Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 4649358, 12 pages
https://doi.org/10.1155/2021/4649358

Research Article

A Real-Time Structured Output Tracker with Scale Adaption for Visual Target Tracking

Kaiyun Yang, Xuedong Wu, and Jingxiang Xu

School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu 212003, China

Correspondence should be addressed to Xuedong Wu; woolcn@163.com

Received 12 August 2020; Revised 20 January 2021; Accepted 29 January 2021; Published 10 February 2021

Academic Editor: Yingkun Hou

The structured output tracking algorithm is a visual target tracking algorithm with excellent comprehensive performance in recent years. However, the algorithm classifier will produce error information and result in target loss or tracking failure when the target is occluded or the scale changes in the process of tracking. In this work, a real-time structured output tracker with scale adaption is proposed: (1) the target position prediction is added in the process of target tracking to improve the real-time tracking performance; (2) the adaptive scheme of target scale discrimination is proposed in the structured support to improve the overall tracking accuracy; and (3) the Kalman filter is used to solve the occlusion problem of continuous tracking. Extensive evaluations on the OTB-2015 benchmark dataset with 100 sequences have shown that the proposed tracking algorithm can run at a highly efficient speed of 84 fps and perform favorably against other tracking algorithms.

1. Introduction

Visual target tracking is an important task in computer vision, which has been widely used in various fields, such as intelligent transportation systems, military, medical, and so on [1–3]. Many scholars have done a lot of research on target tracking and have achieved great progress. However, the tracking target faces many challenges such as target scale changes, occlusion, and illumination changes [4, 5]. Therefore, ensuring the accuracy of the tracking process and improving the real-time performance are of great theoretical significance [6, 7].

The Struck tracker is an algorithm based on discriminant classifier, and it has excellent adaptability to various complex backgrounds in visual target tracking [8, 9]. The Struck tracker is to construct a structured output support vector machine classifier through an online learning method. In the tracking process, the algorithm first samples the local area of target position in the previous frame and then takes the maximum value of classifier discriminant function as the current tracking result [10]. In the process of updating classifier, the classifier abandons the sample labeling process of classifier. Therefore, in the case of occlusion and scale variation, the Struck tracker cannot track target correctly [11, 12]. This paper suggests a structured output SVM tracker with real-time and occlusion detection capabilities. This work includes the following. (1) The target position prediction process is introduced. The sparse sample is used to calculate the rough position of target from the discriminator, which reduces the amount of calculation to determine the target position and improves the real-time performance. (2) A multiscale sampling adaptive scale tracking strategy is proposed. The scale discriminator calculates the scale change of target, which can adjust the size of scale adaptively. (3) The discriminant function of SVM classifier is used to implement occlusion detection. The SVM classifier will stop updating when a certain percentage of occlusion is detected, and the Kalman filter is applied to predict target position when the target has certain motion information, so the target can be successfully tracked.

2. Related Works

With the development of target tracking, many visual target tracking algorithms have been investigated [13]. The mainstream target tracking algorithms include traditional tracking methods, correlation filtering-based tracking methods, and tracking methods based on deep learning [14, 15].

2.1. Traditional Tracking Methods. Caulfield and Dawson-Howe[16] proposed a FragTrack tracking method by comparing template histograms with sub-block histograms to obtain the possible locations of target. Wang and Li [17] suggested a mean shift algorithm which used color as feature to obtain the probability density map of overall image. Wu et al. [18] introduced a multi-instance learning method to solve target drift and improve tracking accuracy. Babenko et al. [19] provided a compressive tracking algorithm to deal with the problem of occlusions and image noise. Chunxiao et al. [20] developed a tracking-learning-detection method combining traditional tracking algorithms with detection algorithms to solve the target occlusions. Hare et al. [21] presented a structured output SVM tracker which was robust to occlusions of tracking target.

2.2. Correlation Filtering-Based Tracking Methods. Bolme et al. [22] used a minimum output sum of squared error filter, which could improve tracking accuracy compared with many traditional algorithms. Henriques et al. [23] developed a circulant structure kernel tracker by building a cyclic matrix to solve the ridge regression problem. Henriques et al. [24] proposed a kernel correlation filter which had excellent performance in tracking accuracy, but it could not solve illumination and scale changes. Zhang and Zheng [25] presented a spatiotemporal context tracking algorithm which had achieved excellent real-time tracking performance. Li and Zhu [26] suggested a scale adaptive multi-feature tracking algorithm for solving the scale changes of tracking target.

2.3. Tracking Methods Based on Deep Learning. Held et al [27] presented a GOTURN tracking method and applied the end-to-end deep learning model to target tracking for the first time. Nam and Han [28] developed an MDNet tracking algorithm and updated the model online to adapt to the changing targets and scenarios. Danelljan [29] proposed a C-COT tracking algorithm, and it used the deep neural network to extract target features and solved target scale changes. Danelljan [30] proposed an effective convolution operator tracking algorithm, which solved the problem of too large C-COT model and improved the real-time performance.

Although these research methods have some effect in dealing with complex background, there are two shortcomings which can be summarized as follows: (1) these algorithms cannot judge the current occlusion state and introduce error information into the classifier when the target is occluded; (2) the performance of the algorithm is significantly reduced when the target scale changes. This work uses the discriminant function value of SVM classifier to achieve occlusion detection by combining the framework of Struck tracker and the motion information of target. The classifier stops updating when partial occlusion is detected, and the Kalman filter is used to predict the target position when the target has certain motion information. Aiming at the change of target scale, a multiscale sampling strategy is proposed, and the optimal scale is calculated with scale discriminator.

3. Proposed Method

First, the Struck tracker and a real-time structured output tracker with scaleadaption and then the whole details of the presented algorithm are given.

3.1. The Struck Tracker. During the tracking process, let $\mathbf{P}_{t-1}$ be the estimated bounding box at time $t-1$. The Struck algorithm estimates the target displacement $\mathbf{y}_t \in \mathbf{Y}$, where $\mathbf{Y}$ is the search space which is given as $\mathbf{Y} = \{(u, v) | u^2 + v^2 < r^2\}$ (r is the search radius, and are two-dimensional space coordinates). The current frame target position $\mathbf{P}_t$ which is expressed as $\mathbf{P}_t = \mathbf{P}_{t-1} \circ \mathbf{y}_t$ can be obtained by shifting the previous frame position.

The prediction function $f: \mathbf{X} \longrightarrow \mathbf{Y}$ is constructed to estimate target transformations between frames, and the output space is the space of all transformations $\mathbf{Y}$ instead of the binary labels. Now, we introduce a discriminant function to predict the target position between frames as follows:

$$\mathbf{y}_t = f\left(\mathbf{x}_t^{P_{t-1}}\right) = \arg\max_{\mathbf{y} \in Y} F\left(\mathbf{x}_t^{P_{t-1}}, \mathbf{y}\right), \tag{1}$$

which evaluates the similarity between sample and target ($\mathbf{w}$ is the coefficient vector and Φ is the kernel function that maps the input space to the feature space). In the given sample set $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$ ($n \geq 1$), for finding the optimal hyperplane, the optimization objective function is

$$\begin{aligned} \min_{w} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i, \\ \text{s.t.} \quad & \forall_i, \forall_{y \neq y_i}: \langle \mathbf{w}, \delta\Phi_i(y) \rangle \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i, \end{aligned} \tag{2}$$

where $\Delta(\mathbf{y}_i, \mathbf{y})$ is the loss function and C is the regularization parameter. This loss function should decrease towards 00 as $\mathbf{y}$ and $\overline{\mathbf{y}}$ become more similar and is $\Delta(\mathbf{y}_i, \overline{\mathbf{y}}) = 0$ only on the condition that $\mathbf{y} = \overline{\mathbf{y}}$. We now choose the loss function based on the overlap of target bounding boxes as

$$\Delta(\mathbf{y}, \overline{\mathbf{y}}) = 1 - s^o_{\mathbf{P}_t}(\mathbf{y}, \overline{\mathbf{y}}), \tag{3}$$

where $s^o_{\mathbf{P}_t}(\mathbf{y}, \overline{\mathbf{y}})$ is the overlap between target bounding boxes. Using the standard Lagrangian duality techniques, the solution of equation (2) can be converted into its equivalent dual form:

$$\begin{aligned}\max_{\boldsymbol{\alpha}} \quad & \sum_{i,\mathbf{y}\neq\mathbf{y}_i} \Delta(\mathbf{y},\mathbf{y}_i)\alpha_i^{\mathbf{y}} - \frac{1}{2}\sum_{\substack{i,\mathbf{y}\neq\mathbf{y}_i\\ j,\bar{\mathbf{y}}\neq\mathbf{y}_i}} \alpha_i^{\mathbf{y}}\alpha_j^{\bar{\mathbf{y}}}\langle\delta\boldsymbol{\Phi}_i(\mathbf{y}),\delta\boldsymbol{\Phi}_j(\bar{\mathbf{y}})\rangle,\\ & \forall_i,\forall\mathbf{y}\neq\mathbf{y}_i\colon\ \alpha_i^{\mathbf{y}}\geq 0,\\ \text{s.t.}\quad & \forall_i\colon\ \sum_{\mathbf{y}\neq\mathbf{y}_i}\alpha_i^{\mathbf{y}}\leq C,\end{aligned} \tag{4}$$

replacing the parameter into equation (4) as follows:

$$\beta_i^{\mathbf{y}} = \begin{cases} -\alpha_i^{\mathbf{y}}, & y\neq y_i,\\ \sum_{\bar{\mathbf{y}}\neq\mathbf{y}_i}\alpha_i^{\bar{\mathbf{y}}}, & \text{otherwise.}\end{cases} \tag{5}$$

According to equation (5), equation (4) can be written as

$$\begin{aligned}\max_{\boldsymbol{\beta}} \quad & -\sum_{i,\mathbf{y}}\Delta(\mathbf{y},\mathbf{y}_i)\beta_i^{\mathbf{y}} - \frac{1}{2}\sum_{i,\mathbf{y},j,\bar{\mathbf{y}}}\beta_i^{\mathbf{y}}\beta_j^{\bar{\mathbf{y}}}\langle\boldsymbol{\Phi}(\mathbf{x}_i,\mathbf{y})\Phi\left(\mathbf{x}_j,\bar{\mathbf{y}}\right)\rangle,\\ & \forall_i,\forall_{\mathbf{y}}\colon\ \beta_i^{\mathbf{y}}\leq\delta(\mathbf{y},\mathbf{y}_i)C,\\ \text{s.t.}\quad & \forall_i\colon\ \sum_{\mathbf{y}}\beta_i^{\mathbf{y}} = 0,\end{aligned} \tag{6}$$

where

$$\delta(\mathbf{y},\bar{\mathbf{y}}) = \begin{cases} 1, & (y=\bar{\mathbf{y}}),\\ 0, & (\text{otherwise}).\end{cases} \tag{7}$$

Then, the discriminant function can be simplified as

$$F(\mathbf{x},\mathbf{y}) = \sum_{i,\bar{\mathbf{y}}}\beta_j^{\bar{\mathbf{y}}}\langle\boldsymbol{\Phi}(\mathbf{x}_i,\mathbf{y}),\boldsymbol{\Phi}\left(\mathbf{x}_j,\bar{\mathbf{y}}\right)\rangle. \tag{8}$$

We refer the sample $(\mathbf{x_i},\mathbf{y})$ with $\beta_i^{\mathbf{y}}\neq 0$ as support vector. For a given support pattern $\mathbf{x}_i$, only the support vector $(\mathbf{x}_i,\mathbf{y}_i)$ is $\beta_i^{\mathbf{y}}>0$ and any other support vectors $(\mathbf{x}_i,\mathbf{y})$ with $\mathbf{y}\neq\mathbf{y_i}$ is $\beta_i^{\mathbf{y}}<0$. We refer to these as positive and negative support vectors, respectively. The selection of support vector is controlled by the following gradient:

$$\begin{aligned} h_i(\mathbf{y}) &= -\Delta(\mathbf{y},\mathbf{y}_i) - \sum_{j,\bar{\mathbf{y}}}\beta_j^{\bar{\mathbf{y}}}\langle\boldsymbol{\Phi}(\mathbf{x}_i,\mathbf{y}),\boldsymbol{\Phi}\left(\mathbf{x}_j,\bar{\mathbf{y}}\right)\rangle\\ &= -\Delta(\mathbf{y},\mathbf{y}_i) - F(\mathbf{x}_i,\mathbf{y}).\end{aligned} \tag{9}$$

It can be known from $\Delta(\mathbf{y},\mathbf{y}_i)$ that the gradient calculation includes the overlap between the sample and the target bounding box, and the algorithm updates $\beta_i^{\mathbf{y}}$ and gradient h_i incrementally during the update process of each frame to implement the classifier learning and the update of the algorithm.

3.2. The Real-Time Structured Output Tracker with Scale Adaption. The output of the Struck tracker is only the target position, and the accuracy of target position decreases when the scale changes, so we suggest a real-time structured output model with scale adaption. By taking the target tracking frame $\mathbf{x}$ of the previous frame, the candidate possible target position set $\mathbf{Y}$ in the current frame, and the candidate target scale set $\mathbf{S}$ in the current frame as input and taking the exact position $\mathbf{y}$ and the target scale in the current frame as output, the model can be represented using the following decision function:

$$g(\mathbf{y},\mathbf{s}) = \arg\max_{(\mathbf{y},\mathbf{s})\in(\mathbf{Y},\mathbf{S})} G(\mathbf{x},\mathbf{y},\mathbf{s}), \tag{10}$$

where G is the discriminant function, and the discriminant function G in the model is

$$G(x,y,s) = \langle w,\Phi(x,y,s)\rangle, \tag{11}$$

where $\mathbf{w}$ is the coefficient vector, $\Phi(\mathbf{x},\mathbf{y},\mathbf{s})$ is the structured feature function, and $\langle,\rangle$ is the inner product operation. By combining equation (12), the adaptive scale tracking model is

$$\begin{aligned}\min_{\mathbf{w}} \quad & \left\{\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\varepsilon_i\right\},\\ & \forall_i\colon\ \varepsilon_i>0,\\ \text{s.t.}\quad & \forall_i,\forall_j,\forall_{\mathbf{y}}\neq\mathbf{y}_i,\forall_s\neq\mathbf{s}_j\colon\ \mathbf{w},\langle\boldsymbol{\Phi}\left(\mathbf{x}_i,\mathbf{y}_i,\mathbf{s}_j\right)-\boldsymbol{\Phi}(\mathbf{x}_i,\mathbf{y},\mathbf{s})\rangle\geq\Delta\left(\mathbf{y}_i,\mathbf{s}_j;\mathbf{y},\mathbf{s}\right)-\varepsilon_i,\end{aligned} \tag{12}$$

where is the sampling number, ε_i is the relaxation amount, and $\Delta(\mathbf{y}_i,\mathbf{s}_j;\mathbf{y},\mathbf{s})$ is the loss function. The loss function can be written as

$$\Delta\left(\mathbf{y}_i,\mathbf{s}_j;\mathbf{y},\mathbf{s}\right) = 1 - \frac{T\left(\mathbf{y}_i,\mathbf{s}_j\right)\cap T(\mathbf{y},\mathbf{s})}{T\left(\mathbf{y}_i,\mathbf{s}_j\right)\cup T(\mathbf{y},\mathbf{s})}, \tag{13}$$

where $T(\mathbf{y},\mathbf{s})$ is the sample blocks when the target positions and scales are $(\mathbf{y},\mathbf{s})$ and $T(\mathbf{y}_i,\mathbf{s}_j)$ is the sample blocks when the target positions and scales are $(\mathbf{y}_i,\mathbf{s}_j)$.

We can transform the structured SVM problem of equation (12) into a dual problem

$$\max_{\alpha} \left\{ \sum_{i,\mathbf{y}\neq\mathbf{y}_i;j,\mathbf{s}\neq\mathbf{s}_j} \Delta\left(\mathbf{y}_i,\mathbf{s}_j;\mathbf{y},\mathbf{s}\right)\boldsymbol{\alpha}_{ij}^{\mathbf{y},\mathbf{s}} - \frac{1}{2} \sum_{\substack{i,y\neq\mathbf{y}_i,m,s\neq\mathbf{s}_m \\ j,\overline{\mathbf{y}}\neq\mathbf{y}_j,n,\overline{\mathbf{s}}\neq\mathbf{s}_n}} \boldsymbol{\alpha}_{im}^{\mathbf{y},\mathbf{s}}\boldsymbol{\alpha}_{jn}^{\overline{\mathbf{y}},\overline{\mathbf{s}}} \langle \delta\boldsymbol{\Phi}_{im}(\mathbf{y},\mathbf{s}), \delta_{jn}(\overline{\mathbf{y}},\overline{\mathbf{s}})\rangle \right\},$$
$$\forall_i, \forall_{\mathbf{y}} \neq \mathbf{y}_i, \forall_{\mathbf{s}} \neq \mathbf{y}_j\colon \boldsymbol{\alpha}_{ij}^{\mathbf{y},\mathbf{s}} \geq 0, \quad \text{s.t.} \quad \forall_i\colon \sum_{\mathbf{y}\neq\mathbf{y}_i,m,\mathbf{s}\neq\mathbf{s}_j} \boldsymbol{\alpha}_{ij}^{\mathbf{y},\mathbf{s}} \leq C, \tag{14}$$

where $\delta\Phi_{im}(\mathbf{y},\mathbf{s}) = \Phi(\mathbf{x}_i,\mathbf{y}_i,\mathbf{s}_m) - \Phi(\mathbf{x}_i,\mathbf{y},\mathbf{s})$. The discriminant function is

$$F(\mathbf{x},\mathbf{y},\mathbf{s}) = \sum_{i,\mathbf{y}\neq\mathbf{y}_i;j,\mathbf{s}\neq\mathbf{s}_j} \alpha_{ij}^{\overline{\mathbf{y}},\overline{\mathbf{s}}} \langle \delta\boldsymbol{\Phi}_{ij}(\overline{\mathbf{y}},\overline{\mathbf{s}}), \boldsymbol{\Phi}(\mathbf{x},\mathbf{y},\mathbf{s})\rangle. \tag{15}$$

When $\overline{\mathbf{y}} \neq \mathbf{y}_i, \overline{\mathbf{s}} \neq \mathbf{s}_j$, set $\beta_{ij}^{\mathbf{y},\mathbf{s}} = -\alpha_{ij}^{\mathbf{y},\mathbf{s}}$ or $\beta_{ij}^{\mathbf{y},\mathbf{s}} = \sum_{\overline{\mathbf{y}}\neq\mathbf{y}_i;j,\overline{\mathbf{s}}\neq\mathbf{s}_j} \alpha_{ij}^{\overline{\mathbf{y}},\overline{\mathbf{s}}}$, and equation (14) can be converted as

$$\max_{\beta} \left\{ \sum_{i,\mathbf{y},j,\mathbf{s}} \Delta\left(\mathbf{y},\mathbf{s};\mathbf{y}_i,\mathbf{s}_j\right)\beta_{ij}^{\mathbf{y},\mathbf{s}} - \frac{1}{2} \sum_{i,\mathbf{y},\mathbf{s},j,\overline{\mathbf{y}},\overline{\mathbf{s}}} \beta_{im}^{\mathbf{y},\mathbf{s}}\beta_{jn}^{\overline{\mathbf{y}},\overline{\mathbf{s}}} \langle \boldsymbol{\Phi}(\mathbf{x}_i,\mathbf{y},\mathbf{s}), \boldsymbol{\Phi}\left(\mathbf{x}_j,\overline{\mathbf{y}},\overline{\mathbf{s}}\right)\rangle \right\},$$
$$\forall_i, \forall_{\mathbf{y}}, \forall_j, \forall_{\mathbf{s}}\colon \beta_{ij}^{\mathbf{y},\mathbf{s}} \leq \delta\left(\mathbf{y},\mathbf{s};\mathbf{y}_i,\mathbf{s}_j\right)C, \quad \text{s.t.} \quad \forall_i, \forall_j\colon \sum_{y,s} \beta_{im}^{\mathbf{y},\mathbf{s}} = 0, \tag{16}$$

where $\delta(\mathbf{y},\mathbf{s};\mathbf{y}_i,\mathbf{s}_j) = 1$ when $y = \mathrm{y}_i$ and $s = \mathrm{s}_j$, else $\delta(\mathbf{y},\mathbf{s};\mathbf{y}_i,\mathbf{s}_j) = 0$. We can get the discriminant function as follows:

$$G(\mathbf{x},\mathbf{y},\mathbf{s}) = \sum_{i,j,\overline{\mathbf{y}},\overline{\mathbf{s}}} \beta_{ij}^{\overline{\mathbf{y}},\overline{\mathbf{s}}} \langle \boldsymbol{\Phi}\left(\mathbf{x}_i,\overline{\mathbf{y}},\overline{\mathbf{s}}\right), \boldsymbol{\Phi}(\mathbf{x},\mathbf{y},\mathbf{s})\rangle. \tag{17}$$

By selecting the proper kernel function $\mathbf{K}(\mathbf{x},\mathbf{y},\mathbf{s};\overline{\mathbf{x}},\overline{\mathbf{y}},\overline{\mathbf{s}}) = \langle \Phi(\mathbf{x}_i,\overline{\mathbf{y}},\overline{\mathbf{s}}), \Phi(\mathbf{x},\mathbf{y},\mathbf{s})\rangle$, the discriminant function is

$$G(\mathbf{x},\mathbf{y},\mathbf{s}) = \sum_{i,j,\overline{\mathbf{y}},\overline{\mathbf{s}}} \beta_{ij}^{\overline{\mathbf{y}},\overline{\mathbf{s}}} \mathbf{K}(\mathbf{x},\mathbf{y},\mathbf{s};\overline{\mathbf{x}},\overline{\mathbf{y}},\overline{\mathbf{s}}). \tag{18}$$

Equation (18) is the final form of discriminant function in the adaptive scale tracking model.

The algorithm of tracking process is divided into two parts: the target position prediction and the target position and scale determination.

In the process of target position prediction, the target position is sampled equally at first, and then the previous frame of target position is evenly sampled according to the sparse sampling method:

$$\left(\mathbf{Y}^{\text{rough}},\mathbf{s}\right) = \left\{\left(\mathbf{y}_1^{\text{rough}},\mathbf{s}\right), \left(\mathbf{y}_2^{\text{rough}},\mathbf{s}\right) \ldots \left(\mathbf{y}_j^{\text{rough}},\mathbf{s}\right) \ldots \left(\mathbf{y}_l^{\text{rough}},\mathbf{s}\right)\right\}. \tag{19}$$

The structural feature vector of the corresponding position is obtained by extracting the feature of the samples:

$$\left\{\boldsymbol{\Phi}\left(\mathbf{x},\mathbf{y}_1^{\text{rough}},\mathbf{s}\right), \boldsymbol{\Phi}\left(\mathbf{x},\mathbf{y}_2^{\text{rough}},\mathbf{s}\right) \ldots \boldsymbol{\Phi}\left(\mathbf{x},\mathbf{y}_j^{\text{rough}},\mathbf{s}\right) \ldots \boldsymbol{\Phi}\left(\mathbf{x},\mathbf{y}_l^{\text{rough}},\mathbf{s}\right)\right\}. \tag{20}$$

The feature vector $\Phi(\mathbf{x},\mathbf{y}_j^{\text{rough}},\mathbf{s})$ (the corresponding sample $(\mathbf{y}_j^{\text{rough}},\mathbf{s})$ is the target position prediction result and $\mathbf{y}_j^{\text{rough}}$ is the rough target position) that maximizes the decision function is selected according to the discriminant

function equation (18) and the decision function equation (10).

The process of determining the target position and scale includes four steps:

(1) We set up a set of scales $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots \mathbf{s}_j \ldots \mathbf{s}_k\}$ ($k \geq 1$), and the collected elements are satisfied with $\mathbf{s}_j - \mathbf{s}_{j-1} = \text{gap}$ (where gap is the step size).

(2) By taking the rough estimated target location $\mathbf{y}_j^{\text{rough}}$ as center, the search area is further reduced and we can sample the scale set $\mathbf{S}$ in the reduced search area and get k sampling results:

$$\left(\mathbf{Y}^{\text{accurate}}, \mathbf{S}\right) = \left\{ \begin{array}{l} \left(\mathbf{y}_{11}^{\text{accurate}}, \mathbf{s}_1\right), \left(\mathbf{y}_{12}^{\text{accurate}}, \mathbf{s}_1\right) \ldots \left(\mathbf{y}_{1q}^{\text{accurate}}, \mathbf{s}_1\right) \ldots \left(\mathbf{y}_{1m}^{\text{accurate}}, \mathbf{s}_1\right); \\ \left(\mathbf{y}_{21}^{\text{accurate}}, \mathbf{s}_2\right), \left(\mathbf{y}_{22}^{\text{accurate}}, \mathbf{s}_2\right) \cdots \left(\mathbf{y}_{2q}^{\text{accurate}}, \mathbf{s}_2\right) \ldots \left(\mathbf{y}_{2m}^{\text{accurate}}, \mathbf{s}_2\right); \\ \vdots \\ \left(\mathbf{y}_{p1}^{\text{accurate}}, \mathbf{s}_p\right), \left(\mathbf{y}_{p2}^{\text{accurate}}, \mathbf{s}_p\right) \ldots \left(\mathbf{y}_{pq}^{\text{accurate}}, \mathbf{s}_p\right) \ldots \left(\mathbf{y}_{pm}^{\text{accurate}}, \mathbf{s}_p\right); \\ \vdots \\ \left(\mathbf{y}_{k1}^{\text{accurate}}, \mathbf{s}_k\right), \left(\mathbf{y}_{k2}^{\text{accurate}}, \mathbf{s}_k\right) \ldots \left(\mathbf{y}_{kq}^{\text{accurate}}, \mathbf{s}_k\right) \ldots \left(\mathbf{y}_{km}^{\text{accurate}}, \mathbf{s}_k\right) \end{array} \right\}. \tag{21}$$

(3) Perform feature extraction for each sample and obtain k structured feature vector

$$\left\{ \begin{array}{l} \boldsymbol{\Phi}\left(\mathbf{x}, \mathbf{y}_{11}^{\text{accurate}}, \mathbf{s}_1\right), \boldsymbol{\Phi}\left(\mathbf{x}, \mathbf{y}_{12}^{\text{accurate}}, \mathbf{s}_1\right) \cdots \boldsymbol{\Phi}\left(\mathbf{x}, \mathbf{y}_{1q}^{\text{accurate}}, \mathbf{s}_1\right) \cdots \boldsymbol{\Phi}\left(\mathbf{x}, \mathbf{y}_{1m}^{\text{accurate}}, \mathbf{s}_1\right); \\ \boldsymbol{\Phi}\left(\mathbf{x}, \mathbf{y}_{21}^{\text{accurate}}, \mathbf{s}_2\right), \boldsymbol{\Phi}\left(\mathbf{x}, \mathbf{y}_{22}^{\text{accurate}}, \mathbf{s}_2\right) \cdots \boldsymbol{\Phi}\left(\mathbf{x}, \mathbf{y}_{2q}^{\text{accurate}}, \mathbf{s}_2\right) \cdots \boldsymbol{\Phi}\left(\mathbf{x}, \mathbf{y}_{2m}^{\text{accurate}}, \mathbf{s}_2\right); \\ \vdots \\ \boldsymbol{\Phi}\left(\mathbf{x}, \mathbf{y}_{p1}^{\text{accurate}}, \mathbf{s}_p\right), \boldsymbol{\Phi}\left(\mathbf{x}, \mathbf{y}_{p2}^{\text{accurate}}, \mathbf{s}_p\right) \cdots \boldsymbol{\Phi}\left(\mathbf{x}, \mathbf{y}_{pq}^{\text{accurate}}, \mathbf{s}_p\right) \cdots \boldsymbol{\Phi}\left(\mathbf{x}, \mathbf{y}_{pm}^{\text{accurate}}, \mathbf{s}_p\right); \\ \vdots \\ \boldsymbol{\Phi}\left(\mathbf{x}, \mathbf{y}_{k1}^{\text{accurate}}, \mathbf{s}_k\right), \boldsymbol{\Phi}\left(\mathbf{x}, \mathbf{y}_{k2}^{\text{accurate}}, \mathbf{s}_k\right) \cdots \boldsymbol{\Phi}\left(\mathbf{x}, \mathbf{y}_{kq}^{\text{accurate}}, \mathbf{s}_k\right) \cdots \boldsymbol{\Phi}\left(\mathbf{x}, \mathbf{y}_{km}^{\text{accurate}}, \mathbf{s}_k\right); \end{array} \right\}. \tag{22}$$

(4) Select the feature vector $\Phi(\mathbf{x}_{pq}, \mathbf{y}_{pq}^{\text{accurate}}, \mathbf{s}_p)$ (the corresponding sample $(\mathbf{y}_{pq}^{\text{accurate}}, \mathbf{s}_p)$, in which $\mathbf{y}_{pq}^{\text{accurate}}$ is the exact position and $\mathbf{s}_p$ is the target scale, is the tracking result) that maximizes the decision function according to equations (18) and (22).

The performance of the Struck tracker is decreased significantly when the target is partially occluded or completely occluded. We perform an active occlusion detection during the tracking process of each frame.

Let $I_t^{\mathbf{P}_{t-1}O\mathbf{y}}$ be the value of classifier discriminant function at the search position $\mathbf{P} = \mathbf{P}_{t-1}O\mathbf{y}$ in frame I_t (where t is the frame number of tracking target):

$$I_t^{\mathbf{P}_{t-1}O\mathbf{y}} = I\left(\mathbf{x}_t^{\mathbf{P}_{t-1}}, \mathbf{y}\right). \tag{23}$$

For the target position of the current frame, the classifier discriminant function value at the satisfying target position $\mathbf{P} = \mathbf{P}_{t-1}O\mathbf{y}_t$ is $I_t^{\mathbf{P}_{t-1}O\mathbf{y}_t}$, then

$$I_t = I\left(\mathbf{x}_t^{\mathbf{P}_{t-1}}, \mathbf{y}_t\right). \tag{24}$$

The change rate of I_t can be defined as

$$V_{I_t} = \frac{I_t - \overline{I}}{\overline{I}}. \tag{25}$$

The queue $\mathbf{Q} = \{I_1, \ldots, I_v\}$ ($v \in t$) is constructed, and it stores the value of v frame history I_t, where $\overline{I} = (1/v)\sum_{i=1}^{i=v}\mathbf{Q_i}$ is the average value of elements in queue $\mathbf{Q}$. After each frame target tracking, the current frame I_t value is added to the queue $\mathbf{Q}$ to update the occlusion detection.

The threshold of occlusion detection γ is introduced to distinguish whether the target is occluded or not. When $V_{I_t} > \gamma$, the algorithm continues to track the target. When $V_{I_t} \leq \gamma$, the algorithm stops updating the element of queue $\mathbf{Q}$, and the classifier to ensure that the classifier does not introduce error information.

3.3. Steps of the Suggested Method. The proposed algorithm steps are shown in Algorithm 1. The steps of the developed method are as follows.

4. Experiments

The tacking performance of six trackers, which include the structured output tracking with kernels (Struck), the kernel correlation filter (KCF), the background-aware correlation

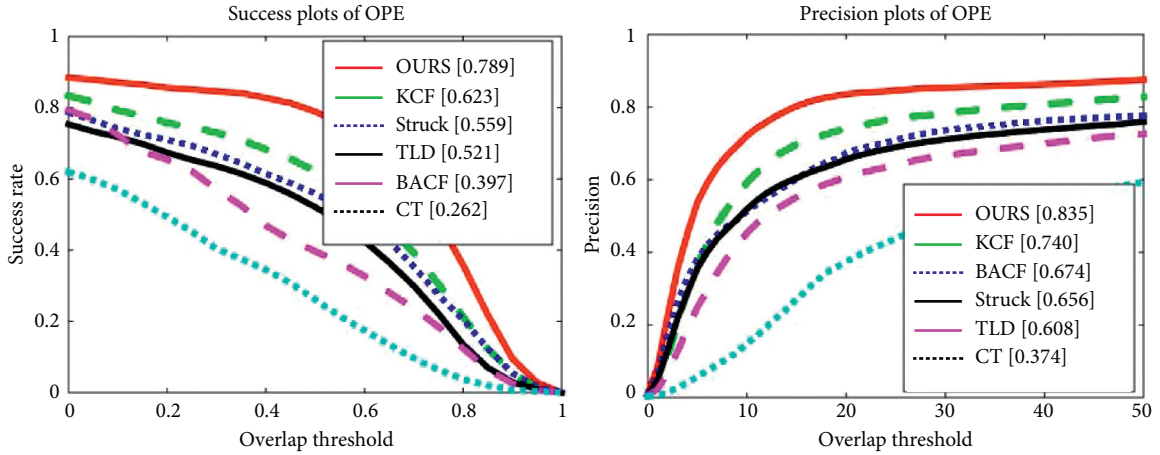

Figure 1: The tracking results under the OTB-2015 dataset.

Input:

I_t: the t^{th} frame video $(t \geq 1)$
$\mathbf{Y}$: target position set of the previous frame
$\mathbf{S}$: target scale set of the previous frame
γ: occlusion threshold

Output:

$\mathbf{Y}^{\text{rough}}$: target prediction position set of the current frame
$\Phi(\mathbf{x}, \mathbf{y}_l^{\text{rough}}, \mathbf{s})$: the structured feature vector of rough position
$\mathbf{y}_j^{\text{rough}}$: the target rough position
$\Phi(\mathbf{x}_{pq}, \mathbf{y}_{pq}^{\text{accurate}}, \mathbf{s}_p)$: the structured feature vector of accurate position
$\mathbf{s}_p$: target scale of the current frame
$\mathbf{y}_{pq}^{\text{accurate}}$: the accurate position of target

End
Target rough position prediction process

(1) Obtain the sparse sampling result $\mathbf{Y}^{\text{rough}}$ of target prediction position from equation (19)
(2) Get the structured feature vectors $\Phi(\mathbf{x}, \mathbf{y}_l^{\text{accurate}}, \mathbf{s})$ according to equation (20) by performing feature extraction on the samples
(3) Take the structured feature vector with maximal decision function as the target rough position $\mathbf{y}_j^{\text{rough}}$ according to equations (10) and (18)

Determination of target accurate position and scale

(4) Set a set of scales $\mathbf{S}$ and obtain k scale samples
(5) Densely sample each scale of the set $\mathbf{S}$ and get k sets sampling results from equation (21) taking estimated position $\mathbf{y}_j^{\text{rough}}$ as the center
(6) Perform feature extraction on the samples to obtain k groups of structured feature vectors according to equation (22)
(7) The target scale $\mathbf{s}_p$ and the accurate position $\mathbf{y}_{pq}^{\text{accurate}}$ can be obtained by calculating the feature vectors $\Phi(\mathbf{x}_{pq}, \mathbf{y}_{pq}^{\text{accurate}}, \mathbf{s}_p)$ which maximize the decision function

The target occlusion judgment process

(8) Set the size of occlusion threshold γ
(9) Calculate the change rate value V_{I_t} of the discriminant function according to equations (24) and (25)

If $V_{I_t} \leq \gamma$

(10) The tracking target is occluded and the classifier and the queue $\mathbf{Q}$ element stop updating

else

(11) The tracking target is not occluded and the tracker continues to track the target

Until the end of all frames
End

Algorithm 1: A real-time structured output tracker with scale adaption for visual target tracking.

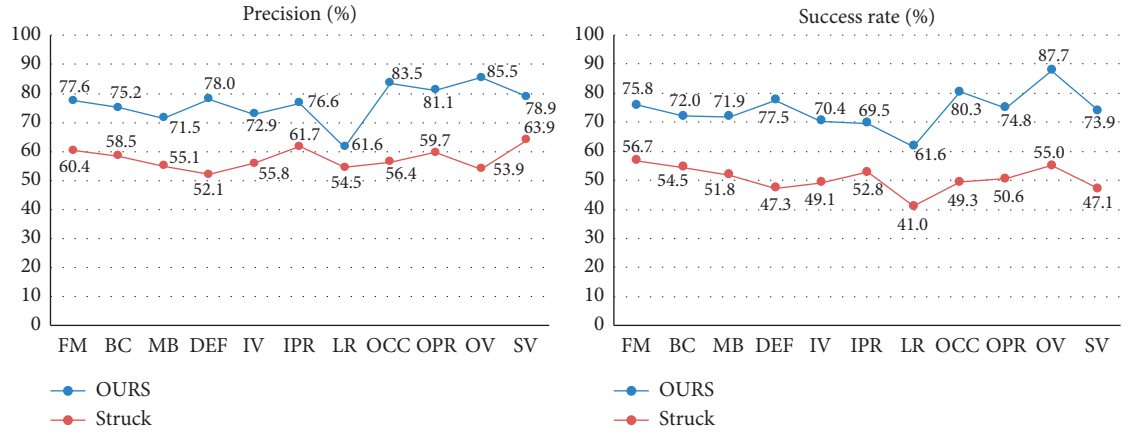

Figure 2: Average success rate plot and average precision in various complex background.

Table 1: The selected sequences information.

Sequence	FPS	Backgrounds
Faceocc1	659	OCC
Coke	569	OCC, BC, FM
Faceocc2	307	IV, OCC
Deer	71	FM, MB
Man	1741	IV
Dog1	500	OPR, SV, IRP

Table 2: The success rate of various trackers in selected sequences.

Method	Faceocc1	Coke	Faceocc2	Deer	Dog1	Man
OURS	99.1	100	99.8	84.5	99.3	99.8
Struck	80.5	71.5	82.5	58.4	86.6	78.0
KCF	87.1	70.3	79.9	80.1	88.1	81.3
BACF	84.3	63.4	75.6	60.1	82.0	72.1
TLD	79.7	54.8	74.3	43.3	70.1	67.7
CT	85.9	32.0	60.5	47.2	53.3	55.3

Table 3: The frames per second of various trackers in selected sequences.

Method	Faceocc1	Coke	Faceocc2	Deer	Dog1	Man
OURS	97	108	119	58	112	109
Struck	34	53	76	28	62	75
KCF	87	81	105	47	99	107
BACF	69	55	61	32	71	77
TLD	12	14	23	11	35	45
CT	37	39	46	17	60	66

filters (BACF), the compressive tracking (CT), the tracking-learning-detection (TLD), and the OURS tracker, is tested on the OTB-2015 benchmark dataset. The OTB-2015 dataset comprises 11 different complex backgrounds, namely, scale variation (SV), deformation (DEF), illumination variation (IV), out of view (OV), background clutter (BC), occlusion (OCC), motion blur (MB), fast motion (FM), out of plane rotation (OPR), in-plane rotation (IPR), and low resolution (LR). The overlap success rate and the center position error are used as evaluation criteria for trackers [31, 32]. The overlap success rate reflects the degree of overlap between the tracked target frame and the actual target frame [33, 34]. The center position error indicates the offset between the output target center position and the labeled position [35, 36].

4.1. Parameter Setup. Experiments are implemented in Matlab on an Intel I7–8565U 1.8 GHz CPU with 8 GB RAM. The regularization parameter C is set to 100, the size of support vector threshold is set to 100, the original search range is three times the target size of the previous frame, the number of scales is 31, the historical frame F_t is set as $\mathbf{G} = \{I_1, \ldots, I_k\}$, the size of k is 10, the threshold

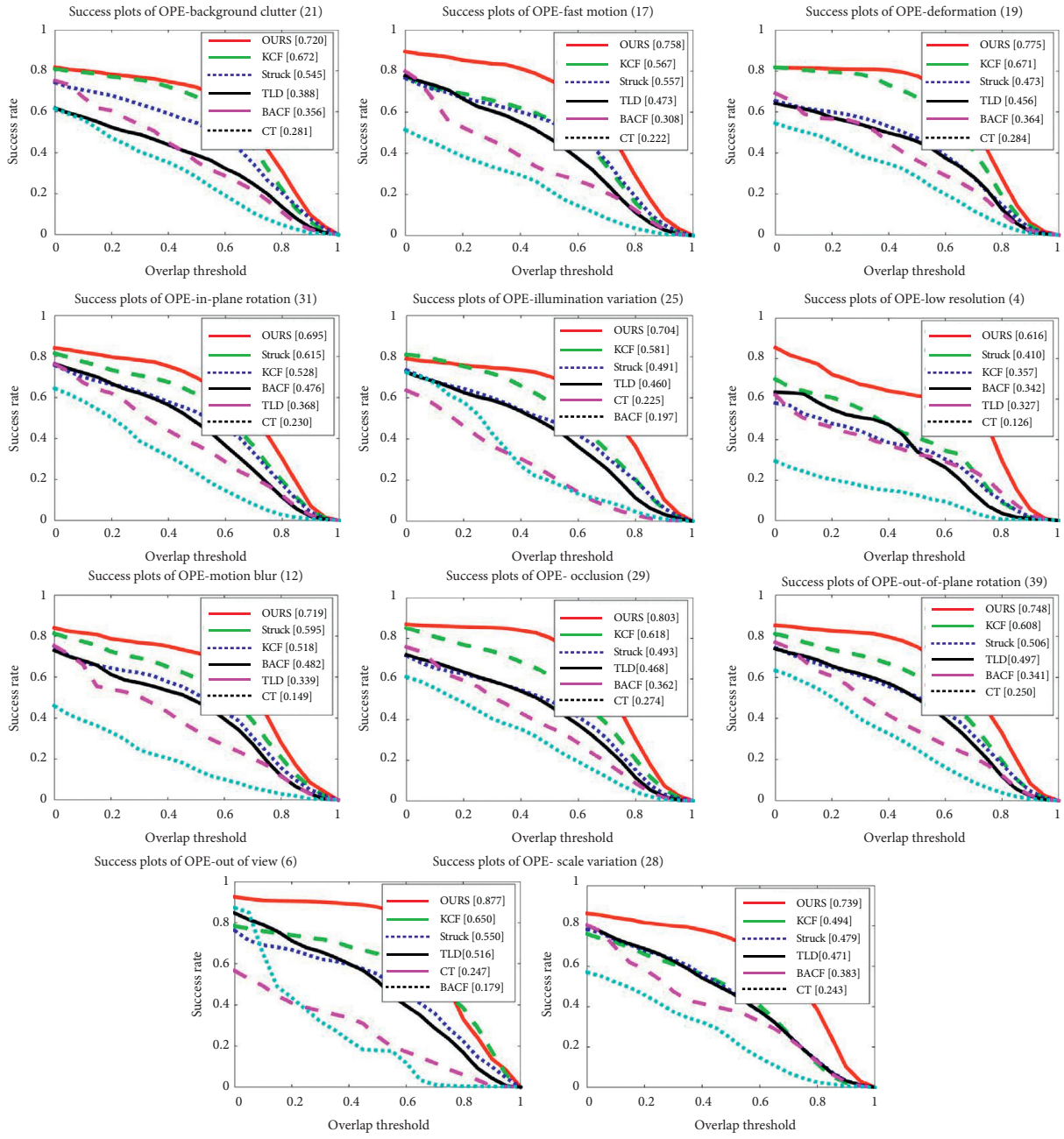

Figure 3: Tracking success rate of trackers in 11 challenging video sequences.

value of occlusion detection γ is −0.3, the target boundary box size change factor δ is 0.1, and the search radius r is 30.

4.2. Experiments on OTB-2015 Benchmark Dataset. The tracking performance of different trackers is tested on the OTB-2015 benchmark dataset. The tracking results under the OTB-2015 dataset are shown in Figure 1, and we can get that the tracking results of OURS tracker are better than comparsion trackers. The OURS tracker and the Struck tracker are also tested under various complex background, and the selected sequence information is shown in Table 1, and their average success rate plot and average precision in various complex background are shown in Figure 2. Figure 2 shows that the performance of the OURS tracker is better than that of the Struck tracker on the OTB-2015 benchmark dataset.

The success rate of various trackers in selected sequences is demonstrated in Table 2. From Table 2, the tracking performance of OURS tracker is better than those of other comparison algorithms.

The frames per second of various trackers in selected sequences are shown in Table 3. From Table 3, the running speeds of OURS tracker is higher than those of other comparison algorithms.

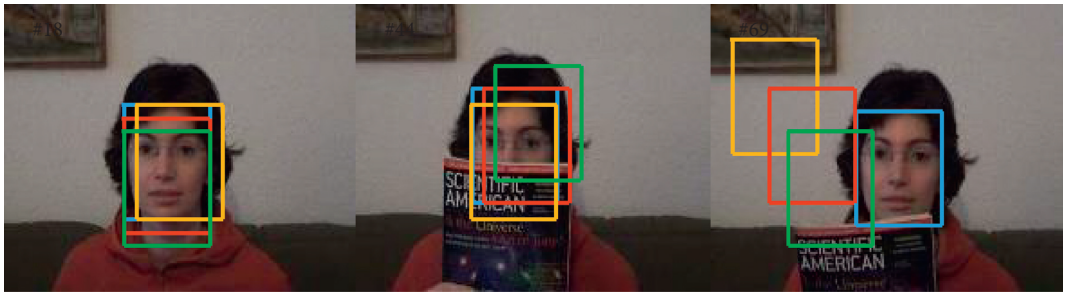

Figure 4: Faceocc1 (frames: 18, 44, 69).

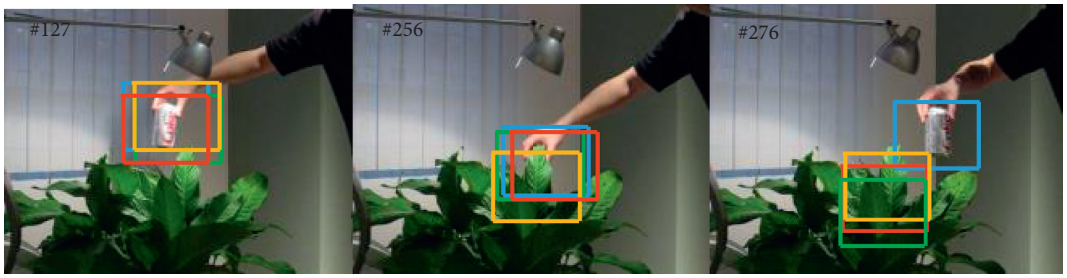

Figure 5: Coke (frames: 127, 256, 276).

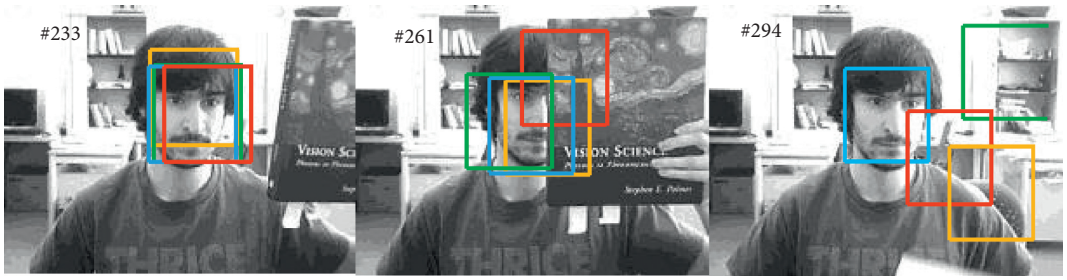

Figure 6: Faceocc2 (frames: 233, 261, 294).

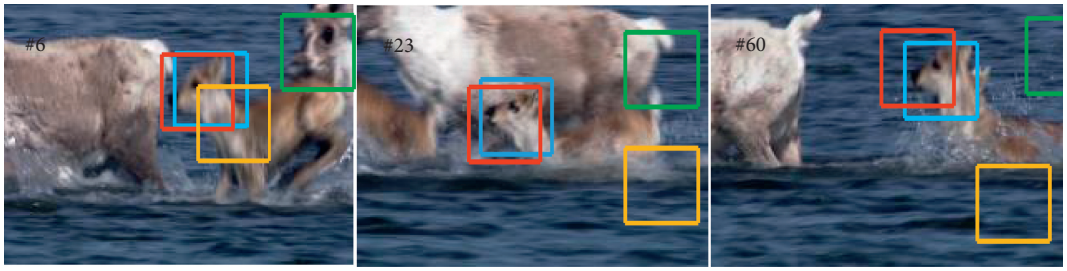

Figure 7: Deer (frames: 6, 23, 60).

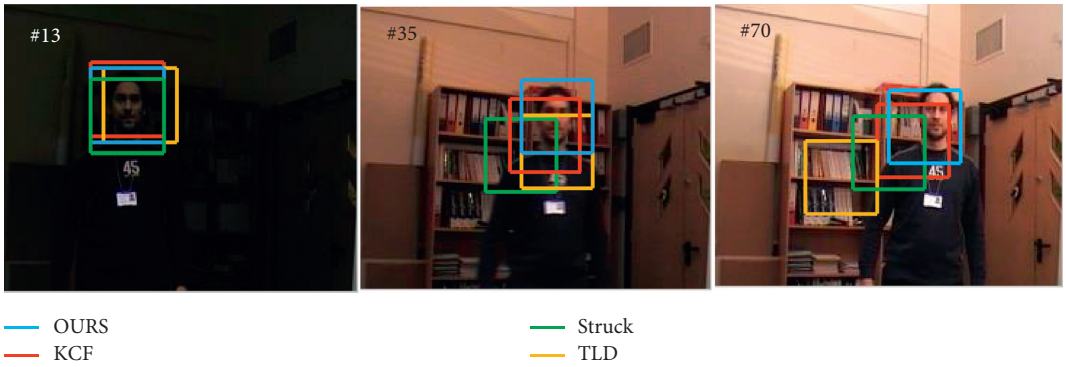

Figure 8: Man (frames: 13, 35, 70).

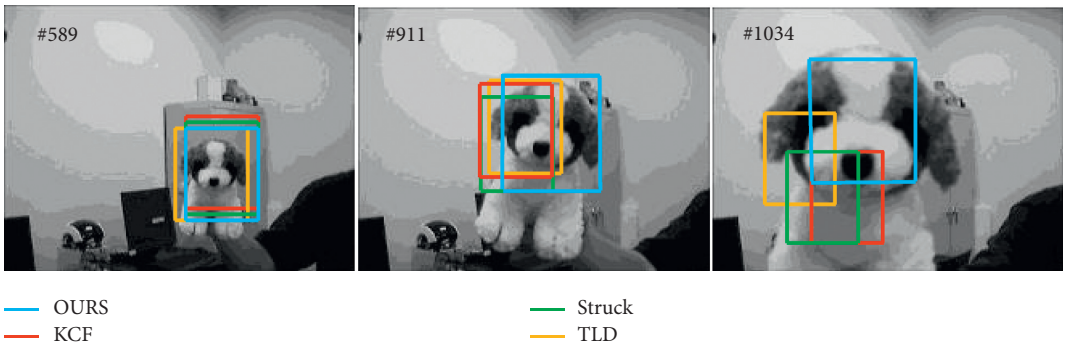

Figure 9: Dog (frames: 589, 911, 1034).

4.3. The Success Rate on More Challenging Video Sequences. In addition, the tracking success rate plots under 11 challenging video sequences are shown in Figure 3. We can see from Figure 3 that the OURS tracker shows excellent tracking performance under 11 challenging video sequences than other trackers. Therefore, OURS tracker processes challenging video sequence effects better than other comparison trackers.

4.4. Experiments on Various Complex Video Sequences. In the experiment on various complex video sequences, the comparison algorithms including OURS, Struck, KCF, and TLD are selected to test various video sequences, and their corresponding performance is discussed. The video sequences are Faceocc1, Coke, Faceocc2, Deer, Man, and Dog. The sequences information is shown in Table 1, and the test results of sequences are shown in Figures 4–9 (the original images of Figures 4–9 are from https://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html).

In the video sequence Faceocc1, the challenge during tracking is partial occlusion. The TLD tracker and the Struck tracker occur to drift at the 44th frame. Due to the removal of the target from partial occlusion, the comparsion trackers lose the target at the 69th frame, but the OURS tracker can deal with the problem of partial occlusion.

In the video sequence Coke, the target in video sequence is affected by complete occlusion. The target is completely occluded at the 256th frame. The Struck tracker, the KCF tracker, and TLD tracker tracking fails at the 276th frame, but the OURS tracker can deal with full occlusion problem.

In the video sequence Faceocc2, the partial occlusion affects the tracking target during the tracking process. The target is partial occluded at the 261th frame. The comparsion trackers lose the target when target removes from partial occlusion at the 294th frame, but the OURS tracker can solve the partial occlusion problem.

In the video sequence Deer, the tracking target appears fast motion. The Struck tracker and the TLD tracker lose the target at the 19th frame. The KCF tracker loses the target at the 60th frame, but OURS tracker can solve the problem of fast motion.

In the video sequence Man, the illumination variation affects the tracking target during the tracking process. The TLD tracker and the Struck tracker occur to drift in video sequence at the 35th frame. All comparison trackers lose the target when the background illumination is changed gradually, but the OURS tracker can solve the illumination variation problem.

In the video sequence Dog, the tracking target scale changes. The comparsion trackers occur to drift at the 911th

frame. The comparsion trackers lose the target at the 1034th frame, but the OURS tracker can adapt to changes in target scale and track the target correctly.

5. Conclusion

In this work, a real-time structured output tracker with scale adaption is proposed: (1) the process of position target prediction which can improve the tracking real-time performance is added during the tracking process; (2) multiscale sampling is used to obtain samples of different scales, and the best scale is obtained by using a discriminator to improve the accuracy of tracking; and (3) the occlusion judgment mechanism is suggested to determine whether to update the classifier or not, and the Kalman filtering is applied to solve the problem of continuous tracking with occlusion.

The tracking performance of OURS tracker is better than those of other trackers in different research cases due to the following advantages. The OURS tracker uses a multiscale sampling strategy to estimate the scale of target during tracking. The OURS tracker uses Kalman filter to solve tracking problem with target occlusion. From the experimental results, the tracker proposed in this paper shows excellent performance when processing various complex backgrounds under the OTB-2015 dataset, and it also achieves excellent success rate and tracking accuracy in different challenging complex backgrounds.

In the future, our research work will focus on applying the proposed algorithm to multitarget tracking due to the successful application of the proposed algorithm on the single target tracking.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61671222) and the Postgraduate Research and Practice Innovation Program of Jiangsu Province (no. KYCX19_1693).

References

[1] H. Hu, B. Ma, J. Shen, and L. Shao, "Manifold regularized correlation object tracking," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1786–1795, 2018.

[2] D. Yuan, X. Lu, D. Li, Y. Liang, and X. Zhang, "Particle filter re-detection for visual tracking via correlation filters," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 14277–14301, 2019.

[3] N. Wang, W. Zhou, Q. Tian, R. Hong, M. Wang, and H. Li, "Multi-cue correlation filters for robust visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4844–4853, Salt Lake City, UT, USA, June 2018.

[4] D. Yuan, W. Kang, and Z. yu, "Robust visual tracking with correlation filters and metric learning," *Knowledge-Based Systems*, vol. 195, Article ID 105697, 2020.

[5] X. Zhang, G.-S. Xia, Q. Lu, W. Shen, and L. Zhang, "Visual object tracking by correlation filters and online learning," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 140, no. 1, pp. 77–89, 2018.

[6] C. Sun, D. Wang, C. L. Hu, and M. H. Yang, "Learning spatial aware regressions for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern*, pp. 8962–8970, Salt Lake City, UT, USA, June 2018.

[7] H. J. Wang and H. J. Ge, "Visual tracking using discriminative representation with l 2 regularization," *Frontiers of Computer Science*, vol. 12, no. 1, pp. 1–13, 2018.

[8] D. Yuan, X. Zhang, J. Liu, and D. Li, "A multiple feature fused model for visual object tracking via correlation filters," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 27271–27290, 2019.

[9] M. Kristan, A. Leonardis, and J. Matas, "The visual object tracking VOT2017 challenge results," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1949–1972, Honolulu, HI, USA, July 2017.

[10] M. Dai, S. Cheng, and X. He, "A structural correlation filter combined with a multi-task gaussian particle filter for visual tracking," 2018, https://arxiv.org/abs/1803.05845.

[11] B. Zhang, Z. Li, X. Cao et al., "Output constraint transfer for kernelized correlation filter in tracking," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 4, pp. 693–703, 2017.

[12] K. H. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1135–1143, Venice, Italy, October 2017.

[13] Z. Zhao, T. Wang, and F. Liu, "Remarkable local resampling based on particle filter for visual tracking," *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 1–26, 2017.

[14] Y. Li, Y. Zhang, Y. Xu, J. Wang, and Z. Miao, "Robust scale adaptive kernel correlation filter tracker with hierarchical convolutional features," *IEEE Signal Processing Letters*, vol. 23, no. 8, pp. 1136–1140, 2016.

[15] J. Valmadre, L. Bertinetto, J. F. Henriques, and A. Vedaldi, "End-to-end representation learning for Correlation Filter based tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5000–5008, Venice, Italy, October 2017.

[16] D. Caulfield and K. Dawson-Howe, "Evaluation of multi-part models for mean-shift tracking," in *Proceedings of the Machine Vision And Image Processing Conference*, pp. 77–82, Mumbai, India, February 2008.

[17] C. Wang and Z. Li, "Mean shift based orientation and location tracking of targets," in *Proceedings of the International Conference on Natural Computation*, pp. 3593–3596, Yantai, China, August 2010.

[18] P. Wu, L. Kong, and F. Zhao, "Particle filter tracking based on color and sift features," in *Proceedings of the Language And Image Processing*, pp. 932–937, Shanghai, China, July 2008.

[19] B. Babenko, M. H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1003–1009, IEEE, Miami, FL, USA, June 2009.

[20] J. Chunxiao, W. Zhongli, W. Xian, and C. Baigen, "A tracking-learning-detection method with local binary pattern improved," in *Proceedings of the International Conference on*

Computer Vision, pp. 123–133, Araucano Park, Las Condes, Chile, December 2015.

[21] S. Hare, A. Saffari, and P. H. Torr, "Struck: structured output tracking with kernels," in *Proceedings of the International Conference on Computer Vision*, pp. 263–270, Venice, Italy, October 2017.

[22] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual target tracking using adaptive correlation filters," *IEEE Transactions on Computer Vision and Pattern Recognition*, vol. 7, no. 3, pp. 2544–2550, 2010.

[23] J. F. Henriques, R. Caseiro, and P. Martins, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proceedings of the European Conference on Computer Vision*, pp. 702–715, Florence, Italy, October 2012.

[24] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[25] H. Y. Zhang and X. Zheng, "Spatio-temporal context tracking algorithm based on dual-object model," *Optics and Precision Engineering*, vol. 24, no. 1, pp. 1215–1223, 2016.

[26] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proceedings of the European Conference on Computer Vision*, pp. 254–265, Zurich, Switzerland, September2014.

[27] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *Proceedings of the European Conference on Computer Vision*, pp. 749–765, Amsterdam, The Netherlands, October 2016.

[28] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the European Conference on Computer Vision*, pp. 4293–4302, Amsterdam, The Netherlands, October 2016.

[29] M. Danelljan, A. Robinson, and F. S. Khan, "Beyond correlation filters: learning continuous convolution operators for visual tracking," in *Proceedings of the European Conference on Computer Vision*, pp. 472–488, Amsterdam, The Netherlands, October 2016.

[30] M. Danelljan, G. Bhat, and F. S. Khan, "Efficient convolution operators for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 21–26, Honolulu, HI, USA, July 2017.

[31] J. Zhang, X. Jin, J. Sun, J. Wang, and SA. Kumar, "Spatial and semantic convolutional features for robust visual object tracking," *Multimedia Tools and Applications*, vol. 79, pp. 15095–15115, 2020.

[32] S. Moradi, P. Moallem, and M. F. Sabahi, "A false-alarm aware methodology to develop robust and efficient multi-scale infrared small target detection algorithm," *Infrared Physics & Technology*, vol. 89, no. 3, pp. 387–397, 2018.

[33] J. Zhang, Y. Wu, W. Feng, and J. Wang, "Spatially attentive visual tracking using multi-model adaptive response fusion," *IEEE Access*, vol. 7, no. 1, pp. 83873–83887, 2019.

[34] A. He, C. Luo, and X. Tian, "A twofold siamese network for real–time object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4834–4843, Long Beach, CA, USA, June 2018.

[35] L. Yong-Hwan, A. Hyochang, A. Hyo-Beom, and L. Sun-Yuong, "Visual object detection and tracking using analytical learning approach of validity level," *Intelligent Automation and Soft Computing*, vol. 25, no. 1, pp. 205–215, 2019.

[36] Z. Yin, C. Min, and L. Xiaofei, "Big data service architecture: a survey," *Journal of Internet Technology*, vol. 21, no. 2, pp. 393–405, 2020.