Robust long-term object tracking with adaptive scale and rotation estimation

Huimin Lu¹, Dan Xiong², Junhao Xiao¹ and Zhiqiang Zheng¹

Abstract

In this article, a robust long-term object tracking algorithm is proposed. It can tackle the challenges of scale and rotation changes during the long-term object tracking for security robots. Firstly, a robust scale and rotation estimation method is proposed to deal with scale changes and rotation motion of the object. It is based on the Fourier–Mellin transform and the kernelized correlation filter. The object’s scale and rotation can be estimated in the continuous space, and the kernelized correlation filter is used to improve the estimation accuracy and robustness. Then a weighted object searching method based on the histogram and the variance is introduced to handle the problem that trackers may fail in the long-term object tracking (due to semi-occlusion or full occlusion). When the tracked object is lost, the object can be relocated in the whole image using the searching method, so the tracker can be recovered from failures. Moreover, two other kernelized correlation filters are learned to estimate the object’s translation and the confidence of tracking results, respectively. The estimated confidence is more accurate and robust using the dedicatedly designed kernelized correlation filter, which is employed to activate the weighted object searching module, and helps to determine whether the searching windows contain objects. We compare the proposed algorithm with state-of-the-art tracking algorithms on the online object tracking benchmark. The experimental results validate the effectiveness and superiority of our tracking algorithm.

Keywords

Scale and rotation, Fourier–Mellin transform, kernelized correlation filter, histogram and variance

Date received: 21 October 2019; accepted: 18 December 2019

Introduction

As a major and challenging research topic in the computer vision community, object tracking has been researched for several decades. A large number of papers and algorithms have been published and presented, and readers can refer to surveys¹–³ or visual object tracking competitions on various benchmarking data sets.⁴–⁶ According to statistical models of visual appearance, trackers can be mainly divided into two categories: generative model-based object tracking and discriminative model-based object tracking.

Generative model-based trackers firstly construct an object appearance model and then find out the most similar region in the image by fitting the model. Wang et al.⁷ presented the spatial color mixture of Gaussians appearance model by simultaneously encoding both color and its spatial layout. Black and Jepson⁸ put forward a vector-based subspace method to model rigid and articulated objects. Chin and Suter⁹ proposed an incremental kernel principal

¹ Robotics Research Center, College of Intelligence Science and Technology, National University of Defense Technology, Changsha, China
² Unmanned Systems Research Center, National Innovation Institute of Defense Technology, Beijing, China

Corresponding author:
Dan Xiong, Unmanned Systems Research Center, National Innovation Institute of Defense Technology, Beijing 100071, China.
Email: nudtxiong@163.com
component analysis method to construct a nonlinear subspace model for tracked objects with constant updating speed and memory usage. Generative models only use the information of object appearance without taking into account the background. As a result, good performance cannot be obtained in cluttered environments. Discriminative model-based trackers take visual object tracking as a binary classification problem, which models both the object and its surroundings. Grabner et al.\textsuperscript{10} employed an online boosting algorithm to train ensemble classifiers to distinguish the tracked object and the background. Zhang et al.\textsuperscript{11} learned Naive Bayes classifiers from compressive features for object tracking. Kalal et al.\textsuperscript{12} proposed a tracking-learning-detection framework for long-term single object tracking. An independent detector can recover or correct a tracker from tracking failures, or with a large drift. The combination of the tracking trajectory generated by the tracker and the object’s location obtained by the detector provided unlabeled training data with structure constraints, which can improve the discriminative performance of binary classifiers.

In recent years, correlation filters have been widely applied in object tracking and object recognition, and good performance has been achieved. Correlation filter-based object tracking methods employ discriminative models to represent the object’s visual appearance, which is very efficient and simple because of the fact that the convolution of two image patches in the time domain equals an element-wise product in the Fourier domain. Bolme et al.\textsuperscript{13} proposed an object tracking algorithm based on minimum output sum of squared error (MOSSE) filters. The tracker is robust to variations in the lighting, pose, and even locally nonrigid deformations and has shown remarkable performance despite their simplicity and high frame rates. However, the MOSSE tracker does not take into consideration variations in the scale and rotation. By employing the kernel trick on the correlation filter, the kernelized correlation filter (KCF) tracker achieved impressive performance in object tracking.\textsuperscript{14} The KCF tracker is also restrained to only estimating the object translation. To sort out this problem, the SAMF tracker was proposed to estimate the object’s scale factor by detecting the tracked object in several scaled images by bilinear interpolation. At the same time, histogram of oriented gradient (HOG) and color naming were fused together to improve the performance.\textsuperscript{15} Danelljan et al.\textsuperscript{16} proposed a discriminative scale space tracker to estimate the object scale by training correlation filters in a scale pyramid using HOG features. However, correlation filter-based object tracking methods usually utilize a periodic assumption to generate the training samples in the target neighborhood, thus boundary effects are inevitable.

To address the problem, a spatial regularization component was introduced to penalize correlation filter coefficients.\textsuperscript{17} Ma et al.\textsuperscript{18} proposed a robust scale adaptive tracking algorithm which used a sequential Monte Carlo method to estimate the object scale and determined the object location by a KCF. Hu et al.\textsuperscript{19} introduced an independent filter to estimate the object’s scale and figured out a local strategy to expand the searching area of the tracker to deal with the object’s fast motion and occlusion. Zhou et al.\textsuperscript{20} proposed a spatiotemporal context learning method with multichannel features using an improved scale adaptive scheme for object tracking, and good performance was achieved. Correlation filter-based trackers are robust to motion blur and illumination changes, but they are notoriously sensitive to the deformation since the learned models depend strongly on spatial layouts of the tracked object. Bertinetto et al.\textsuperscript{21} proposed a simple combination of correlation filters and histogram scores to deal with motion blur, illumination changes, and deformation, and it hit the target that coming up with a good performance. Zuo et al.\textsuperscript{22} proposed an effective and efficient approach to learn robust and discriminative support correlation filters for real-time object tracking. Ma et al.\textsuperscript{23} proposed a long-term correlation tracking algorithm. Three correlation filters were trained to estimate the translation, the scale variations, and the confidence of tracking results. In addition, a random fern classifier was trained to redetect the tracked object in case of tracking failures.

During security robots performing the surveillance and reconnaissance tasks, it is highly impossible to avoid scale changes and rotation motion of the object in tracking. In the past, researchers focused much more on the estimation of the object’s scale while less on the object’s rotation motion else. As a result, robust scale and rotation estimation is still a critical and challenging problem. Furthermore, in object tracking, especially long-term object tracking, trackers sometimes lose the object or suffer from a large drift due to some unavoidable difficulties such as the occlusion, changes of the scene illumination, and the object’s appearance, low-quality or compressed images, fast and complex object maneuvering. It is critical for long-term trackers to be self-recovered or reinitialized from failures. The detection modules are dedicatedly designed to recover or correct the tracker in the literature.\textsuperscript{12,23} However, it is difficult to detect rotated objects robustly and accurately. Moreover, these detectors need to scan each sliding window to determine whether it contains the object, which usually requires a lot of computing time and resources so as to reduce the real-time performance of tracking algorithms. In this article, we proposed a robust long-term object tracking (RLOT) approach with adaptive scale and rotation estimation. The Fourier–Mellin transform takes advantage of the fact that scale changes and rotation motion in the original image are manifested as pure translations in the log-polar space of the magnitude spectrum, which has been widely employed in image registration.\textsuperscript{24–26} Inspired by the Fourier–Mellin transform, we employed the KCF on the log-polar space of the magnitude spectrum and hit the target of robust scale and rotation estimation in object tracking. What’s more, we proposed a weighted object searching method based on the histogram and the variance to redetect
the lost object for trackers. Some high weight candidate sliding windows can be sampled for further processing by Monte Carlo sampling techniques, thus lots of sliding windows are discarded and the real-time performance of the detection module is improved. Furthermore, two other correlation filters are trained to estimate the object translation and the confidence of tracking results. Accurate estimation of the confidence can be used to activate the weighted object searching module for relocating the tracked target when the tracker fails. The proposed algorithm will greatly improve the long-term object tracking ability of security robots in complex environments.

The rest of this article is organized as follows. The proposed scale and rotation estimation based on the Fourier–Mellin transform and the KCF is described in the second section. The estimation of the object translation and the confidence of tracking results is introduced in the third section. The fourth section presents the weighted object searching method based on the histogram and the variance. The framework of our proposed method is shown in the third section. The experiments on the online object tracking benchmark (OTB) are carried out, and the results and analysis are given in the sixth section. Finally, we give a brief summary of this article.

Scale and rotation estimation

Varied image registration techniques recover the rotation, translation, and scale parameters to align two images of the same object or of the same scene acquired from different geometric viewpoints, at different time, or by different image sensors. In object tracking, the object and its surroundings can be regarded to be approximately stationary in a short period. This assumption will hold in most cases considering that the object moves slow relative to the video frame rate, and the appearance of any adjacent two frames has little changes. So many popular and successful image registration methods such as the Fourier–Mellin transform\textsuperscript{24–26} can be applied to estimate the scale and rotation for object tracking. However, only two independent images of the same object or of the same scene acquired when the tracker fails. The proposed algorithm will greatly improve the long-term object tracking ability of security robots in complex environments.

The rest of this article is organized as follows. The proposed scale and rotation estimation based on the Fourier–Mellin transform and the KCF is described in the second section. The estimation of the object translation and the confidence of tracking results is introduced in the third section. The fourth section presents the weighted object searching method based on the histogram and the variance. The framework of our proposed method is shown in the third section. The experiments on the online object tracking benchmark (OTB) are carried out, and the results and analysis are given in the sixth section. Finally, we give a brief summary of this article.

Scale and rotation estimation

Varied image registration techniques recover the rotation, translation, and scale parameters to align two images of the same object or of the same scene acquired from different geometric viewpoints, at different time, or by different image sensors. In object tracking, the object and its surroundings can be regarded to be approximately stationary in a short period. This assumption will hold in most cases considering that the object moves slow relative to the video frame rate, and the appearance of any adjacent two frames has little changes. So many popular and successful image registration methods such as the Fourier–Mellin transform\textsuperscript{24–26} can be applied to estimate the scale and rotation for object tracking. However, only two independent images of the same object or of the same scene acquired when the tracker fails. The proposed algorithm will greatly improve the long-term object tracking ability of security robots in complex environments.

The rest of this article is organized as follows. The proposed scale and rotation estimation based on the Fourier–Mellin transform and the KCF is described in the second section. The estimation of the object translation and the confidence of tracking results is introduced in the third section. The fourth section presents the weighted object searching method based on the histogram and the variance. The framework of our proposed method is shown in the third section. The experiments on the online object tracking benchmark (OTB) are carried out, and the results and analysis are given in the sixth section. Finally, we give a brief summary of this article.

Fourier–Mellin transform

The scale changes and rotation motion of the tracked object in the Cartesian coordinate domain correspond to purely translational motion in the log-polar domain. Similar with the human visual system, the log-polar transformation is performed through a space-variant sampling strategy with the sampling period increasing almost linearly with the distance from the transformation center to actual pixel coordinates of the samples. Therefore, the log-polar transformation is very sensitive to changes of the transformation center, namely to the translational motion of the tracked object in object tracking. The scale and rotation parameters can be estimated in the log-polar image in the time domain. However, the translational motion of the tracked object usually prompts changes of the transformation center in log-polar transformation, resulting in the inaccurate estimation of the scale and rotation parameters. In order to improve the accuracy and reduce the impact of the transformation center, we can firstly estimate the translation of the tracked object in the original image, then obtain the scale and rotation parameters in the log-polar image. However, scale changes and rotation motion of the tracked object sometimes make the estimated translation inaccurate, so it is difficult to estimate the scale and rotation parameters in the time domain. According to the shift property of the Fourier transform, the phase of the cross-power spectrum is equivalent to the phase difference between the two images that differ only by a displacement; thus, bilateral magnitude of the spectrums is likewise. To avoid the impact of the transformation center, the scale and rotation parameters can be estimated in the frequency domain. In this article, the Fourier–Mellin transform in the image registration community is used to estimate the scale and rotation parameters in the log-polar images of the magnitude spectrums.

Let us consider the image patch $T(x, y)$ which is a rotated, scaled, and translated replica of an any image patch $s(x, y)$. We assume that the translation vector, the scale, and the rotated angle are $(\Delta x, \Delta y)$, $\alpha$, and $\Delta \theta$, respectively. According to the Fourier property, the Fourier transforms of $s(x, y)$ and $T(x, y)$ are related by

$$T(u, v) = \frac{e^{-j2\pi(u\Delta x + v\Delta y)}}{\alpha^2} S \left( \frac{u \cos \Delta \theta - v \sin \Delta \theta}{\alpha} , \frac{u \sin \Delta \theta + v \cos \Delta \theta}{\alpha} \right)$$

(1)

Ignoring the multiplication factor $| \frac{e^{-j2\pi(u\Delta x + v\Delta y)}}{\alpha^2} |$, we apply a log-polar transformation to the magnitude spectrum $|T(u, v)|$ and $|S(u, v)|$ and obtain log-polar images $M_t$ and $M_s$ of the magnitude spectrums. In this article, the log-polar image of the magnitude spectrum is referred to as the FMT feature. $M_t$ and $M_s$ are related by

$$M_t(\log(\rho), \theta_t) = M_s(\log(\rho) - \log(\alpha), \theta_t + \Delta \theta)$$

(2)

where $\rho = \sqrt{u^2 + v^2}$ and $\theta_t = \arctan \frac{v}{u}$. Using $(\varepsilon, \eta)$ to represent $(\log(\rho), \theta_t)$, we can write equation (2) as

$$M_t(\varepsilon, \eta) = M_s(\varepsilon - \Delta \varepsilon, \eta - \Delta \eta)$$

(3)

where $\Delta \varepsilon = \log(\alpha)$ and $\Delta \eta = -\Delta \theta$. From equation (3), we know that scale changes and rotation motion between two images in the Cartesian coordinate system are
expressed as pure translations in the log-polar space of the magnitude spectrum.

In object tracking, the FMT features are extracted as follows:

1. Perform a Fourier transform on the input image and obtain the magnitude spectrum.
2. Apply a high-pass emphasis filter on the magnitude spectrum.
3. Apply a log-polar transformation on the magnitude spectrum.

A high-pass filter aims to retain the high-frequency information within an image while reducing the low-frequency information, which emphasizes the edge and contour features being in the ascendancy in scale and rotation estimation. A simple high-pass emphasis filter is used with the transfer function.

\[ H(u, v) = \left(1.0 - X(u, v)\right) \times \left(2.0 - X(u, v)\right) \]  

where \( X(u, v) = \left[\cos\left(\pi\left(\frac{u}{M} - 0.5\right)\right) \cos\left(\pi\left(\frac{v}{N} - 0.5\right)\right)\right] \); \( W \) and \( H \) are the width and the height of the image to be filtered, respectively.

**Kernelized correlation filter**

With the circulant samples and kernel matrices, the kernel ridge regression model of a sample \( z \) can be described as \( f(z) = \sum_{i=1}^{n} w_i \phi(x_i, z) \), where \( x_i, i = 1, 2, ..., n \) are training samples and \( \phi(x) \) is a kernel function. The training samples are cyclic shifts of a base sample \( x \). \( \phi(x) \) is used to map samples into a nonlinear feature space. The solution of the kernel ridge regression can be expressed as

\[ \hat{w} = \left( K + \alpha I \right)^{-1} y \]  

where \( y \) is the regression target of training samples. \( x_i, i = 1, 2, ..., n \). \( K \) is a circulant kernel matrix with element \( K_{ij} = \phi(x_i, x_j) \). Equation (5) can be solved in the frequency domain as

\[ \hat{w} = \left[ \frac{1}{k^{xx} + \alpha \delta} \right]^{\ast} \circ \hat{y} = \frac{\hat{y}}{k^{xx} + \alpha \delta} \]  

where the hat ‘\(^\ast\)’ is a shorthand for the discrete Fourier transform of a vector, and \( \circ \) indicates the element-wise product. \( k^{xx} \) is the first row of the kernel matrix \( K \), and \( k^{xx} \) is real symmetric, so \( \hat{k}^{xx} \) is equal to \( (\hat{k}^{xx})^{\ast} \). To detect the object of interest, we typically wish to evaluate \( f(z_i), z_i, i = 1, 2, ..., n \) on several candidate samples. These samples can be generated from the base sample \( z \) using cyclic shifts. \( f(z) \) can be solved as

\[ f(z) = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i \phi(x_i, z_j) = (K^z)^T w \]  

Notice that \( f(z) \) is a vector, containing the outputs for \( z_i, i = 1, 2, ..., n \). Equation (7) can be solved in the frequency domain as

\[ \hat{f}(z) = \hat{k}^{xz} \circ \hat{w} \]  

where \( k^{xz} \) is the first row of the kernel matrix \( K^z \).

**Scale and rotation estimation**

In object tracking, the appearance of a tracked object is modeled using a correlation filter \( w \) trained on an image patch \( x_{0,0} \). The width and the height of the image patch \( x_{0,0} \) are \( M \) and \( N \), respectively, and we can totally obtain \( M \times N \) samples \( x_{m,n}, (m,n) \in \{0, 1, ..., M-1\} \times \{0, 1, ..., N-1\} \) by cyclic shifts of the base \( x_{0,0} \). Each sample is assigned with a Gauss function label \( y(m,n) \). The kernel ridge regression model is \( f(z) = \sum_{m,n} w_{m,n} \phi(x_{m,n}, z) \), where \( \phi \) is kernel function. The loss function is defined as

\[ L = \min \left\{ \sum_{m,n} ||\phi(x_{m,n}) \cdot w - y(m,n)||^2 + \alpha ||w||^2 \right\} \]  

In terms of scale and rotation estimation, \( x_{0,0} \) is actually the FMT feature in the frequency domain. The training samples in equation (9) are cyclic shifts of \( x_{0,0} \). The model learning based on the KCF and the Fourier–Mellin transform is shown in Figure 1. To minimize the effects of leakage in discrete Fourier transform, the sample \( x_{0,0} \) is weighted by the Hanning window, which smooths
discontinuities at boundaries of $x_{0,0}$ caused by the cyclic assumption.

In object tracking, the FMT feature $z$ with the same size as $x_{0,0}$ is extracted from an image patch. Candidate samples $z_i, i = 1, 2, ..., n$ are modeled from $z$ by cyclic shifts. According to equation (8), we can obtain the response of candidate samples in the frequency domain. Finally, the response map is computed in the time domain as

$$
\tilde{y} = \mathcal{F}^{-1}(\hat{f}(z)) = \mathcal{F}^{-1}\left(\hat{k}^{xz} \odot \hat{w}\right) (10)
$$

Therefore, we can estimate the translation of the object on the log-polar space by searching for the location of the maximal value of $\tilde{y}$. The scale $\alpha$ and the rotated angle $\Delta \theta$ can be computed according to equation (3).

It is important for the regression model to be adaptive to changes in the appearance of the object. In the last frame, the regression model $M_{\alpha}(t-1)$ is composed of the template $\hat{x}(t-1)$ and the regression coefficient $\hat{w}(t-1)$. In the current frame, the translation, scale, and rotated angle of the object is found. An image patch is cropped out, to which the Fourier–Mellin transform is applied. The template $x(t)$ and the regression coefficient $\hat{w}(t)$ in the current frame are used to update $M_{\alpha}(t-1)$ as

$$
x(t) = \beta \hat{x}(t-1) + (1-\beta)x(t)$$

$$
\hat{w}(t) = \beta \hat{w}(t-1) + (1-\beta)\hat{w}(t) (11)
$$

where $\beta$ is the learning rate and $t$ is the index of the current frame.

**Estimation of the translation and the confidence of tracking results**

For object tracking, there are inevitable appearance variations over time caused by the scale, out-of-plane rotation, illumination, occlusions, deformations, and so on. The core of object tracking is to robustly estimate the location of the object in every frame under these challenging circumstances. Besides, it is critical for trackers to measure the confidence of tracking results robustly and accurately. In the study by Henriques et al., trackers exploited the trained template and model to measure the confidence of tracking results while estimating the location of the object. However, drifts of the object template or model updated online always exist. In the literature, trackers employed the template and model trained in the first frame to measure the confidence of tracking results in the following frames. In practical applications, both methods mentioned above cannot provide accurate confidence of tracking results. In this article, we dedicatedly train two KCFs to estimate the object’s translation and the confidence of tracking results, respectively.

The model learning for the estimation of the object’s translation and the confidence of tracking results based on the KCF are shown in Figure 2. The model learning and fast detection of the two filters are equivalent to that of the correlation filter for scale and rotation estimation. The details can be referred to equations (10) and (11). The HOG feature is highly robust to illumination variations and local deformations, which is widely applied in object tracking, and it hits the target that coming up with good tracking performance in previous applications. For the estimation of object translation, HOG features are extracted from an image patch cropped from the object and its surroundings, which are weighted by a Hanning window. For the estimation of the confidence of tracking results, we learn another discriminative regression model $M_c$ only from the most reliable object region. And a layer of spatial weights is not added on the extracted HOG features. The maximal value of $\tilde{y}$ represents the confidence of tracking results. To maintain the model stability and restrain its drifts, we redefine a threshold $T_u$ and update $M_c$ only if $\max(\tilde{y}) > T_u$.

**Weighted object searching based on the histogram and the variance**

Color histograms are invariant to the translation, rotation about an axis perpendicular to the image plane, and change slowly from different angles of view. Histograms for different objects can be different markedly. Simultaneously,
it can acquire the distribution of colors inside the object region. Thus, color histograms have been widely applied in the computer vision community. Let $I$ be an image. The colors in $I$ can be quantized into $n$ distinct color bins. Color histogram is a vector $H_I = [h_1, h_2, ..., h_n]$, where each element $h_i$ represents the number of pixels of the color bin $j$ in the image. In object tracking, we need to compute the histogram in the local region of an image, such as the object region. The histogram of the local region $\Omega$ can be represented as $H^\prime_\Omega$. The size of $\Omega$ is $M \times N$, and the number of color bins is $n$. The probability for each bin $b$ is modeled by the normalized histogram as

$$p(b) = \frac{H^\prime_\Omega(b)}{M \times N}, \quad b = 1, 2, ..., n$$  \hspace{1cm} (12)$$

Here we construct a histogram model $M_h = \{H_p, S_\sigma\}$ for the tracked object. The model can be used to search the object in case of tracking failures. $M_h$ is composed of the normalized histogram $H_p = \{P(b_1), P(b_2), ..., P(b_n)\}$ and the sum of the probability $S_\sigma$. $S_\sigma$ is calculated by

$$S_\sigma = \sum_{i=1}^{M \times N} P(b_x_i)(\Omega)$$  \hspace{1cm} (13)$$

where $b_x_i$ represents the color bin to which the pixel $x_i$ in $\Omega$ belongs.

If the tracker loses the object, it should be able to relocate the object in the whole image $I$, using the model $M_h$. If the pixel $x$ of the image $I$ belongs to the color bin $b_x$, the probability of the pixel $x$ coming from the tracked object is $P(x \in I_s) = P(b_x)$. Eventually, we obtain a likelihood image $L$ where each pixel $x$ represents the probability $P(x \in I_s)$. In the last frame, if the bounding box of the object is recorded as $w_i$ with the width $W$ and the height $H$, Lots of sliding windows at the same size as $w_i$ can be generated in the image $I_s$, which can be expressed as $w_i, i = 1, 2, ..., n$. The sum of the probability for each sliding window is

$$S(w_i) = \sum_{i=1}^{w_i \times H_i} P(b_x_i)(w_i), \quad i = 1, 2, ..., n$$  \hspace{1cm} (14)$$

To increase the calculation speed, the integral image of $L$ can be used to compute $\sum_{i=1}^{w_i \times H_i} P(b_x_i)(w)$. The weight of each sliding is calculated by

$$W_h(w_i) = \frac{|S(w_i) - S_\sigma|}{S_\sigma}, \quad i = 1, 2, ..., n$$  \hspace{1cm} (15)$$

In the histogram, only the probability distribution of each color in the image is taken into account, which is not enough to describe the tracked object. The variance as an important tool reflects how “spread out” a probability distribution. The variance is defined as the average squared deviation from the mean. Here, we exploit the variance to measure how far a set of colors in the image are spread out from their average value. The variance is very effective for weight sliding windows. We use the variance $V_\sigma$ of the object region as the model $M_v = \{V_\sigma\}$. The integral image can also be applied for expediting the calculation speed.\hspace{1cm} (16)

The weight of each sliding window is calculated by

$$W_v(w_i) = \frac{|V(w_i) - V_\sigma|}{V_\sigma}, \quad i = 1, 2, ..., n$$  \hspace{1cm} (16)$$

Both $W_h(w_i)$ and $W_v(w_i)$ are combined together as the final weight. It is computed as

$$W(w_i) = \frac{1}{\lambda W_h(w_i) + (1 - \lambda)W_v(w_i)}, \quad i = 1, 2, ..., n$$  \hspace{1cm} (17)$$

where $\lambda \in [0, 1]$ is a weighting factor. We normalize the weights of all sliding windows as

$$W(w_i) = \frac{W(w_i)}{\sum_{i=1}^{w_i \times H_i}} \quad i = 1, 2, ..., n$$  \hspace{1cm} (18)$$

where $W(w_i)$ is the probability $P(w_i)$ that the window $w_i$ is selected as a candidate sample. Based on Monte Carlo sampling methods, we can randomly choose some of sliding windows as candidate samples to be further processed to decide whether they contain the object or not, which can improve the detection efficiency.

When the tracker fails or suffers from large drifts, the object is likely to be around the location where the object lost. Therefore, the probability that sampled windows contain the tracked object can be boosted only using a limited searching region instead of the whole image. In this article, the searching region is set to $L$ times of the object area. For the weighted object searching module, the model $M_{hv} = \{M_h, M_v\}$, composed of the histogram model $M_h$ and the variance model $M_v$, needs to be updated online to adapt to visual appearance variations. Similar to the updating process of $M_c$, we update $M_{hv}$ only if $\max(\dot{y}) > T_u$.

$$\dot{H}_p = \gamma \dot{H}_p(t-1) + (1 - \gamma)H_p(t)$$

$$\dot{S}_\sigma = \gamma \dot{S}_\sigma(t-1) + (1 - \gamma)S_\sigma(t)$$

$$\dot{V}_\sigma = \gamma \dot{V}_\sigma(t-1) + (1 - \gamma)V_\sigma(t)$$  \hspace{1cm} (19)$$

**The proposed tracking method**

In this article, we train three different correlation filters to estimate the object’s translation, scale, and rotation, and the confidence of tracking results, in order to realize robust and accurate long-time object tracking. At the same time, the weighted object searching module based on the histogram and the variance is employed in case of tracking failures, which relocates the lost object. The architecture of the proposed RLOT with an adaptive scale and rotation estimation is shown in Figure 3. The proposed algorithm is described in Algorithm 1.
The KCF for scale and rotation estimation uses the FMT feature as input feature. The HOG feature is used to train the other two correlation filters for estimating the object’s translation and the confidence of tracking results, and its dimension is 31. We use a Gaussian kernel $\phi(a, b) = \exp(-\frac{1}{2\sigma^2}(a - b)^2)$ for the three filters. According to equations (8) and (10), the kernel correlation filters can be calculated in the frequency domain. In object tracking, we firstly rotate and scale the image region of the object and its surroundings using estimated scale and rotation parameters and then estimate the object’s translation by the model $M_t$. Thus, more accurate object location can be obtained. We compute the histogram model in the Lab color space and calculate the variance model in a gray image. The Lab color space is a color-opponent space with the dimension $L$ for lightness and the color-opponent dimensions $a$ and $b$ for the red-green and blue-yellow opponent color channels, and it describes all the colors visible to the human eye. Lab color space has been widely applied in the computer vision community. Each channel of the Lab color space is quantized into 32 color bins, so the dimension of the color histogram is $32^3 = 32768$. It is possible to maintain a high calculation speed for searching the lost object using integral images. For the proposed RLOT method, the confidence of tracking results will be above 0. If we set $T_c$ to be −1, the weighted object searching module will not be performed and the model $M_{ps}$ will also not be updated. Robust object tracking (ROT) is used to name the algorithm that removes the weighted searching module from RLOT.

**Experiments**

In this section, we thoroughly compare our proposed RLOT tracker and ROT tracker with state-of-the-art methods on the OTB data set that contains 100 video sequences. The OTB data set contains 11 types of visual tracking challenges: illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutters, and low resolution. These video sequences have been annotated manually. In this article, C++ is used to implement the proposed tracking algorithm, and all experiments are performed on a PC with 3.1 GHz i7-5557U CPU and 8 GB RAM.

**Experimental setup**

In experiments, the precision plot and the success plot are used to evaluate the trackers. The tracking precision is based on the center location error, which is defined as the Euclidean distance between the center location of the bounding box and the manually labeled ground truth. The precision plot shows the proportion of frames in which the center location error is within the given threshold. The bounding box overlap is used to estimate whether trackers are successful or not, which is defined by $O = \frac{|B_t \cap B_s|}{|B_t \cup B_s|}$, where $B_t$ and $B_s$ represent the tracked bounding box and the ground truth bounding box, respectively. $\cap$ and $\cup$ mean the intersection and union of two regions, respectively, and $|\cdot|$ denotes the number of pixels in a region. The success plot shows the ratio of successful frames where bounding...
box overlap surpasses the given threshold. To realize a more fair evaluation, the area under the curve of the precision plot and the success plot is used to rank the tracking algorithms.

The annotated ground truth bounding boxes of OTB data sets are standard rectangles (the boundaries of the rectangle are parallel with the boundaries of the image). For OTB data sets, only standard rectangles are used to estimate the performance. In this article, we use the proposed tracking method to evaluate scale and rotation parameters, coming up with tracking results represented as nonstandard rectangles. However, according to the evaluation method provided by OTB data sets, only the external rectangles of nonstandard rectangles can be used for computing bounding box overlap ratio and center location error. The diagrams of target annotations (blue), nonstandard rectangles from our trackers (red), and external rectangles of nonstandard rectangles (green) are shown in Figure 4. It is obvious that the successful rate may be reduced when external rectangles are used to estimate bounding box overlap ratio. However, center locations of external rectangles and nonstandard rectangles are close to each other, thus precision performance may be less affected. To improve the accuracy of performance evaluations, we use nonstandard rectangles and target annotations to directly evaluate the performance of RLOT and ROT. In the meantime, due to without considering rotation motion of the object, target annotations are standard rectangles and not exactly equivalent to actual target area, but the center locations of target annotations and actual target areas are close to each other. Finally, there are still small errors using our proposed evaluation method for estimating bounding box overlap ratio. However, compared to the evaluation method proposed in OTB data sets, the error has been greatly reduced. Center location error can be estimated accurately using our proposed method.

Three KCFs are designed for the RLOT tracking algorithm, and their parameters are shown in Table 1, where $n$ and $m$ mean the width and the height of the object, respectively, measured in HOG cells, and $d$ and $a$ refer to the width and the height of the FMT feature, respectively. For KCFs to estimate the translation, scale, and rotation, the target region and its surrounding region should be cropped out for feature extraction, correlation filters training, and object detection. In this article, the size of the cropped image patch is 2.8 times that of the tracked object area.

**Algorithm 1. Proposed tracking algorithm**

```plaintext
Input: Initial bounding box (tracking state) $S_0$
Output: Estimated tracking state $S_t = (x_t, y_t, \alpha_t, s_t)$, $(x_t, y_t)$, $\alpha_t$, and $s_t$ represent the object's location, scale and rotation, respectively.
1 Initialization: the correlation filter models $M_1$, $M_r$, and $M_c$; the histogram and variance model $M_{hv}$;
2 repeat
3   According to $S_{t-1}$, crop out an image patch $p_i$ in the $t_{th}$ image $I_t$, and estimate the new location $(\hat{x}_t, \hat{y}_t)$ by $M_t$;
4   Extract the FMT feature from $p_i$, and estimate the scale and rotation $\hat{\alpha}_t, \hat{s}_t$ by $M_{r,s}$;
5   Crop out a new image patch $p'_i$ in $I_t$ according to the state $(\hat{x}_t-1, \hat{y}_t-1, \hat{\alpha}_t, \hat{s}_t)$, and estimate the new location $(\tilde{x}', \tilde{y}')$ by $M_t$;
6   Crop out two image patches $p_i'$ and $p_t$, according to $(\tilde{x}', \tilde{y}', \hat{\alpha}_t, \hat{s}_t)$ and $(\hat{x}_t-1, \hat{y}_t-1, \hat{\alpha}_t-1, \hat{s}_t-1)$, respectively, and estimate both confidence $c'_t$, $c_t$ by $M_c$;
7   if $c'_t > c_t$ then
8       $S_t \leftarrow (\tilde{x}', \tilde{y}', \hat{\alpha}_t, \hat{s}_t)$;
9   else
10      $S_t \leftarrow (\hat{x}_t, \hat{y}_t, \hat{\alpha}_t-1, \hat{s}_t-1)$;
11   end
12   if max$(c'_t, c_t) < T_r$ then
13      Rotate the image $I_t$, and obtain candidate states $X$ by $M_{hv}$;
14      foreach $x'_i \in X$ do
15         Crop out the image patch according to $x'_i$, and compute the confidence $c'_i$ by $M_c$;
16         if max$(c'_i) > T_s$ then
17            $S_t \leftarrow x'_i$, where $i = \arg \min_c(c'_i)$;
18      end
19   end
20   Update models $M_t$, $M_{r,s}$;
21   if max$(c'_t, c_t) > T_t$ then
22      Update models $M_{hv}$, $M_c$;
23 until End of image sequences;
```

**Figure 4.** The diagram of target annotations (blue), nonstandard rectangles from our trackers (red), and external rectangles of nonstandard rectangles (green).

**Table 1.** The parameters of three kernelized correlation filters.

| KCF parameters | $M_l$ | $M_c$ | $M_r$ |
|----------------|-------|-------|-------|
| Bandwidth of gauss kernel $\sigma$ | 0.6 | 0.6 | 0.4 |
| Learning rate $\beta$ | 0.012 | 0.012 | 0.075 |
| Bandwidth of gauss labels $s$ | $0.125\sqrt{mn}$ | $0.125\sqrt{mn}$ | $0.075(d, a)$ |
| Regularization $\alpha$ | $10^{-4}$ | $10^{-4}$ | $5 \times 10^{-5}$ |
| Padding | 2.8 | 1 | 2.8 |

KCF: kernelized correlation filter.
From Algorithm 1, a few parameters need to be set in particular. $T_r$, being used to activate the weighted object searching module, is set to 0.25. We set $T_d$ to 0.4 to decide whether the relocated tracked object is accepted and believed or not. We set $T_u$ to 0.4 to update the models $M_{hv}$ and $M_c$. The number of the sampled candidate sliding windows is set to 150. The weighting factor $\lambda$ in equation (17) is set to 0.5. In this article, the searching region parameter $L$ is set to 4 for the weighted object searching module. When ROT is running, we set $T_r$ to $-1$, and the model $M_{hv}$ will not be updated.

Quantitative evaluation

In this section, we evaluate quantitatively RLOT and ROT in comparison with some state-of-the-art tracking methods. The OTB has provided the tracking results of 29 trackers, including TLD, Frag, Struct, CT, SCM, ASLA, CXT, and so on. Besides, we also compare KCF, SAMF, DSST, SRDCF, Staple, and LCT. In order to make a fair and comprehensive comparison, we give the precision plot and the success plot with three standards: the one-pass evaluation (OPE), the temporal robustness evaluation (TRE), and the spatial robustness evaluation (SRE). The OPE refers to that a tracker runs throughout the entire sequence with the initialization from the ground truth location in the first frame. However, it cannot evaluate the impact of the initialization on the performance of trackers. To analyze a tracker’s temporal and spatial robustness, the TRE and the SRE introduce temporal (i.e., starting at different frames) and spatial (i.e., starting by different bounding boxes) correlated perturbations to the initialization, respectively.

The precision plots and success plots of our proposed tracking method and some state-of-the-art tracking methods, evaluated by the OPE, SRE, and TRE standards, are shown in Figure 5, where only the top 10 trackers are presented for clarity. SRDCF and Staple attain superior performance. Our proposed RLOT hits the target of robust object tracking, although there is a slight performance degradation compared to SRDCF and Staple. However, due to the lack of a detection module, ROT performs worse than RLOT. It achieves approximate performance compared to SAMF and LCT and exceeds DSST and KCF. With the OPE standard, RLOT obtains the second highest average precision performance for 0.006 (0.8%) lower than that of SRDCF. Simultaneously, its successful rate is close to that of Staple, and only 0.02 (4.3%) lower than that of SRDCF. ROT acquires similar results with SAMF and LCT in the precision and the successful rate. With the SRE standard, our RLOT realizes approximate precision performance compared to SRDCF and Staple. However, its success score ratio is lower than that of SRDCF, Staple, and SAMF but higher than that of LCT. In the same situations, ROT is also sensitive to spatial correlated perturbations. The imprecise initial bounding box results in the inaccurate estimation of rotation and scale parameters, which

![Figure 5. Precision plots and success plots of tracking algorithms evaluated by the OPE, SRE, and TRE standards, where only the top 10 trackers are presented for clarity. OPE: one-pass evaluation; SRE: spatial robustness evaluation; TRE: temporal robustness evaluation.](image)
eventually leads to a decrease in the tracking performance. For the TRE standard, RLOT reaches the third highest average precision performance which is 0.005 (0.68\%) lower than that of SRDCF. Its successful rate is the same as SAMF, which is slightly worse than Staple and SRDCF. ROT accomplishes similar tracking performance with LCT. RLOT and ROT are given the same parameters, but RLOT contains a unique detection module. RLOT outperforms ROT in the precision and the successful rate. It is obvious that the detection module is effective for object tracking.

The OTB contains 11 different types of tracking challenges, and these challenges have been annotated for each sequence. Thus, the subset with different dominant challenges can be constructed to analyze the performance of trackers under different challenges. RLOT has two main advantages: estimating rotation and scale parameters, and redetecting lost objects. Especially, the OTB data set, which contains in-plane/out-of-plane rotation and scale variation challenges, can be used to evaluate the effectiveness of the proposed rotation and scale estimation algorithm. In object tracking, the tracker is sometimes inevitable to lose its object or suffer from a large drift when it suffers from partial or full occlusion. Thus, these sequences can be used to estimate the performance of the redetection module by sequence-specific challenges including partial or full occlusion. The tracking results under different challenges including out-of-plane rotation, in-plane rotation, scale variation, and occlusion are shown in Figure 6. It is clear that our proposed RLOT gets the best performance under challenges of both in-plane rotation and out-of-plane rotation compared to the existing methods. In case of in-plane rotation, ROT’s performance that ranks second is close to that of LCT. In case of out-of-plane rotation, ROT has a slight decrease in the performance but still surpasses KCF and DSST. In case of scale variation, RLOT achieves the highest and second-class performance in the precision and the successful rate, respectively. ROT obtains the similar tracking results with SAMF. According to the above discussions, the proposed scale and rotation estimation method is quite effective, which can help to improve tracking performance. In case of occlusion, RLOT achieves better tracking performance than ROT, which also verifies the effectiveness of the detection module for object tracking.

Qualitative evaluation

In this subsection, we compare qualitatively RLOT and ROT with current mainstream tracking algorithms, including a correlation filter tracker (SRDCF), a correlation filter tracker with a redetection module (LCT), a tracker based on tracking by detection (Struck), and a tracker with the combination of tracking, learning, and detection (TLD). We focus on 11 representative image sequences selected from the OTB data set. These sequences almost contain all the tracking challenges, as shown in Table 2. RLOT, ROT, SRDCF, LCT, Struck, and TLD are performed on the 11 image sequences, and their tracking results are demonstrated in Figure 7. In SRDCF, several adaptive and effective correlation filters are learned using some powerful features including HOG, color naming, and raw pixels; and a spatial regularization component is introduced in the learning to address boundary effects for a periodic assumption. To estimate the object’s scale, SRDCF refers to a multiple scales searching strategy, which firstly samples the target with different scales and then resizes the samples.

Figure 6. Precision plots and success rate plots of tracking algorithms evaluated by the OPE standard under different visual tracking challenges, where only the top 10 trackers are presented for clarity. OPE: one-pass evaluation.
into a fixed size to compare with the learnt model at each frame. Without the rotation estimation, SRDCF performs well in dealing with most of visual tracking challenges except rotational motion (MotorRolling). In LCT, HOG features are employed to learn correlation filters. In addition, a random fern classifier is used to redetect the tracked object in case of tracking failures. Without the rotation estimation, LCT is not robust to in-plane rotation (MotorRolling). In the meantime, its redetection module suffers from drifts or even failures in cluttered environments, especially when the appearance of the background is similar to that of the tracked object (Soccer). LCT performs well under other visual challenges.

Struck does not perform well in handling scale variations (CarScale and Dog1), deformation (Tiger1), and heavy occlusion (Soccer). It drifts when the object undergoes out-of-plane rotation (David), dark illumination, and background clutters (Singer2 and Soccer) and fails under fast in-plane rotation (MotorRolling). Struck achieves good tracking performance under other visual tracking challenges. In TLD, tracking, learning, and detection are combined to make up their own deficiencies. A tracker can help to provide weakly labeled training samples with structure constraints for a detector which can recover the tracker when large drifts happen or the tracker fails in turn. However, the tracking module of TLD is based on the optical flow, which is not robust to visual appearance variations. The tracking module usually suffers from large drifts, even failures under certain tracking challenges; thus, it is not able to provide its temporal motion cues to the detector. In the meantime, the detector is less effective in handling background clutters, dark illumination conditions, and rotation motion. Due to the reasons mentioned above, TLD does not achieve good tracking performance on image sequences.

Table 2. The visual tracking challenges of the selected 11 image sequences from the OTB.$^a$

| Illumination variation | Out-of-plane rotation | Scale variation | Occlusion | Deformation | Motion blur | Fast motion | In-plane rotation | Out-of-view | Background clutters | Low resolution |
|------------------------|-----------------------|-----------------|-----------|-------------|-------------|-------------|-------------------|-------------|---------------------|---------------|
| CarScale               | 0                     | 1               | 1         | 1           | 0           | 0           | 0                 | 0           | 0                   | 0             |
| MotorRolling           | 1                     | 0               | 1         | 0           | 0           | 1           | 0                 | 0           | 0                   | 0             |
| Dog1                   | 0                     | 1               | 1         | 0           | 0           | 0           | 0                 | 0           | 0                   | 0             |
| FaceOcc2               | 1                     | 1               | 0         | 1           | 0           | 0           | 0                 | 0           | 1                   | 0             |
| Shaking                | 1                     | 1               | 0         | 1           | 0           | 0           | 0                 | 0           | 0                   | 0             |
| Singer2                | 1                     | 1               | 0         | 0           | 0           | 1           | 0                 | 0           | 0                   | 0             |
| David                  | 1                     | 1               | 1         | 1           | 0           | 0           | 0                 | 0           | 0                   | 0             |
| Jogging-2              | 0                     | 1               | 0         | 1           | 0           | 0           | 0                 | 0           | 0                   | 0             |
| Lemming                | 1                     | 1               | 0         | 1           | 0           | 0           | 0                 | 0           | 0                   | 0             |
| Tiger1                 | 1                     | 1               | 0         | 1           | 0           | 0           | 1                 | 0           | 0                   | 0             |
| Soccer                 | 1                     | 1               | 1         | 0           | 0           | 1           | 0                 | 0           | 0                   | 0             |

OTB: object tracking benchmark.

$^a$The value 0 indicates that the sequence does not contain the corresponding challenge, while 1 indicates that the sequence contains the corresponding challenge.

Real-time performance

The real-time performance of different tracking algorithms including our proposed RLOT, SRDCF, LCT, KCF, DSST, TLD, and Struck is compared in Table 3. The average frames per second (fps) is evaluated using all the sequences on the OTB. Obviously, the KCF tracker based on the correlation filter can be run with the highest fps and followed by the proposed RLOT tracker. The KCF tracker has the best real-time performance because it uses a simple KCF to track the object, which makes it has a low computational complexity and a limited tracking performance. Most other trackers, including the proposed RLOT, are based on the KCF, which improved the tracking performance, increasing the computational complexity. The fps of RLOT is lower than that of the KCF due to the additional scale and rotation estimation and the weighted object searching module. The average processing fps of SRDCF is the lowest (only 4 fps). This is because SRDCF employs
Figure 7. Tracking results using our proposed RLOT, ROT, SRDCF, LCT, KCF, Struck, and TLD on 11 OTB image sequences (from top to down: David, CarScale, Dog1, FaceOcc2, Jogging-2, Lemming, MotorRolling, Shaking, Singer2, Tiger1, and Soccer). RLOT: robust long-term object tracking; OTB: object tracking benchmark.
more features, including HOG, color naming, and raw pixels, to learn the object filter. The average processing fps of SRDCF, LCT, DSST, TLD, and Struck is less than 30. In practical applications, RLOT can be run at about 36 fps, so real-time requirements can be met.

**Conclusions**

When the security robots perform surveillance or reconnaissance tasks, the scale of the tracked object may change, the object may rotate, and the object tracking may fail. In this article, an RLOT approach with adaptive scale and rotation estimation is proposed. Our method can be divided into three parts: the estimation of the scale and rotation based on the Fourier–Mellin transform, the estimation of the object’s translation and the confidence of tracking results based on correlation filters, and the weighted object searching module based on the histogram and the variance. Our method is able to estimate the translation, scale, and rotation and the confidence of tracking results accurately and efficiently. The weighted object searching module can be used to redetect the lost object in case of tracking failures. Experiments are performed on the OTB, and comparisons are made among our proposed RLOT and state-of-the-art trackers. The experimental results validate the effectiveness and superiority of the proposed tracking algorithm.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is supported by National Key R&D Program of China (no YFC20170806503) and the National Science Foundation of China (no 61773393 and no U1813205).

**ORCID iD**

Huimin Lu  
https://orcid.org/0000-0002-6375-581X

**Table 3.** The average frame rates of different object tracking algorithms.

| Tracking algorithm | Frames per second (fps) |
|--------------------|------------------------|
| RLOT               | 36                     |
| SRDCF             | 4                      |
| LCT               | 27.4                   |
| KCF               | 167                    |
| DSST              | 27                     |
| TLD               | 20                     |
| Struck            | 28                     |

RLOT: robust long-term object tracking; KCF: kernelized correlation filter.

**References**

1. Yilmaz A, Javed O, and Shah M. Object tracking: a survey. *Acm Comput Surv* 2006; 38(4): 13: 1–13: 45.
2. Hu W, Tan T, Wang L, et al. A survey on visual surveillance of object motion and behaviors. *IEEE T Syst, Man, Cy-C* 2004; 34(3): 334–352.
3. Li X, Hu W, Shen C, et al. A survey of appearance models in visual object tracking. *ACM T Intell Syst Tec* 2013; 4(4): 58: 1–58: 48.
4. Wu Y, Lim J, and Yang MH. Online object tracking: a benchmark. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, San Juan, Puerto Rico, 17–19 June 1997, pp. 2411–2418. DOI:10.1109/CVPR.2013.312.
5. Kristan M, Matas J, Leonardis A, et al. The visual object tracking vot2015 challenge results. In: *Proceedings of the IEEE international conference on computer vision workshop*, Santiago, Chile, 7–13 December, 2015, pp. 564–586. DOI:10.1109/ICCVW.2015.79.
6. Wu Y, Lim J, and Yang MH. Object tracking benchmark. *IEEE Trans Pattern Anal Mach Intell* 2015; 37(9): 1834–1848.
7. Wang H, Suter D, Schindler K, et al. Adaptive object tracking based on an effective appearance filter. *IEEE Trans Pattern Anal Mach Intell* 2007; 29(9): 1661–1667.
8. Black MJ and Jepson AD. Eigentracking: robust matching and tracking of articulated objects using a view-based representation. In: *Proceedings of the 4th European conference on computer vision*, Cambridge, United kingdom, 15–18 April 1996, pp. 329–342.
9. Chin TJ and Suter D. Incremental kernel principal component analysis. *IEEE Trans Image Process* 2007; 16(6): 1662–1674.
10. Grabner H, Grabner M, and Bischof H. Real-time tracking via on-line boosting. In: *Proceedings of the 2006 British machine vision conference*, Edinburgh, United kingdom, 4–7 September 2006, pp. 47–56.
11. Zhang K, Zhang L, and Yang MH. Fast compressive tracking. *IEEE Trans Pattern Anal Mach Intell* 2014; 36(10): 2002–2015.
12. Kalal Z, Mikolajczyk K, and Matas J. Tracking-learning-detection. *IEEE Trans Pattern Anal Mach Intell* 2012; 34(7): 1409–1422.
13. Bolme DS, Beveridge JR, Draper BA, et al. Visual object tracking using adaptive correlation filters. In: *Proceedings of the IEEE computer society conference on computer vision and pattern recognition*, San Francisco, CA, 13–18 June 2010, pp. 2544–2550.
14. Henriques JF, Caseiro R, Martins P, et al. High-speed tracking with kernelized correlation filters. *IEEE Trans Pattern Anal Mach Intell* 2015; 37(3): 583–596.
15. Li Y and Zhu J. A scale adaptive kernel correlation filter tracker with feature integration. In: *Proceedings of European conference on computer vision workshops*, pp. 254–265. Zurich: Springer International Publishing. DOI: 10.1007/978-3-319-16181-518. ISBN 978-3-319-16181-5.
16. Danelljan M, Hger G, Shahbaz Khan F, et al. Accurate scale estimation for robust visual tracking. In: Proceedings of the British machine vision conference. BMVA Press. DOI: 10.5244/C.28.65.
17. Danelljan M, Hger G, Khan FS, et al. Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the 2015 IEEE international conference on computer vision, Santiago de Chile, Chile, 11–18 December 2015, pp. 4310–4318.
18. Ma J, Luo H, Hui B, et al. Robust scale adaptive tracking by combining correlation filters with sequential Monte Carlo. Sensors 2017; 17(3): 512.
19. Hu Q, Guo Y, Lin Z, et al. Fast correlation tracking using low-dimensional scale filter and local search strategy. IEEE Access 2017; 5: 8568–8578.
20. Zhou X, Liu X, Yang C, et al. Multi-channel features spatio-temporal context learning for visual tracking. IEEE Access 2017; 5: 12856–12864.
21. Bertinetto L, Valmadre J, Golodetz S, et al. Staple: Complementary learners for real-time tracking. In: Proceedings of the 2016 IEEE conference on computer vision and pattern recognition, Las Vegas, NV, 27 June–30 June 2016, pp. 1401–1409.
22. Zuo W, Wu X, Lin L, et al. Learning support correlation filters for visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 2018; 41(5): 1158–1172.
23. Ma C, Yang X, Zhang C, et al. Long-term correlation tracking. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR), Boston, MA, 7–12 June 2015, pp. 5388–5396.
24. Zokai S and Wolberg G. Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations. IEEE T Image Process 2005; 14(10): 1422–1434.
25. Reddy BS and Chatterji BN. An FFT-based technique for translation, rotation, and scale-invariant image registration. IEEE T Image Process 1996; 5(8): 1266–1271.
26. Sarvaiya JN, Patnaik S, and Kothari K. Image registration using log polar transform and phase correlation to recover higher scale. J Pattern Recognit Res 2012; 7(1): 90–105.
27. Zhong W, Lu H, and Yang MH. Robust object tracking via sparsity-based collaborative model. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Providence, RI, 16–21 June 2012, pp. 1838–1845.
28. Santner J, Leistner C, Saffari A, et al. Prost: Parallel robust online simple tracking. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition, San Francisco, CA, 13–18 June 2010, pp. 723–730.
29. Swain MJ and Ballard DH. Indexing via color histograms. Berlin: Springer Berlin Heidelberg, 1992, pp. 261–273. ISBN 978-3-642-77225-2.
30. Adam A, Rivlin E, and Shimshoni I. Robust fragments-based tracking using the integral histogram. In: Proceedings of the IEEE computer society conference on computer Vision and Pattern Recognition, Vol. 1, New York, NY, 17–22 June 2006, pp. 798–805.
31. Hare S, Golodetz S, Saffari A, et al. Struck: Structured output tracking with kernels. IEEE Trans Pattern Anal Mach Intell 2016; 38(10): 2096–2109.
32. Wang G, Qin X, Zhong F, et al. Visual tracking via sparse and local linear coding. IEEE Trans Image Process 2015; 24(11): 3796–3809.
33. Dinh TB, Vo N, and Medioni G. Context tracker: exploring supporters and distracters in unconstrained environments. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Providence, RI, 20–25 June 2011, pp. 1177–1184.