difference between the histograms of two images. The accuracy of this distance depends on the number of bins (or partitions) in the histograms. When the number of partitions is small, the distance precision decreases due to overestimation of overlap regions. On the other hand, when the number of partitions is big, precision increases due to the neglect of real overlap regions. Hamming Distance [14] between two binary strings of equal length measures the minimum number of substitutions required to change one string into the other. Because Hamming Distance is simple and fast, it is used as the distance measure in our approach for implementing a coarse matching. More details will be introduced in Section 3.1.3 and Section 4.3.3.

There are many ways to perform a tracking task. Zhang et al. [47] proposed a simple yet fast algorithm called spatio-temporal context tracker (STC), which exploited the dense spatio-temporal context in a Bayesian framework. It uses low-level features from a target and its surrounding regions to model a statistical correlation. Circulant structure tracker (CSK) [17] is a tracking-by-detection approach. It utilizes a dense sampling strategy to achieve a high speed. The adaptive color tracker (ACT) [12] extends the CSK tracker by changing the single-channel gray feature used in CSK to multi-channel color features [38]. This tracker provides a good balance between the photometric invariance and the discriminative power during an object recognition procedure.

Some trackers are based on the Kalman filter framework. Erik et al. [7] reported that Kalman filter can be used to tolerate small occlusions by prediction and correction work phases. Since the complex dynamic trajectories(changes of acceleration) cannot be built as lineal models, they applied the extended Kalman filter (EKF) [31] to model this question as nonlinear equations. Julier et al. first proposed the Unscented Kalman filter [19,37] to address nonlinear state estimation in the context of control theory. The experiments show that UKF is superior to EKF in terms of theory and practical applications. Hence, Peihua et al. [26] implemented the UKF in the visual contour tracking framework. However, UKF algorithm can not be applied to multimodal distribution due to its unimodal distribution.

Many other visual tracking methods, such as L1 tracker (L1) [29], L1 tracker using accelerated proximal gradient (L1-APG) [3], compressive tracker (CT) [48] and a tracker using online informative feature selection (OIFS) [35], cast tracking as a sparse approximation problem. In the L1 tracker, target candidates are sparsely represented in the space spanned by target templates and trivial templates. The key of obtaining the sparsity is regarded as solving an L1-regularized least square problem under a particle filter framework. However, the highly computational complexity of this tracker limits its applications in real-time scenarios. L1-APG further extends the L1 tracker by developing a new and L1-norm related minimization which introduces an L2 norm regularization on the coefficients connected to trivial templates. The CT tracker also utilizes a very sparse matrix to efficiently extract the features for an appearance model. They formulate the tracking task as a binary classification by a Naive-Bayesian classifier with online update under a compressed domain. The OIFS tracker exploits random projections to train a classifier. They novelly introduce spatial layout information into projected features and thereby the projections contain the structure information of a target.

Another class of tracking approaches are based on template matching. The similarities between templates and candidates of a target are used to determine the confidence values of candidates matching the target. The conventional template techniques can be divided into one-template and multi-template approaches [27]. Recently, several studies [27,30, 21,50,46,6] have reported that multi-template approaches often achieve greater precision but have more complex computation than single-template approaches.

All of the above mentioned trackers may fail in the presence of shaking motion due to the following two main issues. Firstly, sometimes the whole tracked target may not be found in any of the candidate patches because it has moved out from the candidate searching scope due to fast and big shaking motion. Secondly, shaking movement may result in blurred images. Trackers can be easily confused by blurred boundaries between a foreground and its background.

To handle the above issues, we propose a robust tracking algorithm which implements a novel sampling strategy, a novel matching, a novel template method and a detector exploiting spatio-temporal contextual cues. The experimental results show that our tracker outperforms other state-of-the-art trackers.

The proposed tracker can be divided into two dominating parts. The first part is a generative tracker where a roughly yet fast search method is conducted via an improved sliding window. The second part is to further meliorate the localization by an improved STC tracker. Five main contributions of this paper are listed as follows.

- A novel tracking method for shaking motions (SMT) is proposed.
- A uniform sampling is formulated based on a sliding window that is adjustable by the previous moving displacements. This sampling method ensures that the matching processing goes through a full search scope of interest.
- The targets found in the initial (i.e., the first) and the previous frames are merged to form a novel double-template

structure. This template preserves the historic and latest features regarding the targets.
- A Gaussian kernel and a Uniform kernel are employed to reduce the computations to linear order. This leads to a very efficient and robust tracking algorithm.
- The conventional STC discovers both temporal and spatial relevance so that it is insensitive to appearance variations. Therefore, in order to better reflect the appearance changes, STC is exploited and its algorithm of context prior probability is upgraded in this paper for improving tracking precision.

The paper is organized as follows. Section 2 introduces some preliminary methods related to our work for immediate reference. In Section 3, we provide detailed information on the proposed approach. Section 4 presents qualitative and quantitative comparisons with twelve state-of-the-art approaches. At last, some concluding remarks are demonstrated in Section 5.

2 Review of STC tracker

STC has been shown to have a simple yet effective performance, so we integrate the STC tracker [47] into our approach.

We provide a brief overview of STC as follows.

The STC tracker formulates the tracking problem at frame t as a confidence map, which is modeled by a Bayesian framework

$$\begin{aligned} m_t(\mathbf{x}) &= P_t(\mathbf{x}|o) \\ &= \textstyle\sum_{\mathbf{c}(\mathbf{z})\in X_t^c} P_t(\mathbf{x},\mathbf{c}(\mathbf{z})|o) \\ &= \textstyle\sum_{\mathbf{c}(\mathbf{z})\in X_t^c} P_t(\mathbf{x}|\mathbf{c}(\mathbf{z}),o)P_t(\mathbf{c}(\mathbf{z})|o), \end{aligned} \tag{1}$$

$$X_t^c = \{\mathbf{c}(\mathbf{z}) = (I_t(\mathbf{z}),\mathbf{z})|\mathbf{z}\in\Omega_c(\mathbf{x}_t^*)\}, \tag{2}$$

where o denotes the object, X_t^c represents the feature set at frame t, $I_t(\mathbf{z})$ represents the image intensity treated with the Zero Mean Normalization (transforming the average value of function to nought) at location $\mathbf{z}$ at frame t, and $\Omega_c(\mathbf{x}_t^*)$ indicates the neighborhood of location $\mathbf{x}_t^*$ at frame t.

In Eq.1,

$$P_t(\mathbf{x}|\mathbf{c}(\mathbf{z}),o) = h_t^{sc}(\mathbf{x}-\mathbf{z}), \tag{3}$$

where $h_t^{sc}(\mathbf{x}-\mathbf{z})$ reveals the spatial context relationship of the object location $\mathbf{x}$ and its local context location $\mathbf{z}$ at frame t; and $P_t(\mathbf{c}(\mathbf{z})|o)$ represents the context prior probability at frame t, which is defined as

$$P_t(\mathbf{c}(\mathbf{z})|o) = I_t(\mathbf{z})w_{\sigma_t}(\mathbf{z}-\mathbf{x}_t^*). \tag{4}$$

Here, $w_{\sigma_t}(\mathbf{z}-\mathbf{x}_t^*)$ is a focus of attention function modeled as a weighted function at frame t:

$$w_{\sigma_t}(\mathbf{z}-\mathbf{x}_t^*) = a\cdot e^{-\frac{|\mathbf{z}-\mathbf{x}_t^*|^2}{\sigma_t^2}}, \tag{5}$$

where a represents a normalization constant obtained by a Hamming window, and σ_t denotes a scale parameter. The confidence map of an object location in Eq.1 is defined as

$$\begin{aligned} m_t(\mathbf{x}) &= b\cdot e^{-\left|-\frac{\mathbf{x}-\mathbf{x}_t^*}{\alpha}\right|^\beta} \\ &= \textstyle\sum_{\mathbf{z}\in\Omega_c(\mathbf{x}_t^*)} h_t^{sc}(\mathbf{x}-\mathbf{z})I_t(\mathbf{z})w_{\sigma_t}(\mathbf{z}-\mathbf{x}_t^*) \\ &= h_t^{sc}(\mathbf{x})\textstyle\bigotimes(I_t(\mathbf{x})w_{\sigma_t}(\mathbf{x}-\mathbf{x}_t^*)), \end{aligned} \tag{6}$$

where b represents a normalization constant, α and β are a scale parameter and a shape parameter respectively, and $\bigotimes$ is the convolution operator.

The calculation is converted from the time domain to the frequency domain. Therefore, Eq.6 can be rewritten as

$$\mathscr{F}\left(b\cdot e^{-\left|-\frac{\mathbf{x}-\mathbf{x}_t^*}{\alpha}\right|^\beta}\right) = \mathscr{F}(h_t^{sc}(\mathbf{x}))\bigodot\mathscr{F}(I_t(\mathbf{x})w_{\sigma_t}(\mathbf{x}-\mathbf{x}_t^*)), \tag{7}$$

where $\mathscr{F}(\cdot)$ is the Fast Fourier Transform (FFT) function and $\bigodot$ denotes the dot product. The spatial context at frame t is defined as

$$h_t^{sc}(\mathbf{x}) = \mathscr{F}^{-1}\left(\frac{\mathscr{F}\left(b\cdot e^{-\left|-\frac{\mathbf{x}-\mathbf{x}_t^*}{\alpha}\right|^\beta}\right)}{\mathscr{F}(I_t(\mathbf{x})w_{\sigma_t}(\mathbf{x}-\mathbf{x}_t^*))}\right), \tag{8}$$

where $\mathscr{F}^{-1}(\cdot)$ is the inverse FFT function. Then, a spatio-temporal context model at frame t can be defined as

$$H_t^{stc}(\mathbf{x}) = \begin{cases} h_1^{sc}(\mathbf{x}), & t=1 \\ (1-\rho)\cdot H_{t-1}^{stc}(\mathbf{x}) + \rho\cdot h_{t-1}^{sc}(\mathbf{x}), & 1<t\le T, \end{cases} \tag{9}$$

where T is the total number of frames, ρ is a learning parameter. Then, by replacing $h_t^{sc}(\mathbf{x})$ with $H_t^{stc}(\mathbf{x})$ in Eq.6, the confidence map at frame t is defined by

$$m_t(\mathbf{x}) = H_t^{stc}(\mathbf{x})\textstyle\bigotimes(I_t(\mathbf{x})w_{\sigma_t}(\mathbf{x}-\mathbf{x}_t^*)). \tag{10}$$

The object location $\mathbf{x}_t^{**}$ is selected to be the location of the ground truth when $t=1$ and is updated by maximizing the new confidence map [47] at $t>1$:

$$\mathbf{x}_t^{**} = \arg\max_{\mathbf{x}\in\Omega_c(\mathbf{x}_t^{**})} m_t(\mathbf{x}). \tag{11}$$

For more details about STC, one may refer to [47].

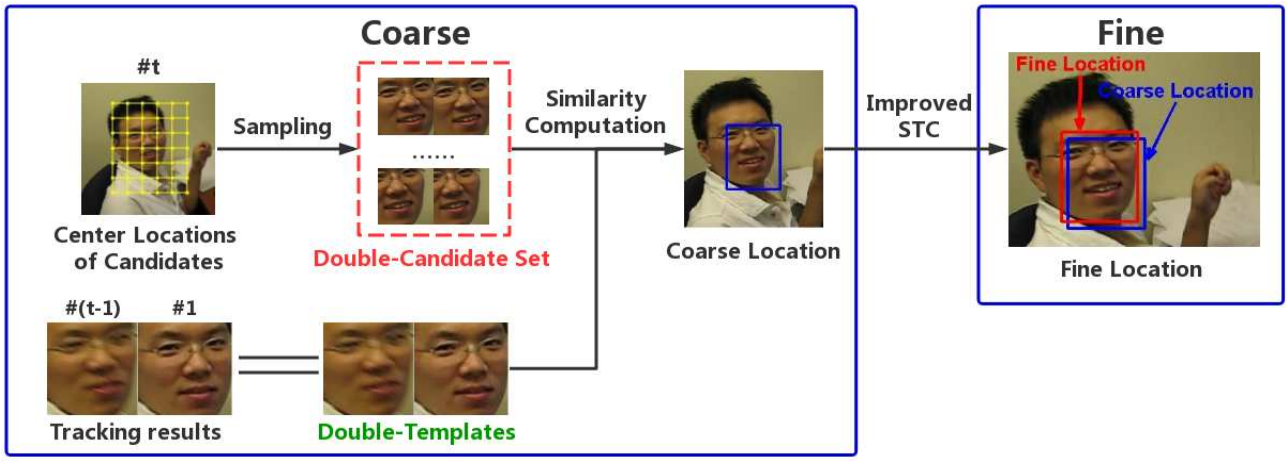

Figure 1 Basic flow of our tracking algorithm at frame t. Candidates are sampled, and the similarities between double-candidates and the double-templates are used to infer the coarse location. Then, the STC detection modifies the coarse location and gets the fine tracked target.

3 Proposed tracking algorithm

The motivation of our method is to track the target in normal videos as well as other challenging videos containing motion blurs, occlusions, fast motions and so on. The overall framework of the proposed algorithm is summarized in Figure 1. Compared with the conventional STC tracker, our tracking process is divided into *Coarse Processing* (Section 3.1) and *Fine Processing* (Section 3.2). The proposed similarity measure and the spatio-temporal context are the main work phases in coarse and fine processing, respectively. In the coarse processing, the result is obtained by the proposed similarity measure based on two kernel mappings, which reduce the impact of motion blur. Extensive experiments showed that this similarity measure has robustness in blurred or clear images. Then, in fine processing, an improved STC method is proposed and used to modify the coarse result by utilizing the spatio-temporal context model containing the relationship information between the tracked target and the information surrounding the target. The application of spatio-temporal context model makes our tracker insensitive to appearance variations, including motion blurs and other factors. Hence, our tracker can handle the problem regarding blurs and other outliers/noise. Besides that, the adaptive sampling strategy in Section 3.1.1 is designed for dealing with the fast motion, since our method is based on the historical motion tendency of a target. The proposed double-template strategy and spatio-temporal context have capability to recover the tracked target, due to preservation of initial image information and background information, respectively. The experimental results can be found in Section 4.4.2 to validate our method. All parameter values used in our approach are determined by experiments. The details of our work including parameter settings are elaborated in the following subsections.

3.1 Coarse processing

Visual tracking can be interpreted as an optimization problem of template matching, and locating an object in a video sequence can be interpreted as maximizing the feature similarity between a template and a sampled candidate. In this section, our coarse matching is driven by an improved sampling strategy, a double-template metric and a similarity measure.

3.1.1 Adaptive sampling strategy

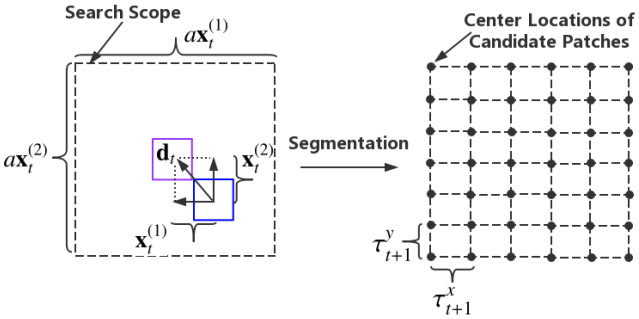

Figure 2 Search scope definition and acquirement of centre locations of candidate patches in the next frame (i.e., frame $t+1$). The blue box denotes the object in the previous frame (i.e., frame $t-1$) and the purple box denotes an object in the current frame (i.e., frame t). Then, we get the decomposition values $\mathbf{x}_t^{(1)}$ and $\mathbf{x}_t^{(2)}$ and lengthen them a times in order to obtain our search scope in the next frame. The right chart shows that the search scope is segmented into equal blocks and each point represents the centre of a candidate patch. The candidates are extracted by the sliding window.

The computational time of the matching-based tracker scales linearly with the sampled candidate number. We propose to use an adaptive sliding-window technique that uniformly samples useful candidates while drastically reducing

the number of candidate samples, thereby reduces the computation costs.

Figure 2 provides an illustrations of how the techniques construct the search scope according to the latest geometrical displacement and how the techniques extract candidate patches based on the adaptive sliding window in the next frame. We define the moving displacement at frame t as $\mathbf{d}_t = \mathbf{x}_t^{(1)} + \mathbf{x}_t^{(2)}$ and set its target centre location to be $L_t = (h,k)$. The search scope corresponds to an image region that is bounded by the last moving displacement of an target, and we propose to use the following equation to efficiently compute the corresponding search scope:

$$\Psi_t = \left\{ (x,y) | x \in \left[h - \left| \frac{a\mathbf{x}_t^{(1)}}{2} \right|, h + \left| \frac{a\mathbf{x}_t^{(1)}}{2} \right| \right], \right. \\ \left. y \in \left[k - \left| \frac{a\mathbf{x}_t^{(2)}}{2} \right|, k + \left| \frac{a\mathbf{x}_t^{(2)}}{2} \right| \right] \right\}, \tag{12}$$

for arbitrary real constants a.

Based on this, we propose a sliding-window searching on the search scope to return a minimal yet valid set of sampling candidates. Our sliding window is adaptive by the previous trajectory of the tracked object. The sliding step at frame t is given by

$$\tau_t^x = \left| \frac{a\mathbf{x}_t^{(1)}}{d} \right|, \qquad \tau_t^y = \left| \frac{a\mathbf{x}_t^{(2)}}{d} \right|, \tag{13}$$

where τ_t^x represents the horizontal step of the sliding window at frame t, and τ_t^y represents the vertical step of the sliding window at frame t, and d is an arbitrary real constant.

3.1.2 Double-template strategy

Target appearance varies over an image sequence due to illumination, camera and object geometry, and these appearance variances affect tracking accuracies. To achieve visual tracking that is robust to appearance changes, a novel double-template structure is proposed (shown in Figure 1). This structure stitches together the target patch in the original frame and the target patch found in the previous frame. The target patch in the initial frame contains non-outlier information, and the relevant outlier information can most likely be found in the target patch in the previous (latest) frame. Therefore, we use the information on only the target patches in the initial and latest frames to form the double-templates in this paper. Unlike the multi-template approaches that request an iterative process to update the templates, the proposed double-template approach updates only the template of the latest frame so the double-template approach is more efficient than the multi-template approaches. In Figure 1, each double-candidate is formed by two (the left and right) identical candidate patches. The matchings of the candidate patches in the current frame (i.e., frame t in Figure 1) to the templates in the initial and the previous frames (i.e., frames $t-1$ and 1 in Figure 1) are independent, so they can be processed in parallel to further improve the tracking speed.

3.1.3 Gaussian-Uniform joint similarity

In the tracking problem, the unexpected outliers (like motion blur) may change image pixel values, and hence make the matching processing unstable and lead the tracker to drift easily. The Least Square distance is commonly used to implement the matching between candidates and templates. However, outliers (e.g., the one shown in Figure 3) may affect the robustness of similarity calculation. According to reference [42], kernel mappings can effectively limit the impact of outliers and improves the robustness. Thus, our approach involves two kernel mappings and uses RGB values of image patches for template matching. Jian et al. [25] reported that Gaussian Kernel has outstanding capability of measuring the remote similarity between any two pictures in a mapping space. Outliers (like blurs or occlusions) can vary the local pixel values of a candidate image, and hence the candidate fails to match the template in the area where the outliers are. The Uniform Kernel, which is also known as the Boxcar Function, can filter out the effect of values where outliers are located. Thus, the matching will be based on a similarity algorithm using a Gaussian Kernel and a Uniform Kernel to handle the problem from outliers.

The Gaussian Kernel (radial-basis kernel) grows exponentially in a 2-dimensional space so it is sensitive to tiny element-wise (or pixel-wise) differences. In the following, we control the bandwidth of the Gaussian Kernel in order to enhance the differences and hence reduce the sensitivity.

Let $\mathbf{s} = [s_1, s_2, \cdots, s_n]^\top$ and $\mathbf{u} = [u_1, u_2, \cdots, u_n]^\top$ be a double-template vector and a double-candidate vector, respectively. Then, as shown in [36], the similarity representation between these two vectors using the Gaussian Kernel can be represented by

$$w_G(\mathbf{s}, \mathbf{u}) = \sum_{i=1}^{n} k_G(s_i, u_i), \tag{14}$$

where

$$k_G(s_i, u_i) = e^{-\frac{\|s_i - u_i\|^2}{\sigma^2}}. \tag{15}$$

In Eq.15, the bandwidth parameter is controlled by the Infinite Norm of the two vectors: $\sigma = \|\mathbf{s} - \mathbf{u}\|_\infty$ so that the results of $k_G(s_i, u_i)$ is controlled to be in the range $[e^{-1}, 1]$, to avoid the similarity (represented in Eq.14) between $\mathbf{s}$ and $\mathbf{u}$ becoming too close, otherwise.

The Uniform Kernel is used to measure the similarity between a template and a candidate by computing their overlap area with a selected threshold so as to reflect the global matching precision. The Uniform similarity $w_U(\mathbf{s}, \mathbf{u})$ is defined below, which is the sum of the Uniform Kernel values

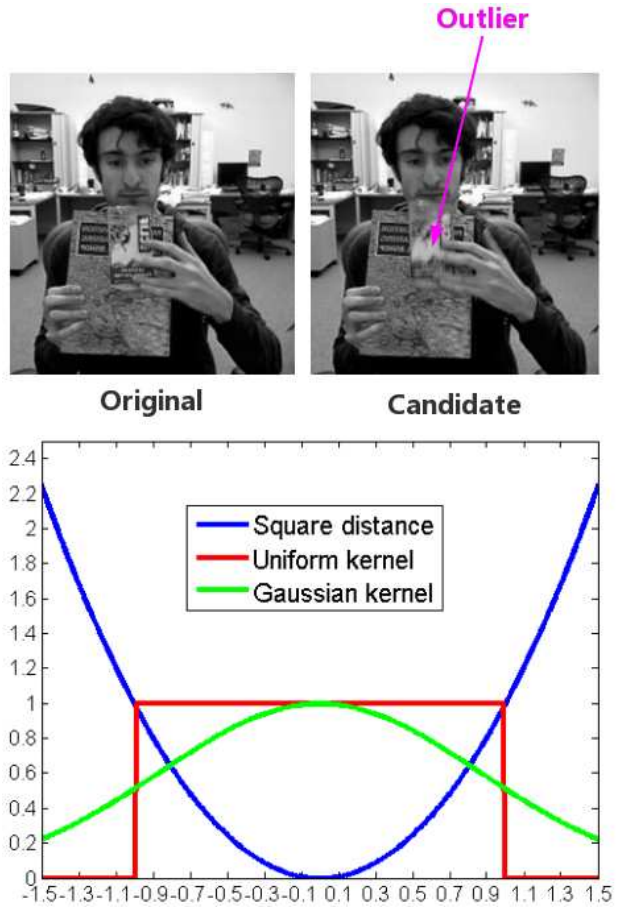

Figure 3 The effect of outliers and the illustration of different similarity measures. The sequence Clifbar contains the outlier caused by motion blur. Three similarity measures for comparisons are the Square distance measure, the Uniform kernel based measure and the Gaussian kernel based measure.

of individual points and defines an overlap score between $\mathbf{s}$ and $\mathbf{u}$.

$$w_U(\mathbf{s},\mathbf{u}) = \sum_{i=1}^{n} k_U(s_i,u_i), \tag{16}$$

with the Uniform Kernel defined by

$$k_U(s_i,u_i) = \begin{cases} \frac{1}{2\sqrt{\kappa}}, & \|s_i-u_i\|^2 \le \kappa \\ 0, & \|s_i-u_i\|^2 > \kappa, \end{cases} \tag{17}$$

where κ is called an arbitrary real threshold. The joint kernel based similarity is defined as

$$w_{joint}\left(\mathbf{s}_t,\mathbf{u}_t^i\right) = w_G\left(\mathbf{s}_t,\mathbf{u}_t^i\right) \times w_U\left(\mathbf{s}_t,\mathbf{u}_t^i\right) \tag{18}$$

At frame t, we denote a set of double-candidate vectors by $\mathbf{U}_t = \left\{\mathbf{u}_t^1,\mathbf{u}_t^2,\cdots,\mathbf{u}_t^m\right\}$ and a double-template vector by $\mathbf{s}_t$. Then, the tracked target at frame t is the candidate with the greatest Gaussian and Uniform similarity scores, i.e.,

$$\mathbf{u}_t^* = \arg\max_{\mathbf{u}_t^i} w_{joint}\left(\mathbf{s}_t,\mathbf{u}_t^i\right), 1 \le i \le m. \tag{19}$$

The centre of $\mathbf{u}_t^*$ is denoted by $\mathbf{x}_t^*$ $(t = 1,2,\cdots,T)$.

3.2 Fine processing

Assume that the target location in the first frame is initialized by some object detection algorithms. The emphasis of the *Fine processing* is laid on enhancement of reliability and accuracy of the coarse results. The STC tracker fully incorporates the temporal and spatial information that surround the tracked targets, so this method is adapted to reposition the target around the current position. Therefore, we follow the STC tracker to use the temporal and spatial information based on gray values of image patches during the fine processing.

To address the appearance variations, the context prior probability $P_t(\mathbf{c}(\mathbf{z})|o)$ as shown in Eq.4 is modified to also take into account the context prior probability of the original frame. To distinguish from $P_t(\mathbf{c}(\mathbf{z})|o)$ represented in Eq.4, we use $\bar{P}_t(\mathbf{c}(\mathbf{z})|o)$ to denote the context prior probability at frame t in our approach. We employ an adhoc method to compute the context prior probability using a simple linear interpolation:

$$\bar{P}_t(\mathbf{c}(\mathbf{z})|o) = \begin{cases} P_1(\mathbf{c}(\mathbf{z})|o), & t = 1 \\ (1-\theta)P_{t-1}(\mathbf{c}(\mathbf{z})|o) + \theta P_1(\mathbf{c}(\mathbf{z})|o), & 1 < t \le T, \end{cases} \tag{20}$$

where θ is a learning parameter, and $P_t(\mathbf{c}(\mathbf{z})|o)$ $(t = 1,2,\cdots)$ is the same as that represented in Eq.4.

Correspondingly, Eq.1 is modified to

$$\bar{m}_t(\mathbf{x}) = \sum_{\mathbf{c}(\mathbf{z})\in X_t^c} \bar{P}_t(\mathbf{x}|\mathbf{c}(\mathbf{z}),o)\bar{P}_t(\mathbf{c}(\mathbf{z})|o), \tag{21}$$

where

$$\bar{P}_t(\mathbf{x}|\mathbf{c}(\mathbf{z}),o) = H_t^{stc}(\mathbf{x}-\mathbf{z}). \tag{22}$$

In Eq.22, $H_t^{stc}(\mathbf{x}-\mathbf{z})$ is the same as that defined in Eq.9 and it reveals the spatial-temporal context relationship at frame t.

Then, from Eqs.4, 20,21 & 22, Eq.10 is modified to

$$\bar{m}_t(\mathbf{x}) = \begin{cases} H_1^{stc}(\mathbf{x}) \bigotimes (I_1(\mathbf{x}) w_{\sigma_1}(\mathbf{x}-\mathbf{x}_1^*)), & t = 1 \\ H_t^{stc}(\mathbf{x}) \bigotimes [(1-\theta) I_{t-1}(\mathbf{x}) w_{\sigma_{t-1}}(\mathbf{x}-\mathbf{x}_{t-1}^*) \\ \quad + \theta I_1(\mathbf{x}) w_{\sigma_1}(\mathbf{x}-\mathbf{x}_1^*)], & 1 < t \le T. \end{cases} \tag{23}$$

and the final object location $\mathbf{x}_t^{**}$ is determined by

$$\mathbf{x}_t^{**} = \arg\max_{\mathbf{x}\in\Omega_c(\mathbf{x}_t^*)} \bar{m}_t(\mathbf{x}), \qquad t = 1,2,\cdots,T \tag{24}$$

Note that $w_{\sigma_{t-1}}(\mathbf{z}-\mathbf{x}_{t-1}^*)$ $(t > 1)$ in Eq.23 is an attention function modeled from a Gaussian distribution (see Eq.5). This function exponentially weakens the value of location which is far from the target center. The targets and/or the background may be changing significantly over time.

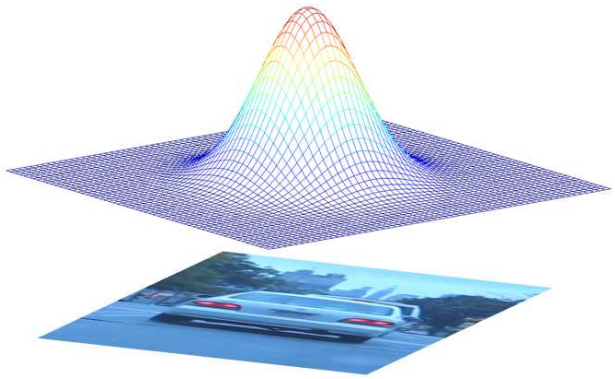

Figure 4 Context prior probability $P_t(\mathbf{c}(\mathbf{z})|o)$ defined in Eq.4. The probability is based on the image intensity $I_t(\mathbf{z})$ and a focus of attention function $w_{\sigma_t}(\mathbf{z}-\mathbf{x}_t^*)$. This function exponentially reduces the effect of location which is far from the target center and strengthens the effect of location around the center. Although the background is varying over time, the effect weight of background pixel remains at a low value.

However, in such a situation, the application of this attention function can significantly weaken the effect of the patch at the beginning tracking for the following reason. When the centre (i.e. $\mathbf{x}_1^*$) of this beginning patch is far away from the target center (i.e. $\mathbf{x}_{t-1}^*$), the weight (i.e. $w_{\sigma_1}(\mathbf{x}-\mathbf{x}_1^*)$) assigned to this beginning patch will usually be low because $\mathbf{x}$ in Eq. 23 is in the neighbourhood of $\mathbf{x}_{t-1}^*$ and is usually also far away from $\mathbf{x}_1^*$. The context prior probability model is shown in Figure 4.

The tracking procedure is summarized in **Algorithm** 1.

Algorithm 1 The proposed tracking method.

Input: Video Frame t=1:T
for t = 1:T **do**
  if t == 1 **then**
    Initialize the target location $\mathbf{x}_1^*$ and $\mathbf{x}_1^{**}$.
    Initialize the double-templates $\mathbf{s}_1$,
    Initialize the spatial context $h_1^{sc}(\mathbf{x})$ and set the spatio-temporal context $H_1^{stc}(\mathbf{x}) = h_1^{sc}(\mathbf{x})$.
  else
    (Begin the coarse processing at location $\mathbf{x}_{t-1}^{**}$)
    Draw m candidates $\mathbf{U}_t = \{\mathbf{u}_t^1, \mathbf{u}_t^2, \cdots, \mathbf{u}_t^m\}$.
    Construct the double-template $\mathbf{s}_t$.
    Compute i joint similarity $w_{joint}(\mathbf{s}_t, \mathbf{u}_t^i)$ via Eq.18.
    Compute the coarse position $\mathbf{x}_t^*$ via Eq.19.
    (Begin the fine processing at location $\mathbf{x}_t^*$)
    Compute $h_t^{sc}(\mathbf{x})$ by Eq.8 and $H_t^{stc}(\mathbf{x})$ by Eq.9 at location $\mathbf{x}_t^*$.
    Compute the confidence map $\bar{m}_t(\mathbf{x})$ based on $\mathbf{x}_t^*$ by Eq.23.
    Compute the ultimate position $\mathbf{x}_t^{**}$ via Eq.24.
  end if
end for
Output: Tracking results $\{\mathbf{x}_1^{**}, \mathbf{x}_2^{**}, \cdots, \mathbf{x}_T^{**}\}$.

4 Experiments

Here, we present the results of our experiments. Firstly, we perform a comprehensive evaluation of our adaptive sampling strategy for visual tracking. Secondly, we evaluate the novel double-template strategy for coarse matching. Thirdly, we evaluate our Gaussian-Uniform joint similarity. Fourthly, we evaluate the proposed video tracking scheme in all 20 videos with motion blur from the total 50 videos in the OTB-50 dataset.

Note that, in order to show that our approach has promising and favorable results on videos without the attribute of motion blue, we also provide a comparison of our tracker with other state-of-the-art methods tested on all videos of OTB-50 dataset and VOT 2015 dataset in Appendix A.1 and Appendix A.2, respectively.

4.1 Experimental environment and parameter settings

The experiments are implemented in Matlab R2014a. All trackers run on an i7 2.80 GHz CPU with 16 GB RAM.

The proposed method has several adjustable parameters. In our approach, we use the same parameter values of $b = 1$, $\beta = 1$ and $\rho = 0.086$ as suggested by [47].

For other parameters, we use all 11 videos with motion blur, selected from the OTB-100 dataset but do not belong to the OTB-50 dataset, to train the parameter values. The 11 videos are BlurCar1, BlurCar3, BlurCar4, Board, Boy, Car2, FleetFace, Girl2, Human2, Human7 and Tiger1. Table 1 shows the 11 video sequences together with their attributes including scale variation (SV), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), background clutters (BC), illumination variation (IV), out-of-view (OV), occlusion (OCC) and low resolution (LR).

We select the parameter values that produce optimal D-P and OP results on the 11 training videos as follows. In the process of primary location computation, the learning parameter in Eq.20 is chosen to be $\theta = 0.2$, which produces the best results as shown in Table 2. The parameters in Eq.12 and Eq.13 are set to be $a = 1.3$ and $d = 6$, which produce the best results as shown in Table 3 and Table 4, respectively. The threshold in Eq.17 is set to be $\kappa = 10$, which produces the best results as shown in Table 5. For the spatio-temporal context, the parameter α of the confidence map function is set to be $\alpha = 1.25$, which produces the best results as shown in Table 6. Note that, in the above parameter settings, the optimal value of each parameter is obtained by changing the values of this parameter and setting the values of the other parameters to the optimal values as mentioned above.

Table 1 The 11 tested sequences containing motion blur attribute. They are selected from the OTB-100 dataset . '$\surd$' indicates that the corresponding sequence has the corresponding attribute, and '$\times$' implies that the corresponding sequence does not have the corresponding attribute. SV, DEF, MB, FM, IPR, OPR, BC, IV, OV, OCC and LR represent the attributes of scale variation, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, background clutters, illumination variation, out-of-view, occlusion and low resolution, respectively.

Sequence	Frames	Main Challenges										
		SV	DEF	MB	FM	IPR	OPR	BC	IV	OV	OCC	LR
BlurCar1	742	×	×	$\surd$	$\surd$	×	×	×	×	×	×	×
BlurCar3	357	×	×	$\surd$	$\surd$	×	×	×	×	×	×	×
BlurCar4	380	×	×	$\surd$	$\surd$	×	×	×	×	×	×	×
Board	698	$\surd$	×	$\surd$	$\surd$	×	$\surd$	$\surd$	×	$\surd$	×	×
Boy	602	$\surd$	×	$\surd$	$\surd$	$\surd$	$\surd$	×	×	×	×	×
Car2	913	$\surd$	×	$\surd$	$\surd$	×	×	$\surd$	$\surd$	×	×	×
FleetFace	707	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	×	×	×	×	×
Girl2	1500	$\surd$	$\surd$	$\surd$	×	×	$\surd$	×	×	×	$\surd$	×
Human2	1128	$\surd$	×	$\surd$	×	×	$\surd$	×	$\surd$	×	×	×
Human7	250	$\surd$	$\surd$	$\surd$	$\surd$	×	×	×	$\surd$	×	$\surd$	×
Tiger1	354	×	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	×	$\surd$	×	$\surd$	×

4.2 Comparison on the 20 video sequences with motion blur in the OTB dataset

Dataset: We employ all 20 video sequences with motion blur attribute in the OTB-50 [41] for the tracking evaluation. The videos are Biker, BlurBody, BlurCar2, BlurFace, BlurOwl, Box, Car1, Clifbar, David, Deer, DragonBady, Human9, Ironman, Jump, Jumping, Liquor, MotorRolling, Soccer, Tiger2 and Woman. Similar to Table 1 for the 11 training video sequences, Table 7 shows the attributes, including scale variation (SV), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), background clutters (BC), illumination variation (IV), out-of-view (OV), occlusion (OCC) and low resolution (LR), of these evaluated video sequences.

Evaluation Methodology: For quantitative analysis, we use three evaluation criteria in [40] to compare the performance: the centre location error (CLE), distance precision (DP) and overlap precision (OP), all computed based on the manually labeled ground truth results of each frame. CLE is calculated based on the average Euclidean Distance between the estimated object centre and the ground-truth. The pixel error in each frame is defined as

$$CLE = \sqrt{(x-x_{gt})^2 + (y-y_{gt})^2}, \tag{25}$$

where (x,y) is the object location calculated by different trackers and (x_{gt}, y_{gt}) is the ground truth of each frame.

Besides CLE, we also compute the precision rate DP, which reflects the correlative number of frames in the video sequence where CLE is below a certain threshold. The DP score in a sequence is defined as $DP = num(CLE < \rho)/N$, where ρ represents the DP threshold, $num(\cdot)$ is the function to count the frames, and N is the frame number of a full sequence. The third evaluated metric OP at every frame is the percentage of frames where the bounding box overlap exceeds a threshold η. The OP score is defined as

$$OP = num\left(\frac{area(ROI \bigcap ROI_{gt})}{area(ROI \bigcup ROI_{gt})} > \eta\right)/N, \tag{26}$$

where $area(\cdot)$ is the function for calculating the area, ROI is the bounding box obtained by a tracker and ROI_{gt} is the bounding box provided by the ground truth.

4.3 Evaluation of techniques for coarse and fine processes

In this section, the experiments will show the impact of the proposed sampling strategy, double-template strategy, joint similarity and update scheme method. The WMIL sampling method is used to compared with the proposed sampling method in Section 4.3.1. The tracking approaches relying on proposed double-templates and various other templates are compared in Section 4.3.2. The tracking results based on the proposed similarity measurement and LS are compared in Section 4.3.3. A comparison of approaches with and without the update process (i.e., fine processing) is presented in Section 4.3.4.

Figure 5 shows a complete comparison of the proposed approach with the variations in sampling, templates and similarity measures with and without the update process (fine processing) on the 20 videos having motion blur selected from the OTB-50 dataset. In Figure 5, 'ours' represents the proposed approach; 'WMIL Sampling' represents the proposed approach with the sampling method replaced by the WMIL sampling method; 'First GT', 'Last Frame', 'Mixed Single Template', 'WMIL Template' and 'ANT Template' represent the proposed approach with the double-templates replaced by the 'First GT', 'Last Frame', 'Mixed Single Template', 'WMIL Template' and 'ANT Template' (referring to Section 4.3.2), respectively; 'LS' represents the proposed approach with the similarity method replaced by LS; and 'Without Updated' represents the proposed approach taken away the fine processing. The numbers in the brackets of Figure 5 stand for the the best mean OP values.

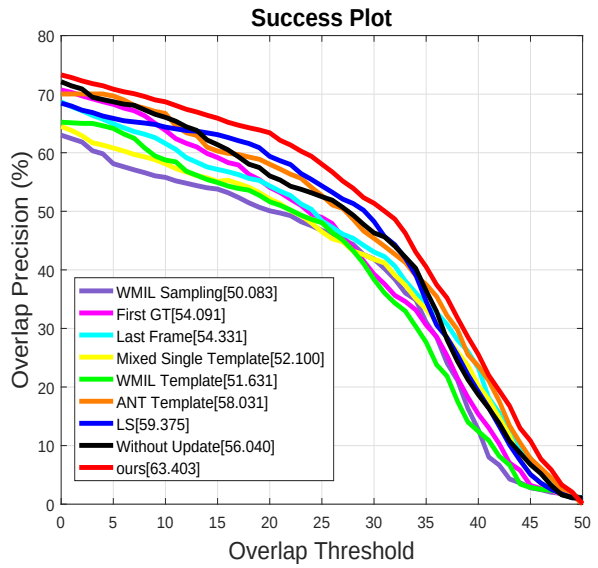

Figure 5 A comparison of the proposed approach with variations in sampling, templates and similarity measures with and without the update process tested on the 20 videos that have motion blur and are selected from the OTB-50 dataset. The figure of the success plot contains the mean overlap precision at a threshold ρ of 20 pixels (referring to [40]) for each method.

Table 8 A processing time (FPS) comparison between the WMIL sampling method and the proposed sampling method over the 20 videos with motion blur selected on OTB-50 dataset.

	WMIL sampling method	proposed method
Average FPS	1.41	29.47

4.3.1 Sampling strategy evaluation

As mentioned earlier, how to make a tracker be real-time is a crucial issue for most practical applications. In this paper, we propose a novel sampling strategy using an improved sliding window. This sampling technique, introduced in Section 3.1.1, is applied to touch the maximal search scope to return a minimal yet valid set of sampling candidates. Figure 5 and Table 8 show the results obtained using the proposed adaptive sampling method and the WMIL sampling method [46]. WMIL uses the conventional sliding window which slides across the whole search scope pixel by pixel. For fairness, we considered the proposed whole tracking method as a basic framework, and replaced the sampling processing by our sampling method and WMIL sampling method respectively to conduct comparison in Table 8. Furthermore, frame-per-second (FPS) regarding the run-timer performance is also provided for analyzing the trade-off between accuracy and efficiency in Tables 3 and 4. Based on the results, it is clear that $a = 1.3$ and $d = 6$ produce the best performance and the FPS approximates to the average FPS. The results clearly show that our sampling method provides a significant gain in speed while having higher tracking accuracies.

4.3.2 Robust double-template strategy

Figure 5 and Table 9 show a comparison of the processing speeds (FPS) of the proposed approach using different templates. These templates include the ground-truth target patch (named First GT), the obtained target patch in the previous frame (named Last Frame), a linear combination of the target patches found in the previous two frames (named Mixed Single template), the multi-templates proposed in the WMIL-based tracker [46] (named WMIL Template) and the multi-templates proposed in the ANT tracker [6] (named ANT Template). In this experiment, the Mixed Single Template at frame t is represented as the weighted sum of the two tracked target patches at frames $t-2$ and $t-1$ with weights of 0.05 and 0.95, respectively. From Figure 5 and Table 9, it shows that our method using the double-templates obtains the best mean OP and have the fast processing speed.

4.3.3 Gaussian-Uniform joint similarity evaluation

Figure 5 shows and compares the tracking results using the joint similarity measure proposed in Section 3.1.3 and L-S similarity measure. The joint similarity measure weakens the negative effects of outliers caused by motion blur. The results using the joint similarity measure achieves a mean OP of 63.403% at a threshold ρ of 20 pixels (referring to [40]), and the results using the LS similarity measure has an inferior mean OP of 59.375%. To conclude, the joint kernel similarity measure does improve the performance when compared with LS similarity measure.

4.3.4 Robust update scheme

This experiment shows the impact of the proposed update scheme (i.e., the fine processing) described in Section 3.2 after the coarse matching processing. Figure 5 shows that on all 20 evaluated sequences, the proposed update scheme significantly improves the performance of the tracker (please refer to the red and the black curves in the figure for the performance with and without the fine processing).

4.4 Comparison with state-of-the-art approaches

We compare the proposed approach with 12 state-of-the-art trackers, include STC [47], L1 [29], L1-APG [3], CT [48], ACT [12], HGDHT [8], TLD [20], STRUCK [15], C-NT [45], DLT [39], SRDCF [11], and DSST [10]. Note that STC, ACT, HGDHT, SRDCF, DSST and the proposed tracker are all using the Correlation Filter [12], L1 and L1-APG are based on the Particle Filter, and CNT and DLT are recently proposed CNN-based trackers.

4.4.1 Quantitative analysis

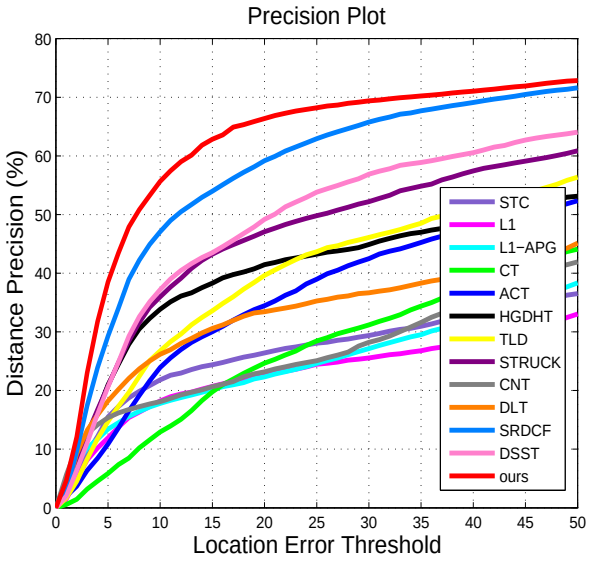

Figure 6 Precision plots for all 20 evaluated sequences containing motion blur selected from the OTB-50 dataset with location errors below a threshold ρ in the range of $[0,50]$ (pixels). The mean distance precision of each tracker is reported.

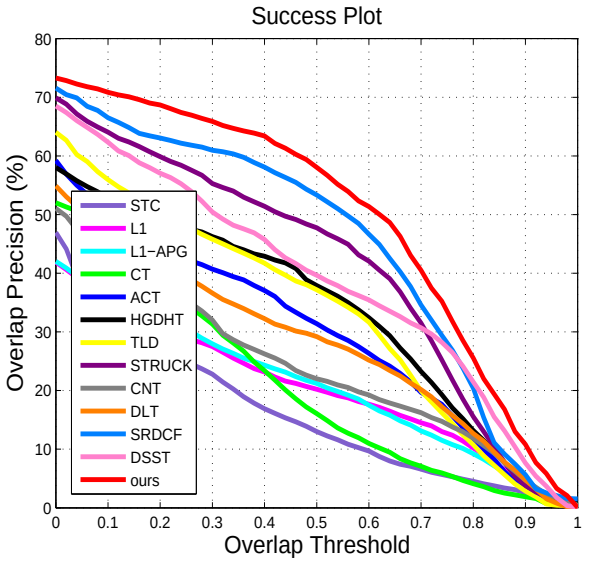

Figure 7 Success plots for all 20 evaluated sequences having motion blur selected from the OTB-50 dataset with overlap percentages over a threshold η in the range of $[0,1]$. The mean overlap precision of each tracker is reported.

Table 10 shows a comparison with the 12 state-of-the-art methods on the 20 challenging sequences containing motion blur in terms of mean distance precision (DP), mean overlap precision (OP) and frames per second (FPS). In Table 10. The best, second best and the third best results are shown in red, blue and green, respectively. As seen in Table 10, the performance of the proposed approach has always been the best in terms of DP and OP. Implemented on MATLAB, the proposed tracker runs at 29.47 FPS (a real-time speed) on an i7 2.80 GHz CPU with 16 GB RAM, putting our approach in the second best position in terms of processing speed.

Figures 6 and 7 show the precision and success plots in terms of DP and OP over all 20 sequences containing motion blur selected from the OTB-50 dataset. The values in the figure are the mean DPs at thresholds in the range of $[0,50]$ (pixels) and the mean OPs at thresholds in the range of $[0,1]$, respectively. As shown in these two figures, the proposed tracker has outperformed all of the 12 state-of-the-art trackers in terms of DP and OP for all of the threshold values of ρ in the range of [0,50] (pixels) and η in the range of [0,1], respectively.

4.4.2 Qualitative analysis

In Section 4.4.1, we have made a quantitative analysis of the proposed approach and have shown that the proposed tracker outperforms the 12 state-of-the-art approaches in terms of DP and OP on the 20 video sequences with motion blur. To demonstrate the superiority of the proposed tracker visually and provide a qualitative analysis, we select only three video sequences from the 20 video sequences and compare with six state-of-the-art trackers that also have concerned the effects of motion blur. As shown in Table 7, the three video sequences have the attribute of motion blur, and also have the attributes of scale variation, deformation, fast motion or in-plane rotation. Therefore, we make the qualitative analysis corresponding to these six attributes of scale variation, deformation, motion blur, fast motion and in-plane rotation, respectively.

Scale Variation: From Table 7, the objects suffer scale changes in the sequences BlurBody and BlurCar2. In the BlurBody sequence, Figure 8 demonstrates that our tracking method performs well when the objects undergo severe scale variation from frame #251 to frame #305, while the methods including STC, L1, L1-APG, CT and ACT completely fail to track the objects and HGDHT drifts to the background. This can be attributed to the reasons that: (1) the proposed double-template method takes into account the appearance of the object in the latest frame although the scale of the target is changed; and (2) our context prior probability is updated over time so as to be robust to appearance variations introduced by the scale variations. In the BlurCar2 sequence, our method and HGDHT perform better than other methods at frames #163, #285, #312 and #497. The other methods suffer from severe drift and some of these methods completely fail to track.

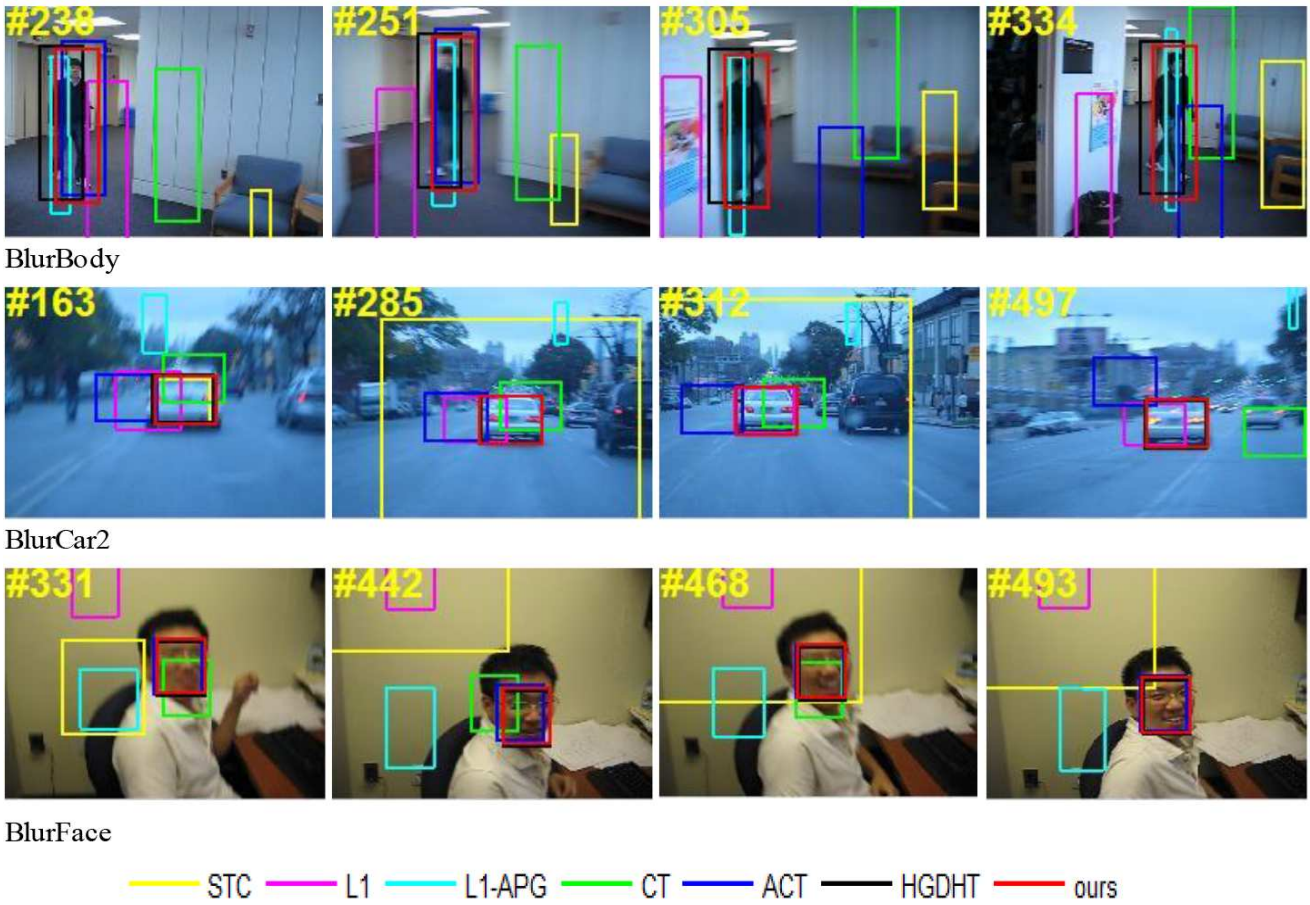

Figure 8 Snapshots of tracking results of our tracker and 6 state-of-the-art trackers from OTB-50 dataset. Note that the three videos have the attribute of motion blur. These videos also have the attributes of scale variation, deformation, fast motion or in-plane rotation.

Deformation: From Table 7, the BlurBody (Figure 8) sequence has the deformation attribute. Figure 8 demonstrates that our tracking method performs well at frames #238, #251, #305 and #334, while the other six methods fail to track the object in these frames. This can be attributed to the reasons that: (1) object representations do not experience significant changes in adjacent frames, and the proposed double-template strategy can record the current representation information of the deformed object. Thus, this template can adapt to the appearance changes of objects frame by frame; and (2) our updated prior probability also has a good adaptability for the appearance change caused by the deformation. Thus, our method can handle the non-rigid object deformation.

Motion Blur: From Table 7, the target regions are blurred due to the motion of the targets or the cameras in all sequences: BlurBody, BlurCar2 and BlurFace. Only the proposed tracker performs well over all three sequences while the rest methods suffer from severe drifts and even fail to track. This can be attributed to the reasons that: (1) the proposed kernel-based joint similarity measurement can effectively limit the impact of outliers and improves the robustness; and (2) the *Fine Processing* (Section 3.2) uses the spatial relationships and appearances of local contexts to discriminatively separate the target from background.

Fast Motion: From Table 7, all of the sequences BlurBody, BlurCar2 and BlurFace have the attribute of fast motion. It is hard to predict the location of a target when it undergoes a random and fast motion. As illustrated in the BlurBody sequence, when the camera suddenly and dramatically deviates from the original direction at frame #305, all evaluated algorithms except the proposed tracker cannot perform well. HGDHT performs well in the sequences BlurCar2 and BlurFace, but it has severe drifts in the BlurBody sequence. Although many evaluated sequences suffer from abrupt and random motions unpredictably, our tracker based on the proposed coarse-to-fine strategy still performs well. This can be attributed to the reasons that: (1) our sliding-window based sampling strategy explores the geometrical relationship between a search scope and a target displacement; and (2) the advantage of our optimization is to discover the reciprocal connection of objects and surroundings in time and space.

In-plane Rotation: From Table 7, there exist object rotations in the sequences: BlurBody and BlurFace. In the Blur-

Body sequence, our method performs better than other methods at frames #251 and #305. The other methods suffer from severe drift and some of these methods completely fail to track. This can be attributed to that the outstanding abilities of the double-templates and updated prior probability are robust to the appearance variations. Thus, our method can handle appearance changes and it is not sensitive to in-plane rotation. The same advantages of our tracking method are also demonstrated in the BlurFace sequence.

4.5 Discussion

As shown in our experiments, our method can address these factors including motion blurs, fast motions, cluttered background, occlusions and illumination changes more effectively. The reasons are listed as follows. (1) The proposed sampling method is combined with the latest geometrical displacement of a target to construct a reliable search scope, and it can ensure that the real tracked target image is included in the sample set, in the presence of fast motions. (2) The proposed double-template method contains the initial and updated information of a target. Therefore, it can effectively do the template matching for relocating, when the occluded target recovers again. (3) The joint kernel based similarity utilized a kernel mapping to reduce the effect of outliers on feature representation. This can improve the tracking accuracy in the scenarios of motion blurs and cluttered background. (4) The optimization approach is to construct a confidence map based on the Bayesian algorithm, and it can make full use of the structure advantage of the spatio-temporal context to infer the current location of a target. Therefore, the proposed tracking method can handle the target deformations, occlusions and cluttered background.

Overall, our method performs favorably against the other state-of-the-art tracking methods in the challenge sequences.

5 Conclusions and future work

This paper has proposed a novel tracking framework for shaking motions (SMT). This tracker interprets the tracking as a process of searching and then optimizing the searching for targets. We have employed two kernels to reduce the computation complexcity to linear order, and the kernel joint similarity actually leads to a nonparametric tracking algorithm. Specifically, we have introduced the double-templates for parallel matching. The use of the joint kernel similarity together with the parallel matching process has resulted in even more reduction in computation. To further improve the robustness, we have used the latest displacement of the target to guide a uniform sampling. Our method has exploited both temporal and spatial context information to optimize the tracking. Particularly, we have modified the context prior probability for the sake of a better adaption to the target appearance variations. Experimental results on some challenging video sequences have shown that SMT can robustly track shaking targets and outperforms the existing 12 state-of-the-art trackers.

We have used all 11 videos with motion blur, and they are selected from the OTB-100 dataset but do not belong to the OTB-50 dataset, for training the parameters in the proposed algorithm. Then, we have used all 20 videos with motion blur from the total 50 videos in the OTB-50 dataset for testing. For these videos with motion blur, our tracker SMT has achieved the best performance in terms of precision and success rates, compared with the state-of-the-art methods, and the comparison results have been shown in Section 4.4. Although our algorithm is particularly designed for videos with motion blur and both OTB-50 dataset and VOT 2015 dataset contain many videos without any motion blur, we have also done the experiments on the whole dataset and the results using the proposed method are still favorable. A comparison with the state-of-the-art methods testing on the complete datasets of OTB-50 and VOT 2015 can be found in Appendix A.

An interesting direction for further work is to introduce CNN based methods [33] and [44] into our work, which may further improve performance with some computation costs. We also aim to generalize this framework to other operators in the future, such as scale variations or non-rigid deformations.

6 Conflict of interest

No conflict of interest.

Acknowledgements This work was supported by Fujian Provincial Department of Science and Technology (Grant No. 2015H0021).

References

1. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on, vol. 1, pp. 798–805. IEEE (2006)
2. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33**(8), 1619–1632 (2011)
3. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust l1 tracker using accelerated proximal gradient approach. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 1830–1837. IEEE (2012)
4. Bhattacharyya, A.: On a measure of divergence between two multinomial populations. Sankhyā: The Indian Journal of Statistics pp. 401–406 (1946)

5. Black, M.J., Jepson, A.D.: Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. International Journal of Computer Vision **26**(1), 63–84 (1998)
6. Čehovin, L., Leonardis, A., Kristan, M.: Robust visual tracking using template anchors. In: Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on, pp. 1–8. IEEE (2016)
7. Cuevas, E., Zaldivar, D., Rojas, R.: Kalman filter for vision tracking (2005)
8. Dai, M., Cheng, S., He, X.: Hybrid generative–discriminative hash tracking with spatio-temporal contextual cues. Neural Computing and Applications pp. 1–11 (2016)
9. Dai, M., Lin, P., Wu, L., Chen, Z., Lai, S., Zhang, J., Cheng, S., He, X.: Orderless and blurred visual tracking via spatio-temporal context. In: MultiMedia Modeling, pp. 25–36. Springer (2015)
10. Danelljan, M., Hager, G., Khan, F.S., Felsberg, M.: Discriminative scale space tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence (2016)
11. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4310–4318 (2015)
12. Danelljan, M., Khan, F.S., Felsberg, M., Weijer, J.v.d.: Adaptive color attributes for real-time visual tracking. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pp. 1090–1097. IEEE (2014)
13. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: Computer Vision–ECCV 2008, pp. 234–247. Springer (2008)
14. Hamming, R.W.: Error detecting and error correcting codes. Bell System technical journal **29**(2), 147–160 (1950)
15. Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.M., Hicks, S.L., Torr, P.H.: Struck: Structured output tracking with kernels. IEEE transactions on pattern analysis and machine intelligence **38**(10), 2096–2109 (2016)
16. Hare, S., Saffari, A., Torr, P.H.: Struck: Structured output tracking with kernels. In: Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 263–270. IEEE (2011)
17. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: Computer Vision–ECCV 2012, pp. 702–715. Springer (2012)
18. Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: Robust online appearance models for visual tracking. Pattern Analysis and Machine Intelligence, IEEE Transactions on **25**(10), 1296–1311 (2003)
19. Julier, S.J., Uhlmann, J.K., Durrant-Whyte, H.F.: A new approach for the nonlinear transformation of means and covariances in linear filters. IEEE Transactions on Automatic Control (1996)
20. Kalal, Z., Matas, J., Mikolajczyk, K.: Pn learning: Bootstrapping binary classifiers by structural constraints. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 49–56. IEEE (2010)
21. Koikkalainen, J., Lötjönen, J., Thurfjell, L., Rueckert, D., Waldemar, G., Soininen, H., Initiative, A.D.N., et al.: Multi-template tensor-based morphometry: application to analysis of alzheimer's disease. NeuroImage **56**(3), 1134–1144 (2011)
22. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Cehovin, L., Fernandez, G., Vojir, T., Hager, G., Nebehay, G., Pflugfelder, R.: The visual object tracking vot2015 challenge results. In: The IEEE International Conference on Computer Vision (ICCV) Workshops (2015)
23. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 1269–1276. IEEE (2010)
24. Kwon, J., Lee, K.M.: Tracking by sampling trackers. In: Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 1195–1202. IEEE (2011)
25. Lei, J.B., Yin, J.B., Shen, H.B.: Gfo: A data driven approach for optimizing the gaussian function based similarity metric in computational biology. Neurocomputing **99**(1), 307–315 (2013)
26. Li, P., Zhang, T., Ma, B.: Unscented kalman filter for visual curve tracking. Image and Vision Computing **22**(2), 157–164 (2004)
27. Liu, M., Zhang, D., Shen, D.: Relationship induced multi-template learning for diagnosis of alzheimers disease and mild cognitive impairment. IEEE transactions on medical imaging **35**(6), 1463–1474 (2016)
28. Mahalanobis, P.C.: On the generalized distance in statistics. Proceedings of the National Institute of Sciences (Calcutta) **2**, 49–55 (1936)
29. Mei, X., Ling, H.: Robust visual tracking using l1 minimization. In: Computer Vision, 2009 IEEE 12th International Conference on, pp. 1436–1443. IEEE (2009)
30. Min, R., Wu, G., Cheng, J., Wang, Q., Shen, D.: Multi-atlas based representations for alzheimer's disease diagnosis. Human brain mapping **35**(10), 5052–5070 (2014)
31. Models, S.: Stochastic models, estimation, and control. Academic Press, (1979)
32. Oron, S., Bar-Hillel, A., Levi, D., Avidan, S.: Locally orderless tracking. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 1940–1947. IEEE (2012)
33. Oyedotun, O.K., Khashman, A.: Deep learning in vision-based static hand gesture recognition. Neural Computing and Applications pp. 1–11 (2016)
34. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. International Journal of Computer Vision **77**(1-3), 125–141 (2008)
35. Song, H.: Robust visual tracking via online informative feature selection. Electronics Letters **50**(25), 1931–1933 (2014)
36. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Computer Vision, 1998. Sixth International Conference on, pp. 839–846. IEEE (1998)
37. Uhlmann Simon J. Julier, J.K.: A new extension of the kalman filter to nonlinear systems **3068**, 182–193 (1997)
38. Van De Weijer, J., Schmid, C., Verbeek, J., Larlus, D.: Learning color names for real-world applications. Image Processing, IEEE Transactions on **18**(7), 1512–1523 (2009)
39. Wang, N., Yeung, D.Y.: Learning a deep compact image representation for visual tracking. In: Advances in neural information processing systems, pp. 809–817 (2013)
40. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2411–2418 (2013)
41. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: Computer vision and pattern recognition (CVPR), 2013 IEEE Conference on, pp. 2411–2418. IEEE (2013)
42. Yang, C., Duraiswami, R., Davis, L.: Efficient mean-shift tracking via a new similarity measure. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 176–183. IEEE (2005)
43. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. Acm computing surveys (CSUR) **38**(4), 13 (2006)
44. Zhang, H., Cao, X., Ho, J.K.L., Chow, T.W.S.: Object-level video advertising: An optimization framework. IEEE Transactions on Industrial Informatics **PP**(99), 1–1 (2016)
45. Zhang, K., Liu, Q., Wu, Y., Yang, M.H.: Robust visual tracking via convolutional networks without training. IEEE Transactions on Image Processing **25**(4), 1779–1792 (2016)
46. Zhang, K., Song, H.: Real-time visual tracking via online weighted multiple instance learning. Pattern Recognition **46**(1), 397–411 (2013)
47. Zhang, K., Zhang, L., Liu, Q., Zhang, D., Yang, M.H.: Fast visual tracking via dense spatio-temporal context learning. In: Computer Vision–ECCV 2014, pp. 127–141. Springer (2014)
48. Zhang, K., Zhang, L., Yang, M.H.: Real-time compressive tracking. In: Computer Vision–ECCV 2012, pp. 864–877. Springer (2012)

49. Zhou, Q.H., Lu, H., Yang, M.H.: Online multiple support instance tracking. In: Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on, pp. 545–552. IEEE (2011)
50. Zhou, T., He, X., Xie, K., Fu, K., Zhang, J., Yang, J.: Robust visual tracking via efficient manifold ranking with low-dimensional compressive features. Pattern Recognition **48**(8), 2459–2473 (2015)

Table 2 The average DPs and average OPs over all 11 tested video sequences. We highlight the results when $\theta = 0.2$. Here, we set the DP values at the threshold ρ of 20 pixels and the OP values at the threshold η of 0.5 (referring to [40]).

θ	0.00	0.05	0.10	0.15	**0.20**	0.25	0.30	0.35	0.40	0.45	0.50
DP	68.96	63.07	73.07	83.08	**83.48**	82.68	80.45	78.96	79.34	75.63	79.80
OP	73.06	74.42	78.14	77.65	**81.09**	76.59	78.97	75.41	73.71	75.37	74.14
θ	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00	
DP	75.86	74.26	78.44	77.25	75.71	73.15	78.07	69.02	71.83	70.05	
OP	77.44	76.36	74.38	78.46	76.74	80.63	76.98	74.89	68.31	62.43	

*DP represents the means of the average DP; OP represents the means of the average OP.

Table 3 The average DPs and average OPs over all 11 tested video sequences. We highlight the results when $a = 1.3$. Here, we set the DP values at the threshold ρ of 20 pixels and the OP values at the threshold η of 0.5 (referring to [40]).

a	1.00	1.10	1.20	**1.30**	1.40	1.50	1.60	1.70	1.80	1.90	2.00
DP	81.28	83.27	81.30	**83.48**	78.39	79.04	82.62	81.03	83.08	76.25	77.09
OP	73.29	75.75	74.14	**81.09**	68.39	67.04	68.37	71.91	72.24	73.94	74.17
FPS	29.51	29.50	29.48	**29.47**	29.46	29.46	29.45	29.44	29.42	29.41	29.41

*DP represents the means of the average DO; OP represents the means of the average OP.

Table 4 The average DPs and average OPs over all 11 tested video sequences. We highlight the results when $d = 6$. Here, we set the DP values at the threshold ρ of 20 pixels and the OP values at the threshold η of 0.5 (referring to [40]).

d	1.00	2.00	3.00	4.00	5.00	**6.00**	7.00	8.00	9.00	10.00
DP	79.49	77.48	78.70	76.85	81.42	**83.48**	78.37	79.53	81.91	76.45
OP	77.77	77.99	77.94	77.94	79.29	**81.09**	79.29	79.29	79.29	79.29
FPS	29.33	29.34	29.37	29.38	29.41	**29.47**	29.50	29.53	29.54	29.57

*DP represents the means of the average DP; OP represents the means of the average OP.

Table 5 The average DPs and average OPs over all 11 tested video sequences. We highlight the results when $\kappa = 10$. Here, we set the DP values at the threshold ρ of 20 pixels and the OP values at the threshold η of 0.5 (referring to [40]).

κ	1.00	1.50	2.00	2.50	3.00	3.50	4.00	4.50	5.00	5.50	6.00	6.50	7.00
DP	75.60	75.60	75.04	75.04	71.25	72.25	78.76	74.26	75.81	76.12	80.41	75.41	70.47
OP	64.41	63.75	62.72	63.05	67.58	69.29	70.34	71.56	73.29	74.17	75.36	74.75	76.98
κ	7.50	8.00	8.50	9.00	9.50	**10.00**	10.50	11.00	11.50	12.00	12.50	13.00	13.50
DP	75.47	75.52	76.85	75.52	72.08	**83.48**	81.29	82.02	80.41	77.20	76.25	74.14	72.74
OP	77.81	77.29	75.56	78.59	79.86	**81.09**	77.87	79.29	78.29	78.94	76.93	75.91	74.90
κ	14.00	14.50	15.00	15.50	16.00	16.50	17.00	17.50	18.00	18.50	19.00	19.50	20.00
DP	74.46	69.46	62.43	67.47	63.97	62.07	61.46	60.47	59.89	58.46	58.02	57.46	55.70
OP	73.39	72.31	71.89	70.39	69.86	67.82	68.39	65.34	64.57	62.08	61.82	60.82	58.96

*DP represents the means of the average DP; OP represents the means of the average OP.

Table 6 The average DPs and average OPs over all 11 tested video sequences. We highlight the results when $\alpha = 1.25$. Here, we set the DP values at the threshold ρ of 20 pixels and the OP values at the threshold η of 0.5 (referring to [40]).

α	1.00	1.05	1.10	1.15	1.20	**1.25**	1.30	1.35	1.40	1.45	1.50	1.55	1.60	1.65
DP	83.39	83.21	82.18	82.01	82.39	**83.48**	82.91	79.47	83.17	81.74	83.16	78.63	76.14	76.12
OP	76.27	77.86	77.68	78.34	79.76	**81.09**	80.29	79.82	77.74	76.69	75.74	74.36	72.56	71.45
α	1.70	1.75	1.80	1.85	1.90	1.95	2.00	2.05	2.10	2.15	2.20	2.25	2.30	2.35
DP	76.63	75.49	74.58	74.40	73.12	73.17	71.51	71.12	72.21	75.07	78.02	80.78	78.26	77.46
OP	72.74	71.04	69.78	69.09	72.69	75.74	72.35	76.47	77.36	76.43	75.87	79.09	76.47	75.49
α	2.40	2.45	2.50	2.55	2.60	2.65	2.70	2.75	2.80	2.85	2.90	2.95	3.00	
DP	82.72	83.07	83.16	81.63	82.77	81.98	82.35	82.35	81.68	80.15	78.23	76.22	75.66	
OP	73.45	72.78	71.75	75.87	72.89	76.85	74.78	73.19	72.64	75.15	77.85	78.64	77.52	

*DP represents the means of the average DP; OP represents the means of the average OP.

Table 7 The 20 video sequences that are selected from OTB-50 dataset and have the attribute of motion blur. '$\surd$' indicates that the corresponding sequence has the corresponding challenge, and '$\times$' implies that the corresponding sequence does not have the corresponding attribute. SV, DEF, MB, FM, IPR, OPR, BC, IV, OV, OCC and LR represent the attributes of scale variation, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, background clutters, illumination variation, out-of-view, occlusion and low resolution, respectively.

Sequence	Frames	Main Challenges										
		SV	DEF	MB	FM	IPR	OPR	BC	IV	OV	OCC	LR
Biker	142	$\surd$	$\times$	$\surd$	$\surd$	$\times$	$\surd$	$\times$	$\times$	$\surd$	$\surd$	$\times$
BlurBody	334	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
BlurCar2	585	$\surd$	$\times$	$\surd$	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
BlurFace	493	$\times$	$\times$	$\surd$	$\surd$	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
BlurOwl	631	$\surd$	$\times$	$\surd$	$\surd$	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
Box	1161	$\surd$	$\times$	$\surd$	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\times$
Car1	1020	$\surd$	$\times$	$\surd$	$\surd$	$\times$	$\times$	$\surd$	$\surd$	$\times$	$\times$	$\surd$
Clifbar	472	$\surd$	$\times$	$\surd$	$\surd$	$\surd$	$\times$	$\surd$	$\times$	$\surd$	$\surd$	$\times$
David	761	$\surd$	$\surd$	$\surd$	$\times$	$\surd$	$\surd$	$\times$	$\surd$	$\times$	$\surd$	$\times$
Deer	71	$\times$	$\times$	$\surd$	$\surd$	$\surd$	$\times$	$\surd$	$\times$	$\times$	$\times$	$\surd$
DragonBaby	113	$\surd$	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\times$	$\surd$	$\surd$	$\times$
Human9	305	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\times$	$\times$	$\surd$	$\times$	$\times$	$\times$
Ironman	166	$\surd$	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
Jump	122	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\times$	$\times$	$\surd$	$\times$
Jumping	313	$\times$	$\times$	$\surd$	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
Liquor	1741	$\surd$	$\times$	$\surd$	$\surd$	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\times$
MotorRolling	164	$\surd$	$\times$	$\surd$	$\surd$	$\surd$	$\times$	$\surd$	$\surd$	$\times$	$\times$	$\surd$
Soccer	392	$\surd$	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\surd$	$\times$
Tiger2	365	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\surd$	$\surd$	$\surd$	$\times$
Woman	597	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\surd$	$\times$	$\surd$	$\times$	$\surd$	$\times$

Table 9 A comparison the processing time (FPS) of the proposed approach with five different templates including the proposed double-templates over the 20 videos with motion blur selected OTB-50. The best results are shown in **red** while the second and third ones are shown in **blue** and **green** respectively.

	First GT	Last Frame	Mixed Single Template	WMIL Template	ANT Template	ours
Average FPS	**25.19**	**25.37**	24.89	10.65	8.12	**29.47**

Table 10 Quantitative comparison of our trackers with 12 state-of-the-art methods on 20 challenging sequences with motion blur attribute on OTB-50 dataset. The results are reported in distance precision (DP) (%). We also provide the average values of DP. Here, we set the DP values at a threshold ρ of 20 pixels (referring to [40]) and the OP values at a threshold η of 0.5 (referring to [40]). The best results are shown in **red** while the second and third ones are shown in **blue** and **green** respectively. Note that the proposed approach achieves the best average performance in terms of average DP and average OP, and the second best in terms of FPS.

	STC[47]	L1[29]	L1-APG[3]	CT[48]	ACT[12]	HGDHT[8]	TLD[20]	STRUCK[15]	CNT[45]	DLT[39]	SRDCF[11]	DSST[10]	OURS
Average DP	26.40	22.69	22.33	24.66	34.44	41.45	39.63	47.06	23.17	33.44	**59.17**	**49.18**	**66.38**
Average OP	13.01	20.22	21.21	16.06	31.38	37.87	37.17	**47.74**	21.99	29.18	**53.33**	39.64	**58.03**
Average FPS	25.19	2.13	10.65	13.56	**103.39**	**29.12**	10.56	13.91	0.42	7.08	3.46	24.61	**29.47**

Appendix A

A.1 Comparison on the whole OTB-50 dataset

Although the focus of our work is on the motion blur, we also provide the experimental results over other videos on the whole OTB-50 dataset.

Table 11 provides a comparison with the 12 state-of-the-art trackers in terms of OP, DP and FPS on the whole OTB-50 dataset. The best three results are marked in red, blue and green respectively.

Figures 9 and 10 show the precision and success plots showing the mean distance precision (DP) and mean overlap precision (OP) over all sequences in OTB-50 dataset. The values in the figure are the mean DPs at the thresholds in the range of $[0,50]$ (pixels) and the mean OPs at the thresholds in the range of $[0,1]$, respectively. In the precision plot, our tracker presents the second best performance. In the success plot, our tracker is among the best three. These two figures show that our tracker can also perform well in other videos although they do not have the attribute of motion blur.

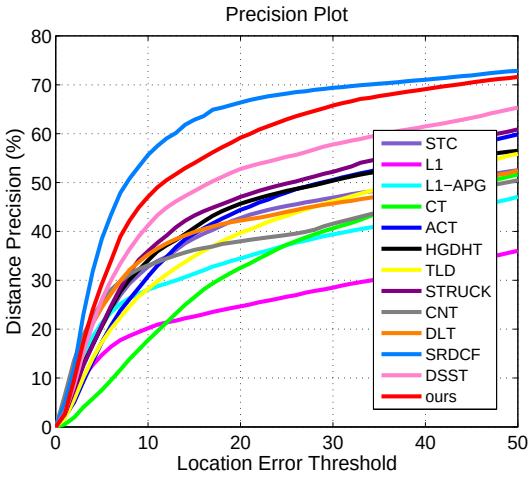

Figure 9 Precision plot for all 50 sequences in OTB-50 dataset with location errors below a threshold in the range of $\rho \in [0,50]$ (pixels). The mean distance precision of each tracker is reported.

A.2 Comparison on the whole VOT 2015 dataset

In this section, we present the results on the whole VOT 2015 dataset [22]. We compare our tracker with 10 state-of-the-art trackers that have also been previously tested over all 60 video sequences in this dataset. In VOT 2015, the trackers are compared in terms of both accuracy and robustness. The comparison results are shown in Table 12. In Table 12, the Accuracy Rank is equivalent to the number of frames where the overlap exceeds a certain threshold. The

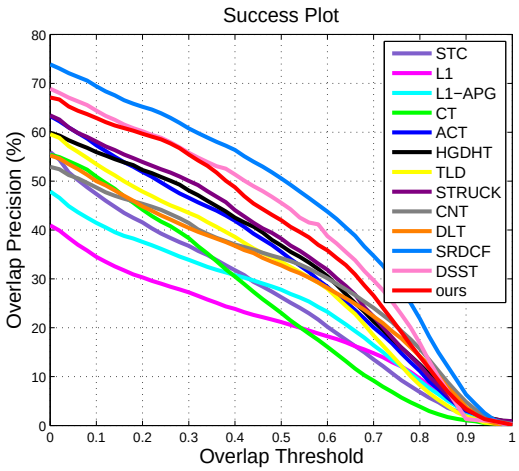

Figure 10 Success plot for all 50 evaluated sequences in OTB-50 dataset with overlap percentages over a threshold in the range of $\eta \in [0,1]$. The mean overlap precision of each tracker is reported.

Robustness Rank counts the number of times each tracker fails track in a video. Each tracker always restarts at the fifth frame after a tracking failure occurs. The Final Tracker Rank is the mean score of a tracker in terms of accuracy and robustness over all of the video sequences. More details how to evaluate the tracking on the VOT 2015 can be found in [22].

Figure 11 shows the ranking plots, in terms of Accuracy Rank and Robustness Rank, of the proposed tracker and the 10 state-of-the-art trackers in the VOT 2015 dataset. Our approach still works well both with second best results in terms of Accuracy Rank and the fourth best in terms of Robustness Rank.

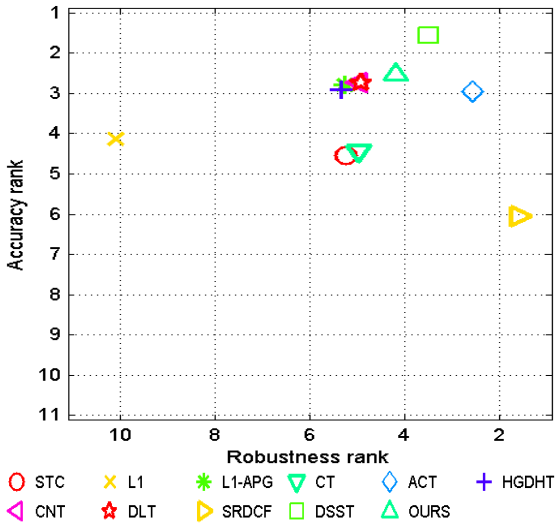

Figure 11 Ranking plot for the experiment baseline in the VOT 2015 dataset. The accuracy and robustness ranks are plotted along the vertical and horizontal axis respectively. Our method (denoted by the green triangle) achieves superior results in the accuracy-robustness experiments.

Table 11 A comparison of our tracker with 12 state-of-the-art trackers on the whole OTB-50 dataset. The results are reported in terms of average overlap precision(OP)(%), average distance precision(DP)(%) and frame per second (FPS). Here, DP values are obtained at a threshold ρ of 20 pixels (referring to [40]) and the OP values are obtained at a threshold η of 0.5 (referring to [40]). The best results are displayed in **red** while the second and third best results are shown in **blue** and **green**, respectively. The results of our tracker are among the top three.

	STC[47]	L1[29]	L1-APG[3]	CT[48]	ACT[12]	HGDHT[8]	TLD[20]	STRUCK[15]	CNT[45]	DLT[39]	SRDCF[11]	DSST[10]	OURS
Average DP	43.92	23.87	34.16	32.60	47.09	45.86	39.98	47.16	37.98	42.44	**66.38**	**52.78**	**59.92**
Average OP	27.20	20.58	28.15	23.09	36.81	37.07	33.62	38.15	34.23	32.87	**52.48**	**45.51**	**42.89**
Average FPS	22.77	1.47	9.64	11.85	**167.00**	22.21	10.5	11.87	0.42	7.57	4.09	**37.72**	**23.59**

Table 12 A baseline comparison of our method with 10 state-of-the-art trackers on all 60 challenging videos in the VOT 2015 dataset. The accuracy and robustness ranks, along with the final averaged ranking score, are reported. The average overlaps and failures over the videos are also shown in the last two rows. The best results are marked in **red** while the second and third best results are marked in **blue** and **green**, respectively.

	STC[47]	L1[29]	L1-APG[3]	CT[48]	ACT[12]	HGDHT[8]	CNT[45]	DLT[39]	SRDCF[11]	DSST[10]	OURS
Accuracy Rank	4.55	4.13	2.80	4.45	2.95	2.92	**2.73**	2.75	6.07	**1.57**	**2.52**
Robustness Rank	5.25	10.10	5.27	4.95	**2.57**	5.35	4.95	4.90	**1.60**	**3.48**	4.18
Final Rank	4.90	7.12	4.04	4.70	**2.76**	4.14	3.84	3.83	3.84	**2.53**	**3.35**
Overlap	0.40	0.44	**0.47**	0.39	0.46	0.45	**0.48**	**0.47**	0.32	**0.54**	**0.47**
Failures	3.75	14.64	4.65	4.09	**2.05**	4.07	3.84	3.61	**1.07**	**2.56**	3.09

Note that, although our algorithm is particularly designed for videos with motion blur and VOT 2015 dataset contains many videos without any motion blur, the experiments performed on the whole dataset of VOT 2015 still show favorable results as demonstrated in Table 12 and Figure 11.