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

Hybrid Generative-Discriminative Hash Tracking with Spatio-Temporal Contextual Cues

Manna Dai · Shuying Cheng · Xiangjian He

Received: date / Accepted: date

Abstract Visual object tracking is of a great application value in video monitoring systems. Recent work on video tracking has taken into account spatial relationship between the targeted object and its background. In this paper, the spatial relationship is combined with the temporal relationship between features on different video frames so that a real-time tracker is designed based on a Hash algorithm with spatio-temporal cues. Different from most of the existing work on video tracking, which is regarded as a mechanism for image matching or image classification alone, we propose a hierarchical framework and conduct both matching and classification tasks to generate a coarse-to-fine tracking system. We develop a generative model under a modified Particle Filter with Hash fingerprints for the coarse matching by the Maximum a Posteriori (MAP) and a discriminative model for the fine classification by maximizing a confidence map based on a context model. The confidence map reveals the spatio-temporal dynamics of the target. Because Hash fingerprint is merely a binary vector and the modified Particle Filter uses only a small number of particles, our tracker has a low computation cost. By conducting experiments on 8 challenging video sequences from a public benchmark, we demonstrate that our tracker outperforms 8 state-of-the-art trackers in terms of both accuracy and speed.

M. Dai
Fuzhou University, China
University of Technology, Sydney, Australia
E-mail: Manna.Dai@student.uts.edu.au

S. Cheng (Corresponding author)
Fuzhou University, China
E-mail: sycheng@fzu.edu.cn

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

Keywords Hash algorithm · spatio-temporal cues · hierarchical framework · Maximum a Posteriori (MAP) · confidence map

1 Introduction

Visual object tracking is attracting more and more researchers' attention due to its potential applications, which are commonly found in video surveillance [7,8,23, 16,27], sports analysis [19,29] and human motion recognition [11,31,34]. Interesting applications reported in these studies include crowd control, traffic monitoring and transportation security. Several factors, such as cluttered scene, illumination change, motion blur and fast motion complicate a tracking problem. Most state-of-the-art tracking approaches either rely on generative methods [14,15,25,38,40], discriminative methods [2, 6,9,10,13] or hybrid methods [33,39] to handle visual tracking.

Generative methods aim at building a model based on the object appearance of interest, and then search for the object appearance which best matches the learned appearance. Recently, Zhou et al. [40] introduced a novel appearance model that fused colour distributions and spatio-temporal motion energy. Zhang et al. [38] proposed a part matching tracker (PMT) which utilized the leveraging multi-mode target templates. The local orderless tracker (LOT) [22] applied the Earth Mover's Distance (EMD) [24] into the generatively probabilistic model. Generative methods often outperform discriminative methods when the size of training data is small. However, the common weakness of generative methods is that they cannot discriminate an object of interest from the background because they consider only object similarity. In this case, they are prone to drift.

Discriminative methods regard the tracking as a binary classification problem. They attempt to distinguish an object from its surrounding background without the description of the object. We state some of the most recent methods as follows. The multiple instance learning tracker (MIL) in [1] trained a classifier online which was bootstrapped to extract positive and negative examples. The weighted MIL tracker (WMIL) in [35] improved MIL by assigning weights to the samples according to their importance when it was trained. Henriques et al. [10] derived the Kernelized Correlation Filter (KCF) tracker based on Histogram of Oriented Gradients (HOG) features instead of raw pixels. Discriminative methods often outperform generative methods if there are enough training data. However, discriminative methods, in general, cannot well adapt to appearance changes [12], especially in the scenarios with motion blur. Furthermore, a discriminative method may extract insignificant positive samples during its learning stage when sample's importance is not known, so that its tracking performance may be degraded.

Recently, several hybrid algorithms were proposed for benefitting from both types of methods. Qian et al. [33] proposed to encode appearance changes by a generative model and reacquired the targeted object after a full occlusion occurred by a discriminative classifier. Similarly, Zhong et al. [39] combined a sparsity-based discriminative classifier with a histogram-based generative model for tracking an object of interest.

In this paper, we present a Hybrid Generative and Discriminative Hash Tracker (HGDHT), which is insensitive to scene clutter, illumination variation, motion blur and abrupt motion. Different from the existing hybrid methods that perform discriminative classification only when it is needed, our method sequentially executes both a generative tracker and a discriminative tracker.We intend to solve the main issues described as follows. Firstly, feature vectors with high dimensionality lead to expensive computation. Secondly, searching algorithms generally fail to balance the efficiency and speed. Thirdly, an effective matching and a classifier are hard to design when the object and its background are similar.

To tackle the above problems, we implement the integration of Hash fingerprints and spatio-temporal contextual cues. We generatively construct the appearance model of an object based on the Hash fingerprints of the primarily located object. Both a low-density sampling and simple features in the Hash algorithm [17] can reduce the computational complexity under the Particle Filter framework [20,26]. The spatial and temporal cues of the object and its surroundings are discriminatively fused into a confidence map, which is converted to the Fourier domain by FFT for expediting the detection process. This fusion strategy helps improve the detection accuracy, especially when the object appearance is changed due to deformation, occlusion, rotation, illumination variation and motion blur. The optimized confidence map in each frame provides the tracking results. Experimental results demonstrate our superiority when compared with state-of-the-art approaches.

This paper is an extension of our paper showing preliminary results in [5]. We highlight the main and new contributions of this paper as follows.

- We propose a hybrid tracker which has both a generative tracker and a discriminative tracker. The "generative to discriminative" scheme generates a "coarse to fine" result, and hence results in fast yet accurate performance.
- We design a simplified Particle Filter framework for primary position calibration. We use only a small number of particles to establish the Maximum a Posteriori (MAP), instead of massive particles that the conventional Particle Filter uses, in order to infer the coarse position of an object.
- Compared with the preliminary work in [5], our HGDHT makes use of Hash fingerprints and can help decrease the calculation complexity.

The rest of this paper is organized as follows. Section 2 introduces the related work for immediate reference. In Section 3, the proposed approach is presented. Section 4 makes qualitative and quantitative comparisons with 8 state-of-the-art approaches on 8 publicly available video sequences. At last, some concluding remarks are demonstrated in Section 5.

2 Related Work

2.1 Spatio-Temporal Context Model

Our method uses the spatio-temporal context model in the STC tracker [36] which reveals the relationship between an object of interest and its local context. Here, we provide a brief overview of this approach.

In STC tracker, a tracking problem is formulated by computing a confidence map, namely the object location likelihood. We get the current object location $\mathbf{x}^*$ and the feature set $X^c = \{\mathbf{c}(\mathbf{z}) = (I(\mathbf{z}), \mathbf{z}) | \mathbf{z} \in \Omega_c(\mathbf{x}^*)\}$ where $I(\mathbf{z})$ represents the image intensity at location $\mathbf{z}$ and $\Omega_c(\mathbf{x}^*)$ is the neighborhood of location $\mathbf{x}^*$. The object location likelihood at $\mathbf{x}$ under the Bayesian framework is computed by

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

where o represents the object.

The spatial context model is defined as a conditional probability function

$$P(\mathbf{x}|\mathbf{c}(\mathbf{z}), o) = h^{sc}(\mathbf{x} - \mathbf{z}), \tag{2}$$

where $h^{sc}(\mathbf{x}-\mathbf{z})$ is a spatial context function regarding the relative displacement between object location $\mathbf{x}$ and its local context location $\mathbf{z}$.

We model the context prior probability in Eq. 1 as

$$P(\mathbf{c}(\mathbf{z})|o) = I(\mathbf{z})w_\sigma(\mathbf{z} - \mathbf{x}^*)h_{win}, \tag{3}$$

where $I(\cdot)$ is the image intensity of the context, $w_\sigma(\mathbf{z} - \mathbf{x}^*)$ is a focus of attention function, with a scale parameter σ, defined as

$$w_\sigma(\mathbf{z} - \mathbf{x}^*) = e^{-\frac{|\mathbf{z}-\mathbf{x}^*|^2}{\sigma^2}}, \tag{4}$$

and

$$h_{win} = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi}{\tau}t), & |t| \le \frac{\tau}{2}, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Eq. 5 defines a Hamming window and it is applied in Eq. 3 to reduce the frequency influence from the image boundary on the FFT [4,21].

Substitute Eqs. 2 and 3 into Eq. 1, the confidence map of the object location $\mathbf{x}$ is defined and computed by

$$\begin{aligned} m(\mathbf{x}) &= b \cdot e^{-\left|\frac{\mathbf{x}-\mathbf{x}^*}{\alpha}\right|^\beta} \\ &= \textstyle\sum_{\mathbf{z}\in\Omega_c(\mathbf{x}^*)} h^{sc}(\mathbf{x}-\mathbf{z})I(\mathbf{z})w_\sigma(\mathbf{z}-\mathbf{x}^*)h_{win} \\ &= h^{sc}(\mathbf{x}) \bigotimes (I(\mathbf{x})w_\sigma(\mathbf{x}-\mathbf{x}^*)h_{win}), \end{aligned} \tag{6}$$

where b is a normalization constant, and α and β are a scale parameter and a shape parameter respectively. Eq. 6 is transformed to a frequency domain for fast convolution through:

$$F(b\cdot e^{-\left|\frac{\mathbf{x}-\mathbf{x}^*}{\alpha}\right|^\beta}) = F(h^{sc}(\mathbf{x})) \bigodot F(I(\mathbf{x})w_\sigma(\mathbf{x}-\mathbf{x}^*)h_{win}), \tag{7}$$

where $F(\cdot)$ denotes the Fast Fourier Transform (FFT) function and $\bigodot$ denotes the dot product. The $h^{sc}(\mathbf{x})$ is computed by

$$h^{sc}(\mathbf{x}) = F^{-1}\left(\frac{F(b\cdot e^{-\left|\frac{\mathbf{x}-\mathbf{x}^*}{\alpha}\right|^\beta})}{F(I(\mathbf{x})w_\sigma(\mathbf{x}-\mathbf{x}^*)h_{win})}\right), \tag{8}$$

where $F^{-1}(\cdot)$ denotes the inverse FFT (IFFT) function. More details can be found in [36].

2.2 Orderless and Blurred Visual Tracking

Dai et al. [5] presented a robust tracker, namely orderless and blurred tracker (OBT), which adapts to both rigid and deformable objects in the scenarios containing orderless motion and image blurs. In this paper, the RGB vector of an image is resized into 2 × 2 and the Euclidean distance is used as the similarity measurement between a candidate and a template for the preliminary screening. Then, the best target location is obtained by computing a confidence map based on the spatio-temporal context.

3 Hybrid Generative-Discriminative Tracking

3.1 Framework

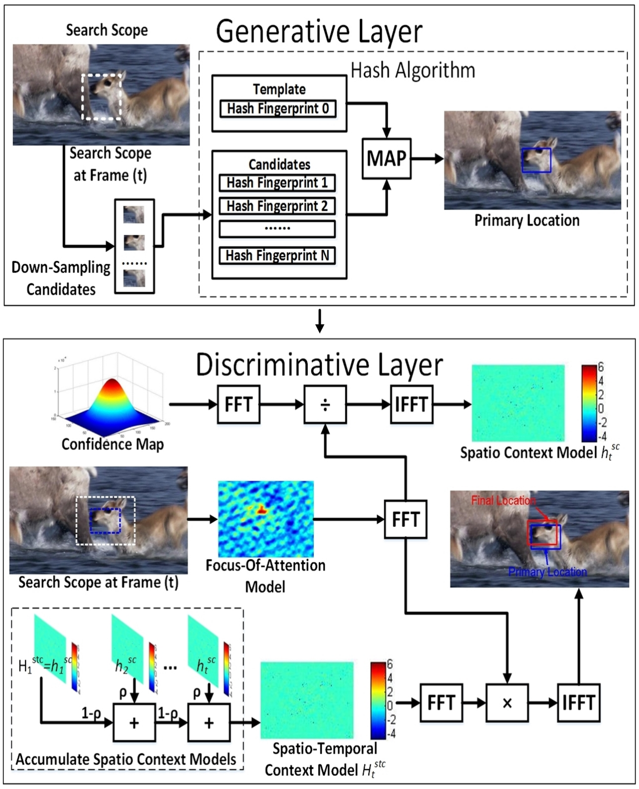


Figure 1 Framework of our tracking algorithm. The generative layer (coarse tracking) estimates the primary location (tentative object) by the Maximum a Posteriori (MAP) of Hash fingerprints. Then, the discriminative layer (fine tracking) integrates the tentative object and its surroundings to construct a spatio-temporal context model and a focus-of-attention model. The final location is inferred by maximizing the image convolution.

Figure 1 shows the basic flow of our proposed tracking algorithm. The hierarchical tracker consists of two parts as follows.

Generative Layer Assume that this process is under the framework of the Particle Filter. Firstly, we crop out a candidate set from a search scope restrained to the previous trajectory. Then, the Average Hash method is integrated into our similarity metric to help estimate the particle weights. Afterwards, the primary result is estimated by the Maximum a Posteriori (MAP).

Discriminative Layer At the location indicated in the primary result, we firstly fuse the cues of the local context region to construct a focus-of-attention model in the first frame and a spatial context model for each frame. The spatial context models constructed from the previous frames are used to form a spatio-temporal context model by a weighted accumulative addition. Then, the confidence map is obtained by the convolution of the focus-of-attention model and the spatio-temporal context model. Finally, the largest response of the confidence map is regarded as the tracking result.

3.2 Fast Tracking on the Generative Layer

In this paper, we employ the framework of Particle Filter to conduct the candidate sampling. The conventional Particle Filter described in [20,26,30,32], which is based on the Monte Carlo method, uses a Bayesian sequential importance sampling technique to approximate the posterior distribution of state variables. Meanwhile, this solution is restricted by the following factors. On one hand, to collect plentiful observations of particles is critical for doing a better posterior distribution approximation. On the other hand, to extract excessive similar samples will generate the problems of particle degeneracy and redundancy, and is time-consuming. Two important steps for avoiding particle degeneracy are choosing a proper proposal distribution and selecting a resampling algorithm. Therefore, we design a simplified Particle Filter framework to search the object based on MAP.

Instead of using massive particles to infinitely approach a real posterior distribution, we tend to use only a small number of particles for pointing out the region where the object is located. We extract the particles that are significant to form a simplified candidate set $\mathbf{x}_t$ in order to decrease the calculation load and redundancy. For doing this, we define a state set $\mathbf{x}_t$, of which each element is a pair of the x coordinate and y coordinate of a candidate's location corresponding to an object at time t:

$$\begin{aligned} &\mathbf{x}_t = \{(x,y)|(x,y) = (f_x^i, f_y^j)\}, \\ &f_x^i = x_{t-1}^{**} + i \times \frac{\Delta x_{t-1}^{**}}{2}, \\ &f_y^j = y_{t-1}^{**} + j \times \frac{\Delta y_{t-1}^{**}}{2}, \\ &i,j = 0,1,2, \end{aligned} \tag{9}$$

where (x,y) denotes the location of a candidate at time t, f_x^i and f_y^j are the functions to generate the x coordinate and the y coordinate of the candidate's location, respectively, and $(x_{t-1}^{**}, y_{t-1}^{**})$ indicates the object position at time $t-1$, and Δx_{t-1}^{**} and Δy_{t-1}^{**} represent the moving distances of an object from time $t-2$ to $t-1$ in x and y directions respectively.

Given that our goal is to gain a potential location with little computation, we use the Average Hash algorithm [17] based on the low-frequency cues to achieve the optimal similarity solution as follows.

Let $\hat{x}_t$ be the template and $\mathbf{x}_t$ be the candidate set at time t . Let $G(\hat{x}_t)$ be the template's gray image centered $\hat{x}_t$ and $\mathbf{G}(\mathbf{x}_t)$ be the set of candidate's gray images centered at the individual elements of $\mathbf{x}_t$ respectively. By doing the down sampling, we can quickly remove the high-frequency features and details, and preserve only the basic information, such as the structure and intensity of an image. Meanwhile, the color of pictures is also simplified.

Each matrix $G = [G_{ij}]_{m\times n}$ representing an gray image is converted to its corresponding Boolean matrix $B = [B_{ij}]_{m\times n}$ by the following function:

$$B_{ij} = \begin{cases} 1, & G_{ij} \geq \mu \\ 0, & \text{otherwise} \end{cases}, \tag{10}$$

where each Boolean matrix's element B_{ij} is converted from the gray matrix's element G_{ij}, and μ is the mean value of the elements in the gray matrix. Then, we sequence B to form a vector and gain a Hash fingerprint $\mathbf{K}$ by $\mathbf{K} = [B_{11}, B_{12}, \cdots, B_{mn}]$.

The relative distance between the Hash fingerprints of the template $\hat{x}_t$ and the l-th candidate x_t^l at time t, denoted by $\mathbf{K}(\hat{x}_t)$ and $\mathbf{K}(x_t^l)$ respectively, can be estimated by

$$\begin{aligned} &d(\hat{x}_t, x_t^l) = \exp(-HamDis(\mathbf{K}(\hat{x}_t), \mathbf{K}(x_t^l))), \\ &l = 1,2,\cdots,N, \end{aligned} \tag{11}$$

where $HamDis(\cdot)$ stands for the Hamming distance.

Given the l-th sample of state at time t, x_t^l and the observations up to time t, $y_{1:t}$, our method estimates the posteriori probability $p(x_t^l \mid y_{1:t})$ of state x_t^l under the framework of Particle Filter with the following formulation:

$$p(x_t^l \mid y_{1:t}) = \frac{p(y_t|x_t^l)p(x_t^l|y_{1:t-1})}{p(y_t|y_{1:t-1})}, l = 1,2,\cdots,N, \tag{12}$$

where $p(y_t|x_t^l)$ denotes the observation likelihood. The likelihood is defined as $p(y_t|x_t^l) = d(\hat{x}_t, x_t^l)$.

The posterior $p(x_t^l \mid y_{1:t})$ is approximated given a finite set of N samples $\{x_t^l\}_{l=1,\cdots,N}$ with importance weights $\{\omega_t^l\}_{l=1,\cdots,N}$. The candidate x_t^l is drawn from an importance distribution $q(x_t^l|x_{1:t-1}^l, y_{1:t})$, and then the weight of the sample is updated by

$$\omega_t^l = \omega_{t-1}^l \frac{p(y_t|x_t^l)p(x_t^l|x_{t-1}^l)}{q(x_t^l|x_{1:t-1}^l, y_{1:t})},\ l = 1, 2 \cdots, N. \tag{13}$$

In the bootstrap filter, there is $q(x_t^l|x_{1:t-1}^l, y_{1:t}) = p(x_t^l|x_{t-1}^l)$ which is equivalent to Eq. 9. Then, Eq. 13 can be calculated by

$$\begin{aligned} \omega_t^l &= \omega_{t-1}^l p(y_t|x_t^l) \\ &= \omega_{t-1}^l d(\hat{x}_t, x_t^l) \end{aligned} \quad l = 1, 2 \cdots, N. \tag{14}$$

Let

$$\omega_t = \{\omega_t^1, \omega_t^2, \cdots, \omega_t^N\}. \tag{15}$$

Then, the values of ω_t^l for $l = 1, 2, \cdots, N$ at time t are reassigned by

$$\omega_t^l = \begin{cases} 1, & \omega_t^l = \max \omega_t \\ 0, & \text{otherwise} \end{cases} \quad l = 1, 2, \cdots, N. \tag{16}$$

Eqs. 9 - 16 can reduce the particle degeneracy by abandoning the less significant particles and increasing the weights of more significant particles.

Then, the aforementioned best configuration of an object, x_t^*, can be obtained by the weighted particles over the N number of particles at each time t.

$$x_t^* = \sum_{l=1}^{N} x_t^l w_t^l \tag{17}$$

where x_t^l indicates the l-th sample of the state $\mathbf{x}_t$. For the Particle Filter, the maximum posteriori probability $p(x_t^l|y_{1:t})$ is equal to $\sum_{l=1}^{N} x_t^l w_t^l$. In this way, we can say that x_t^* is obtained by the Maximum a Posteriori (MAP) estimate

$$x_t^* = \arg\max_{x_t^l} p(x_t^l|y_{1:t}),\ l = 1, 2, \cdots, N. \tag{18}$$

3.3 Optimization on the Discriminative Layer

Let us denote the $h^{sc}(\mathbf{x})$ in Eq. 8 at time t by $h_t^{sc}(\mathbf{x})$. Let us also denote the corresponding context feature set used in deriving Eq. 8 at time t by $X_t^c = \{\mathbf{c}(\mathbf{z}) = (I_t(\mathbf{z}), \mathbf{z}|\mathbf{z} \in \Omega_c(x_t^*))\}$, where $I_t(\mathbf{z})$ represents the image intensity at location $\mathbf{z}$ and $\Omega_c(x_t^*)$ is the local context region centered at x_t^* obtained in Eq. 18 at time t. $h_t^{sc}(\mathbf{x})$ is used to update the spatio-temporal context model [36] $H_t^{stc}(\mathbf{x})$ at time t in Eq. 19 below.

$$\begin{aligned} &H_1^{stc}(\mathbf{x}) = h_1^{sc}(\mathbf{x}) \\ &H_t^{stc}(\mathbf{x}) = (1-\rho) \cdot H_{t-1}^{stc}(\mathbf{x}) + \rho \cdot h_t^{sc}(\mathbf{x}), \text{ for } t > 1, \end{aligned} \tag{19}$$

where ρ is considered as a learning parameter. The object location x_t^{**} is selected as the location of ground truth at time $t = 1$ and is calculated by maximizing the new confidence map [36] at $t > 1$:

$$x_t^{**} = \arg\max_{\mathbf{x} \in \Omega_c(x_t^*)} m_t(\mathbf{x}), \tag{20}$$

where

$$m_t(\mathbf{x}) = F^{-1}(F(H_{t-1}^{stc})(\mathbf{x}) \bigodot F(I_t(\mathbf{x}) w_\sigma(\mathbf{x} - x_t^*) h_{win})), \tag{21}$$

which is deduced from Eq. 7.

Note that we use the zero mean treatment to every frame in order to remove the effect of illumination changes.

Finally, the template in Section 3.2 is updated by $G(\hat{x}_t) = G(x_t^{**})$.

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

Algorithm 1 The proposed tracking method.

```
Input: Video frame $t = 1 : F$
for $t = 1 : F$ do
    if $t == 1$ then
        Generate the gray matrix of the template $G(\hat{x}_1)$ and
        the $h_1^{sc}$, and then construct the spatio-temporal context as $H_1^{stc} = h_1^{sc}$.
        Obtain the location $x_1^{**}$ of the tracking object from ground truth.
    else
        Generate the gray matrices of the candidate set
        $\mathbf{G}(\mathbf{x}_t) = \{G(x_t^1), G(x_t^2) \cdots, G(x_t^N)\}$.
        Construct the Hash fingerprints of $\mathbf{K}(\hat{x}_t)$ and $\mathbf{K}(x_t^l)$
        for $l = 1, 2, \cdots, N$, and obtain the primary location $x_t^*$ by the MAP.
        Construct the spatio context model $h_t^{sc}$ based on $x_t^*$.
        Compute the confidence map $m_t(\mathbf{x})$ based on $H_{t-1}^{stc}$.
        Estimate the object location $x_t^{**}$ at time $t$ by maximizing the confidence map.
        Update $G(\hat{x}_t)$ and $H_t^{stc}$.
    end if
end for
Output: Tracking results $\{x_1^{**}, x_2^{**}, \cdots, x_F^{**}\}$.
```

4 Experiments

We compare our method with 8 state-of-the-art methods: spatio-temporal context tracker (STC) [36], multiple instance learning tracker (MIL) [1], weighted MIL tracker (WMIL) [35], compressive tracker (CT) [37], L1 minimization tracker (L1) [18], L1 tracker using accelerated proximal gradient (L1-APG) [3], local orderless tracker (LOT) [22] and orderless and blurred tracker (OBT) [5].

Our comparison is done on the Visual Tracker Benchmark [28], particularly focusing on the video sequences with the impact factors including illumination variation, motion blur, fast motion and cluttered scene. Table 1 shows the details of the evaluated video sequences. For some trackers involving randomness, we repeat the experiments five times on each video and get the averaged results.

Table 1 Evaluated video sequences. '√' denotes that the sequence contains the corresponding challenge, and '×' implies that the challenge is excluded.

			Main Challenges			
Sequence	Object Size	Frames	Illumination variation	Motion Blur	Fast Motion	Cluttered Scene
Body	87*319	334	×	√	×	×
Car2	122*99	585	×	√	×	√
Car4	170*149	380	×	√	×	√
Face	94*114	493	×	√	×	×
Deer	95*65	71	×	×	√	√
David	51*54	761	√	×	×	×
Shaking	61*71	365	√	×	×	√
Bike	67*56	228	×	×	√	×

4.1 Parameter settings

The proposed method has several adjustable parameters. In the process of spatio-temporal context, the parameters of the map function are set to $\alpha = 1.8$ and $\beta = 1$. The learning parameter is set to $\rho = 0.086$. Here, β and ρ are set to the same values as those in [36]. α is a scale parameter as found in Eq. 6 for computation of $m(\mathbf{x})$. The greater α is, the bigger weight is given to each $\mathbf{x}$ further away to the object centre in computing $m(\mathbf{x})$. Noting that the focus of our work is on tracking with motion blur, the information near the indistinct outlier is less reliable than that close to the object center. Therefore, it is a good idea to assign a small weight to each $\mathbf{x}$ near the outlier by setting a small α. Instead of using the fixed $\alpha = 2.25$ as shown in [36], we test our results using various values of α in [1,3] with the increment of 0.05 in this paper. The average CLE and average DP are chosen as the evaluation criteria.

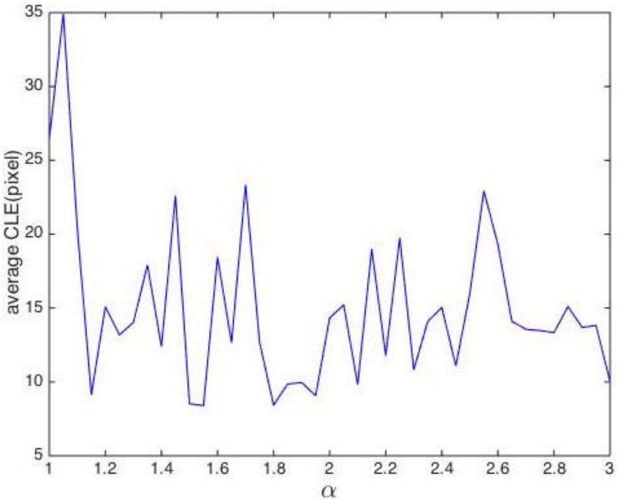


Figure 2 The means of average CLE plots of all tested sequences with various values of parameter $\alpha \in [1, 3]$.

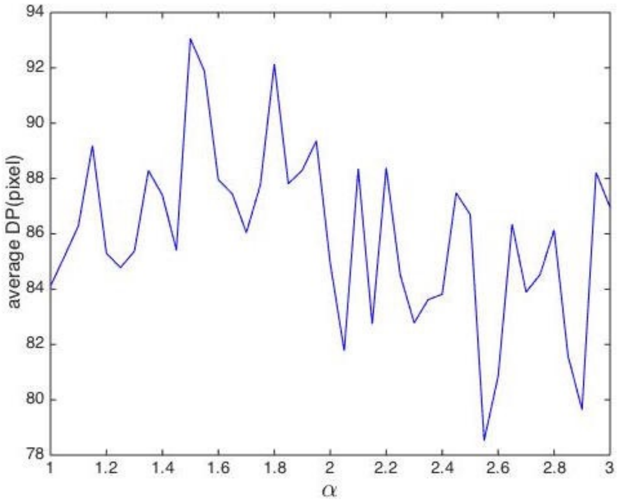


Figure 3 The means of average DP plots of all tested sequences with various values of parameter $\alpha \in [1, 3]$.

As shown in Figures 2-3 and Table 2, we get the second best (slightly worse than the best) average results over all eight videos in terms of both CLE and DP when $\alpha = 1.8$. Although the average DP over the eight videos reaches to its maximum when $\alpha = 1.5$, the average CLE result is only the third best for the same α value. Similarly, although the average CLE over the eight videos reaches to its maximum when $\alpha = 1.55$, the average DP result is only the third best for the same α value. Therefore, by taking into account both CLE and DP results, we decide to set $\alpha = 1.8$ in this paper. The full set of values showing the average DP and CLE results over all eight videos is displayed in Table 2. With the selection of this parameter value, our tracker achieves relatively lower average CLEs and relatively higher average DPs over the eight videos. Note that,

Table 2 The detailed data of the means of average CLEs and the means of average DPs of 8 video sequences. We highlight the results in $\alpha = 1.80$ and $\alpha = 2.25$.

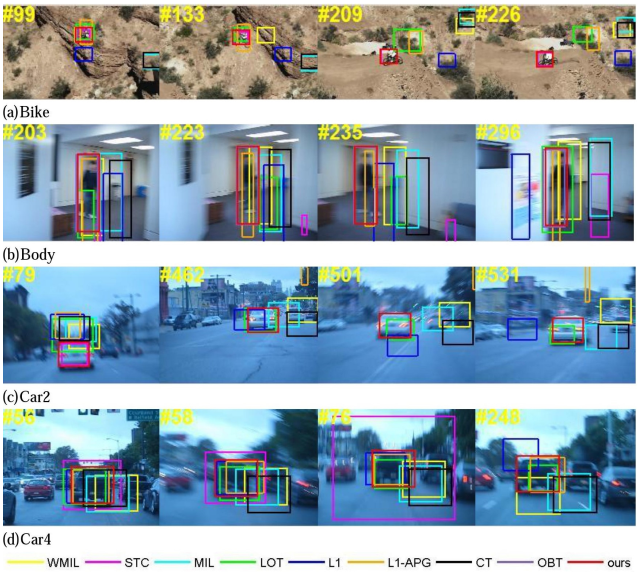
α	1.00	1.05	1.10	1.15	1.20	1.25	1.30	1.35	1.40	1.45	1.50	1.55	1.60	1.65
CLE	26.46	34.92	20.39	9.13	15.10	13.18	14.03	17.92	12.40	22.60	8.52	8.39	18.45	12.66
DP	84.11	85.19	86.29	89.19	85.29	84.78	85.37	88.29	87.40	85.41	93.06	91.88	87.95	87.44
α	1.70	1.75	**1.80**	1.85	1.90	1.95	2.00	2.05	2.10	2.15	2.20	**2.25**	2.30	2.35
CLE	23.34	12.68	**8.42**	9.86	9.95	9.07	14.34	15.21	9.82	18.99	11.79	**19.75**	10.82	14.11
DP	86.04	87.76	**92.14**	87.81	88.30	89.36	84.92	81.77	88.35	82.74	88.38	**84.51**	82.78	83.62
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	
CLE	15.04	11.09	15.98	22.93	19.30	14.10	13.56	13.48	13.34	15.10	13.69	13.82	10.04	
DP	83.81	87.48	86.70	78.53	80.85	86.35	83.89	84.53	86.13	81.58	79.64	88.22	86.95	

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

even when $\alpha = 2.25$ (the value used in STC [36]), our tracker achieves the average CLE of 19.75 and the average DP of 84.51 (shown in Table 2), which are still significantly better than the STC's average CLE of 88.43 and average DP of 67.49 (shown in Tables 3 and 4 respectively).

4.2 Qualitative analysis

Figures 4 - 5 visually demonstrate some tracking results using different tracking methods.



(a)Bike

(b)Body

(c)Car2

(d)Car4

Figure 4 Comparison of our approach with state-of-the-art trackers on videos Bike, Body, Car2 and Car4.

Illumination variation: For ***David*** sequence (Figure 5(e)), most of the existing methods fail to track on a frame (e.g., #134), where the target is in a very dark area. The tracking results on frame #134 demonstrate that OBT, STC and our method perform relatively well while the other methods (e.g., WMIL, MIL, LOT, L1, L1-APG and CT) completely fail to track the objects. These results are attributed to that our tracker and STC use the zero mean treatment (as mentioned in Section 3.3) to reduce the influence of uneven illumination. On frames #339, #438 and #666, OBT, STC and our method show their superiority over other methods because they apply updated cues.

In ***Shaking*** sequence (Figure 5(h)), the dramatic variation of the stage light makes the tracking even harder. OBT, STC and our method use the spatio-temporal context models, so they can discover the relevance between object appearance and candidate samples. Therefore, OBT, STC and our method also outperform other trackers when an video experience significant illumination and appearance variations.

Motion Blur: The proposed method is robust to motion blur as shown in ***Body*** (Figure 4(b)), ***Car2*** (Figure 4(c)), ***Car4*** (Figure 4(d)) and ***Face*** (Figure 5(g)) sequences. Other methods suffer from severe drift and even fail to track. This robustness is attributed to the hierarchical structure which has both generative and discriminative merits. The generative method can detect the most similar patch to a target and the discriminative method uses the spatial relationships and appearances of local contexts to separate the target from its background. Furthermore, the Hash fingerprint introduced in the generative appearance model can effectively remove the complex information (i.e., high-frequency information) and preserve the basic information (i.e., low-frequency information). Therefore, our generative method under the Particle Filter framework can estimate similarities between the template model and candidate samples in a fast and accurate way.

Fast Motion: It is difficult to capture a target which is undergoing a random and fast motion.

In ***Bike*** sequence (Figure 4(a)), the proposed method and STC are more appropriate than other methods for foreground segmentation from background. This is because that the two methods can reveal the relevance between an object and its contextual cues while other methods cannot. Besides, the tracking will be more difficult when some analogues appear in a scene.

As shown in ***Deer*** sequence (Figure 5(f)), only our tracker performs well throughout the whole sequence while other trackers fail to complete the tracking task.

Table 3 Center location error (CLE) (in pixels). The best results are shown in red while the second and third ones are shown in blue and green.

Sequence	WMIL[35]	STC[36]	MIL[1]	LOT[22]	L1[18]	L1-APG[3]	CT[37]	OBT[5]	OURS
Body	54.4	148	128	84.5	131	**36.7**	122	**18.1**	**16.8**
Car2	163	**5.41**	73.9	26.2	49.9	213	104	**5.14**	**5.02**
Car4	101	**18.2**	146	25.3	61.6	20.5	161	**16.2**	**15.62**
Face	127	113	123	**33.4**	149	91.9	55.8	**3.91**	**3.86**
Deer	**15.6**	401	202	63.7	78.1	214	211	**5.43**	**5.41**
David	36.3	**6.61**	43.8	103	63.4	67.1	44.9	**5.45**	**5.41**
Shaking	12	**8.2**	145	73.6	29.1	23.7	11.2	**10.7**	**8.09**
Bike	120	**7.03**	217	24.1	136	26.5	216	**7.35**	**7.15**
Average CLE	78.66	88.43	134.84	**54.23**	87.26	86.68	115.74	**9.04**	**8.42**

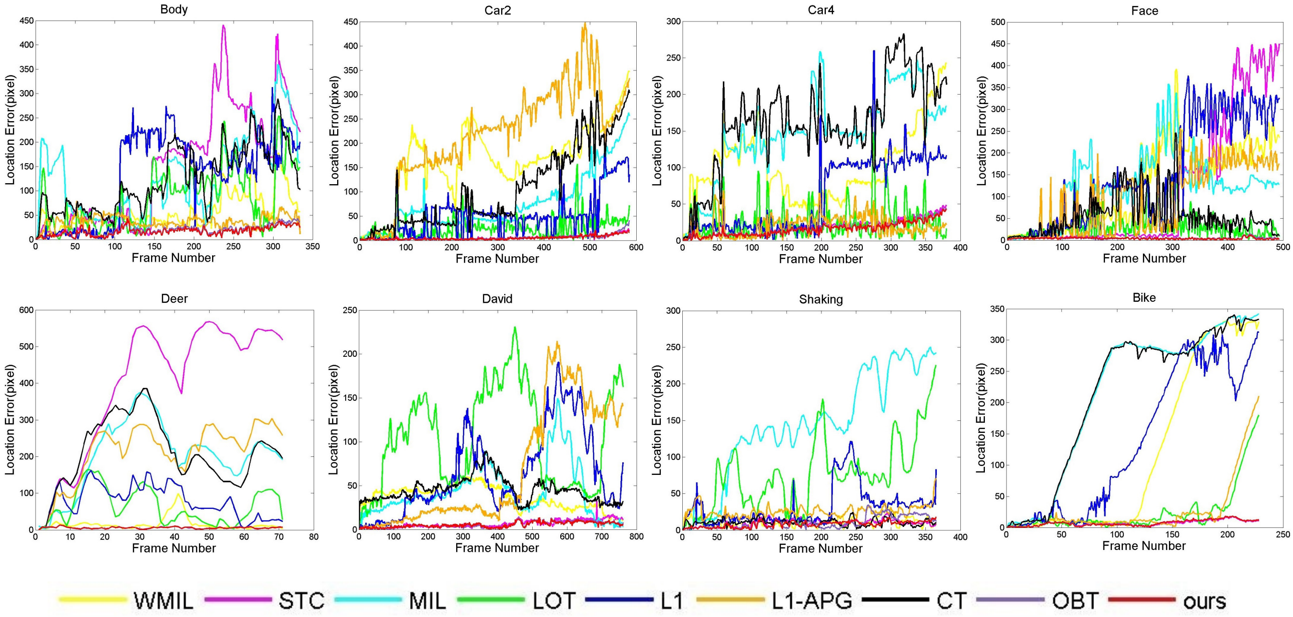


Figure 6 Error plots of all tested sequences for different tracking methods.

Table 4 Distance precision (DP) (in pixels). The best results are shown in red while the second and third ones are shown in blue and green.

Sequence	WMIL[35]	STC[36]	MIL[1]	LOT[22]	L1[18]	L1-APG[3]	CT[37]	OBT[5]	OURS
Body	16.2	16.5	2.99	14.1	**27.2**	7.49	1.2	**55.4**	**66.2**
Car2	7.69	**99**	15.9	43.9	30.4	12	7.52	**95.7**	**99.3**
Car4	3.42	**57.4**	3.42	52.4	33.2	50.3	3.68	**68.9**	**71.6**
Face	18.1	**62.9**	15.4	**39.4**	12.4	30.8	19.9	**100**	**100**
Deer	**87.3**	4.23	5.63	**19.7**	9.86	4.23	4.23	**100**	**100**
David	10.1	**99.9**	16.6	1.97	17.2	**31.5**	0.131	**100**	**100**
Shaking	**83.8**	**100**	12.3	17	46.3	26	**94.5**	72.3	**100**
Bike	52.2	**100**	17.1	**71.9**	26.8	**72.4**	17.1	**100**	**100**
Average DP	34.85	**67.49**	11.17	32.55	25.42	29.34	18.53	**86.54**	**92.14**

Although WMIL is also robust under the situations that the surroundings are similar to the initial state, it lost the target when there are sharply changing surroundings as shown in frames #8 and #40. In fact, our success is attributed to the coarse-to-fine structure. Within a search scope, we firstly extract an image patch which is most likely to contain a target. Then, we attempt to find the exact location of the object within the extracted patch and its surroundings. This strategy makes our method search a large scope efficiently while other methods are restricted to their small search scopes.

Cluttered Scene: Similar foreground and background can cause confusion during tracking. As seen in ***Car4*** sequence (Figure 4(d)), there are many similar cars passing by and the image blur caused by the shaking camera also increases the tracking difficulty. Given that our tracker removes the high-frequency information (e.g., contours) by down sampling, it can avoid the influence of fuzzy boundaries. In addition, our tracker predicts the location of an object with the help of relationships between the object and its surroundings. Therefore, our method can succeed in tracking while

Table 5 Comparison with average frames per second (FPS). The best results are shown in red while the second and third ones are shown in blue and green.

Sequence	WMIL[35]	STC[36]	MIL[1]	LOT[22]	L1[18]	L1-APG[3]	CT[37]	OBT[5]	OURS
Body	**20.68**	**20.63**	0.87	0.3	1.82	15.03	12.92	18.35	**21.22**
Car2	19.23	**22.22**	1.42	0.41	1.94	8.43	12.47	**27.66**	**25.76**
Car4	20.2	**20.21**	1.24	0.3	2.85	15.12	12.65	**22.04**	**22.25**
Face	20.52	**22.52**	1.02	0.1	2.61	9.51	12.69	**28.02**	**26.41**
Deer	19	**26.07**	3.26	0.1	2.67	6.63	13.55	**29.36**	**26.42**
David	31.47	**32.54**	1.73	1.11	3.1	10.21	16.85	**53.29**	**41.77**
Shaking	20.87	**30.86**	1.59	0.45	2.79	8.44	15.76	**37.72**	**31.8**
Bike	20.86	**30.31**	2.09	0.58	1.47	8.1	14.2	**37.48**	**33.69**
Average FPS	21.6	**25.67**	1.65	0.42	2.41	10.18	13.89	**31.74**	**28.67**

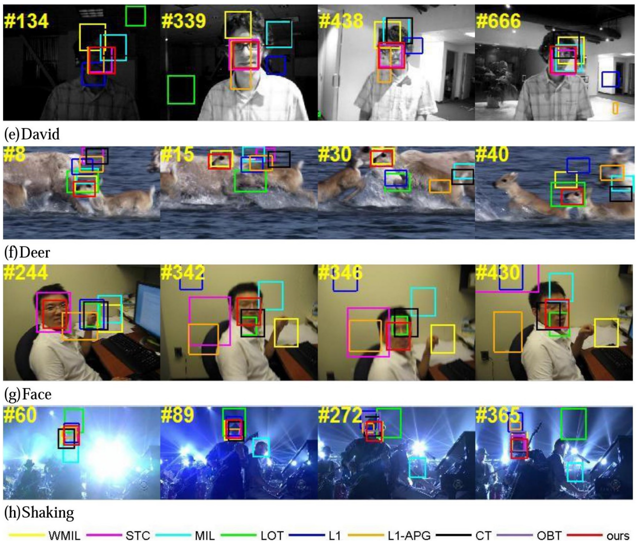


Figure 5 Comparison of our approach with state-of-the-art trackers on videos David, Deer, Face and Shaking.

other trackers undergo severe drifts and tracking failures.

In ***Car2*** (Figure 4(c)), ***Deer*** (Figure 5(f)) and ***Shaking*** (Figure 5(h)) sequences, our tracker also shows its superiority compared with other methods.

4.3 Quantitative analysis

Here, we use two evaluation criteria as introduced in [28] for experimental comparison: center location error (CLE) and distance precision (DP). The speed performance is evaluated by frames per second (FPS). CLE is calculated based on the average Euclidean Distance between an object's center and its ground-truth. The pixel error in each frame is defined as:

$$CLE = \sqrt{(x_{ob} - x_{gt})^2 + (y_{ob} - y_{gt})^2}, \tag{22}$$

where (x_{ob}, y_{ob}) is the object location in each frame, and (x_{gt}, y_{gt}) is the ground truth of each frame.

Besides CLE, we also compute the precision rate DP, which embodies the correlative number of frames where CLEs are below a certain threshold. Here, we set the DP values at the threshold of 20 pixels [28]. The DP score in a sequence is calculated by:

$$DP = \frac{Num(CLE < \tau)}{N}, \tag{23}$$

where τ is the DP threshold, and $Num(\cdot)$ is the function to accumulate the total number of frames where CLEs are smaller than τ. The denominator N is the number of frames in a full sequence.

Table 3 and Figure 6 report the center location error and smaller CLE reflects better performance. In Table 3, each row shows the average CLE of the different methods tested on a certain video sequence. The best results are shown in red while the second and third ones are shown in blue and green. Table 4 reports the DP which records the success frame number of a sequence. Larger DP means more accurate results. As seen in Tables 3 - 4 and Figure 6, our tracker achieves the best performance in ***Body***, ***Car2***, ***Car4***, ***Face***, ***Deer***, ***David*** and ***Shaking***, when compared with WMIL, STC, MIL, LOT, L1, L1-APG and CT. For ***Bike*** sequence, in spite that our CLE is a little greater than the counterpart of STC in Table 3, our CLE curve (shown in Figure 6) almost overlaps the curve of STC and hence demonstrates almost equal performance to STC. Our slightly poorer performance in terms of CLE compared with STC is because the object's center is not estimated accurately enough at the generative stage so that some parts of the target are excluded from our searching scope in the discriminative stage. The main reason for the above to happen is the low sampling density that occurred when down sampling was performed to reduce the computation complexity of the Particle Filter. Nevertheless, the results in terms of DP verify the outstanding performance and superiority of our approach.

Furthermore, we also use the average frames per second (FPS) (see Table 5) to evaluate the speed of each method. Actually, speed is a crucial factor for many real-world applications. In our method, we use the down sampling and FFT transformation for rapid calculation. Implemented in MATLAB, our tracker runs at 28.67 PFS on average on an i7 2.80 GHz CPU with 16 GB

RAM. PFS results show the suitability of our method for real-time applications.

To summary, Tables 3 - 5 show that our method performs excellently in terms of both speed and accuracy on 8 challenging sequences. Our approach outperforms the 8 state-of-the-art methods in terms of average CLE and average DP, and is second best in terms of average FPS.

5 Conclusions and future work

In this paper, a real-time method which is named hybrid generative-discriminative Hash tracker (HGDHT) has been proposed. Firstly, the particles representing the potential centers of a target are generated in the generative stage. The Hash fingerprint matching is applied to formulate the observation likelihood. Then, the preliminary location of the target is estimated by an improved Maximum a Posteriori (MAP). This preprocess has also extended the search scope in order to capture the object which undergos a random and fast motion. In the discriminative stage, we optimize a confidence map derived using spatio-temporal context to find the accurate target location. As a consequence, our method is robust to appearance variations. Experiments on some challenging video sequences have demonstrated the superiority of the proposed approach over 8 existing state-of-the-art methods in terms of both accuracy and robustness.

In the future, we will improve the scale adaptability of our tracker so that extracted rounding box containing a target can be more precise.

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

References

1. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 983–990. IEEE (2009)
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. Bolme, D.S., Draper, B.A., Beveridge, J.R.: Average of synthetic exact filters. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 2105–2112. IEEE (2009)
5. 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)
6. 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)
7. Dewan, M.A.A., Granger, E., Marcialis, G.L., Sabourin, R., Roli, F.: Adaptive appearance model tracking for still-to-video face recognition. Pattern Recognition **49**, 129–151 (2016)
8. Duffner, S., Garcia, C.: Exploiting contextual motion cues for visual object tracking. In: Computer Vision-ECCV 2014 Workshops, pp. 232–243. Springer (2014)
9. 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)
10. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. Pattern Analysis and Machine Intelligence, IEEE Transactions on **37**(3), 583–596 (2015)
11. Hong, S., Kwak, S., Han, B.: Orderless tracking through model-averaged posterior estimation. In: Computer Vision (ICCV), 2013 IEEE International Conference on, pp. 2296–2303. IEEE (2013)
12. Jang, S.I., Choi, K., Toh, K.A., Teoh, A.B.J., Kim, J.: Object tracking based on an online learning network with total error rate minimization. Pattern Recognition **48**(1), 126–139 (2015)
13. 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)
14. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 1269–1276. IEEE (2010)
15. Kwon, J., Lee, K.M.: Tracking by sampling trackers. In: Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 1195–1202. IEEE (2011)
16. Liu, L., Liu, Y.J., Li, D.J.: Intelligence computation based on adaptive tracking design for a class of non-linear discrete-time systems. Neural Computing and Applications **23**(5), 1351–1357 (2013)
17. Ma, C., Liu, C., Peng, F., Liu, J.: Multi-feature hashing tracking. Pattern Recognition Letters **69**, 62–71 (2016)
18. Mei, X., Ling, H.: Robust visual tracking using l1 minimization. In: Computer Vision, 2009 IEEE 12th International Conference on, pp. 1436–1443. IEEE (2009)
19. Morais, E., Ferreira, A., Cunha, S.A., Barros, R.M., Rocha, A., Goldenstein, S.: A multiple camera methodology for automatic localization and tracking of futsal players. Pattern Recognition Letters **39**, 21–30 (2014)
20. Najim, K., Ikonen, E., Del Moral, P.: Open-loop regulation and tracking control based on a genealogical decision tree. Neural Computing & Applications **15**(3-4), 339–349 (2006)
21. Oppenheim, A.V., Willsky, A.S., Nawab, S.H.: Signals and systems, vol. 2. Prentice-Hall Englewood Cliffs, NJ **6**(7), 10 (1983)
22. 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)

23. Pan, C., Lai, X., Yang, S.X., Wu, M.: A bioinspired neural dynamics-based approach to tracking control of autonomous surface vehicles subject to unknown ocean currents. Neural Computing and Applications **26**(8), 1929–1938 (2015)
24. Peleg, S., Werman, M., Rom, H.: A unified approach to the change of resolution: Space and gray-level. Pattern Analysis and Machine Intelligence, IEEE Transactions on **11**(7), 739–742 (1989)
25. 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)
26. Su, Y., Zhao, Q., Zhao, L., Gu, D.: Abrupt motion tracking using a visual saliency embedded particle filter. Pattern Recognition **47**(5), 1826–1834 (2014)
27. Wu, J., Su, B., Li, J., Zhang, X., Ai, L.: Global adaptive neural tracking control of nonlinear mimo systems. Neural Computing and Applications pp. 1–13 (2016)
28. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2013)
29. Xiao, J., Stolkin, R., Leonardis, A.: Multi-target tracking in team-sports videos via multi-level context-conditioned latent behaviour models. In: Proceedings of the British Machine Vision Conference. BMVA Press (2014)
30. Yi, S., He, Z., You, X., Cheung, Y.M.: Single object tracking via robust combination of particle filter and sparse representation. Signal Processing **110**, 178–187 (2015)
31. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. Acm computing surveys (CSUR) **38**(4), 13 (2006)
32. Yu, G., Hu, Z., Lu, H., Li, W.: Robust object tracking with occlusion handle. Neural Computing and Applications **20**(7), 1027–1034 (2011)
33. Yu, Q., Dinh, T.B., Medioni, G.: Online tracking and reacquisition using co-trained generative and discriminative trackers. In: Computer Vision–ECCV 2008, pp. 678–691. Springer (2008)
34. Yu, T., Wu, Y.: Collaborative tracking of multiple targets. In: Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, vol. 1, pp. I–834. IEEE (2004)
35. Zhang, K., Song, H.: Real-time visual tracking via online weighted multiple instance learning. Pattern Recognition **46**(1), 397–411 (2013)
36. 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)
37. Zhang, K., Zhang, L., Yang, M.H.: Real-time compressive tracking. In: Computer Vision–ECCV 2012, pp. 864–877. Springer (2012)
38. Zhang, T., Jia, K., Xu, C., Ma, Y., Ahuja, N.: Partial occlusion handling for visual tracking via robust part matching. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pp. 1258–1265. IEEE (2014)
39. Zhong, W., Lu, H., Yang, M.H.: Robust object tracking via sparsity-based collaborative model. In: Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on, pp. 1838–1845. IEEE (2012)
40. Zhou, H., Fei, M., Sadka, A., Zhang, Y., Li, X.: Adaptive fusion of particle filtering and spatio-temporal motion energy for human tracking. Pattern Recognition **47**(11), 3552–3567 (2014)