Extended kernel correlation filter for abrupt motion tracking

Huanlong Zhang[1], Jianwei Zhang[2*], Qinge Wu[1], Xiaoliang Qian[1], Tong Zhou[1] and Hengcheng FU[1]

[1]College of electric and information engineering, Zhengzhou University of Light Industry, Zhengzhou 450002,P.R. China

[e-mail:zhl_lit@163.com]

[2]Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou 450002, P.R. China

*Corresponding author: Jianwei Zhang

Received February 19, 2017; revised April 19, 2017; accepted May 16, 2017; published September 30, 2017

Abstract

The Kernelized Correlation Filters (KCF) tracker has caused the extensive concern in recent years because of the high efficiency. Numerous improvements have been made successively. However, due to the abrupt motion between the consecutive image frames, these methods cannot track object well. To cope with the problem, we propose an extended KCF tracker based on swarm intelligence method. Unlike existing KCF-based trackers, we firstly introduce a swarm-based sampling method to KCF tracker and design a unified framework to track smooth or abrupt motion simultaneously. Secondly, we propose a global motion estimation method, where the exploration factor is constructed to search the whole state space so as to adapt abrupt motion. Finally, we give an adaptive threshold in light of confidence map, which ensures the accuracy of the motion estimation strategy. Extensive experimental results in both quantitative and qualitative measures demonstrate the effectiveness of our proposed method in tracking abrupt motion.

Keywords KCF, The Simulated Annealing, Swarm intelligence, Abrupt Motion

[1]This work is supported by National Natural Science Foundation of China (61503173,61672471,61501407), Key Science and Technology Program of Henan Province (172102210062，162102210060) and Doctor fund project of Zhengzhou University of Light Industry(2016BSJJ002,2016BSJJ006)

https://doi.org/10.3837/tiis.2017.09.014 **ISSN : 1976-7277**

1. Introduction

Visual object tracking is a hot topic in computer vision and related fields for its various applications ranging from security and surveillance, medical imaging, robotics, to human computer interaction. Up to now, researchers have made significant progress. There still exist many challenging problems, such as motion blur, partial occlusions, illumination variation, fast motion, etc. To further improve the ability of the tracker, numerous algorithms have been proposed, some of which adopt generative model [1-4], while the others adopt discriminative model [5,6].

Recently, researchers attach great importance to correlation filter (CF), which has already been applied in visual tracking successfully. The basic idea of CF is that it can transform the time-consuming convolution operation in time domain to an element-wise multiplication in Fourier domain. Trackers based on CF locate the tracking object by identifying the location of maximal correlation response. Using correlation filters for tracking starts from Bolme et al.[7], where the filter is trained directly in the Fourier domain using an adaptive framework, reaching a runtime of 600-700 FPS. Then, Henriques et al.[8] utilize the circulant matrix produced by a base image patch to design a kernelized correlation filter(KCF), which achieves a better performance. In addition, numerous improvements have been made. Tang et al.[9]adopt a multi-kernel correlation filter (MKCF), Danelljan et al.[10]and Li et al.[11]use the adaptive multi-scale correlation filters, for handling scale variations. Zhang et al.[12]incorporate context information into filter for tracking accuracy. Liu et al.[13], Li et al.[14]and Liu et al.[15]bring part-based strategy to trackers, which are less sensitive to partial occlusion.

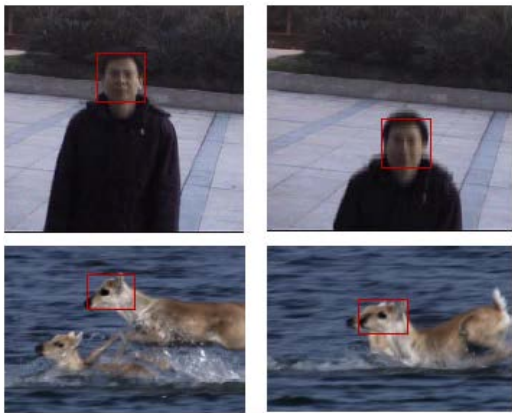


Fig. 1. The abrupt motion tracking results

Although achieved the appealing results both in accuracy and robustness, these correlation filter based trackers cannot deal with Large displacement motion well. As shown **Fig. 1**, The tracking object undergoes a large displacement motion between the consecutive image frames.

On this occasion, these trackers may be failure because of a local optimum. However, in real world abrupt motion is a very common phenomenon resulting from fast motion, low frame rate (LFR) and camera shaking et al.

To alleviate the above problems, we present an extended kernel correlation filter tracker based on swarm intelligence. The initial motivation for our method is to enable KCF to adjust the location of the base image patch with the help of the improved SA. Since SA can escape the local minimum to obtain global optimum, when the traditional KCF tracker is failure, SA provides a more reliable base image patch to generate the circulant matrix so as to get better object candidates for abrupt motion. The main contribution of our work includes three aspects: 1) we design a unified framework to track smooth or abrupt motion simultaneously;2) we give a global motion estimation method, in which the exploration factor is use to cover the whole state space;3) we adopt an adaptive threshold to switch model for coping with abrupt motion.

2. Related Work

Around these challenging tracking problems, there have been many attractive research results in visual tracking over the past decade. Generally speaking, trackers have been divided to the two main categories in the literature (generative and discriminative). Here, the methods that are most relevant to our tracker are introduced for describing the proposed method well.

Generative visual tracking tries to design an appearance model to represent the target, the goal is to search for the target candidate location that has the most similar feature to the model. Numerous representative trackers have been reported successively. COMANICIU et al.[16]build a color histogram-based visual representation regularized by a spatially smooth isotropic kernel. Using the Bhattacharyya coefficient as the similarity function, and mean shift procedure is firstly performed for object localization. Ross et al.[17]propose the incremental visual tracking (IVT) framework, the method learns the dynamic appearance of the target via incremental principal component analysis (PCA). Wang et al. [18] apply the linear representation to maintain holistic appearance model and propose the probability continuous outlier model to handle partial occlusions. Kwon et al.[19]decompose the tracking problem into two basic motion models and four observation models and allow different basic models to interact for abrupt motion. Mei and Ling et al.[20]firstly bring sparse representation into visual tracking and achieve significant performance. The method has attracted much attention and rapidly become a research hot problem. Bao et al. [21] use a fast numerical solver based on the APG approach to solve the L1 norm minimization problem. Zhou et al.[22] propose a Graph Regularized and Locality-constrained Coding (GRLC) method and apply it to visual tracking combined with sparse theory. To further improve tracking performance, Zhang et al. [23-25] propose numerous improved trackers based on sparse representation.

Different from the above frameworks, discriminative visual tracking formulates object as a binary classification problem. The method aims at finding a good classifier that can best

distinguish the target from background and is updated in time with appearance changes. Avidan et al.[26]first propose an offline SVM-based discriminative model for visual tracking. However, the method needs substantial prior training data in advance. On this occasion, the appearance changes are easy to tracking failure. Grabner et al. [27] adopt online AdaBoost to select discriminative features for object tracking. It's result is affected by background clutter and can easily drift. Babenko et al. [28] present a multiple instance learning (MIL) tracker. The method uses sample bag to instead the single sample for training a better classifier. Recently, deep learning has gained a lot of interest in computer vision due to the strong capability in various applications. Wang et al. [29] firstly attempt to apply it into visual tracking. In this method, a deep autoencoder network is adopted to extract image feature and then a particle filtering approach is used to track object. However, the pretraining of tracker may not be very suitable for object tracking in time. Other improved trackers [30-32] have been presented, they show good tracking performance.

Our proposed method is closely related to the correlational filter-based trackers, which have attracted considerable attention to visual tracking and achieved top performance due to its computational efficiency and robustness. KCF tracker is first introduced in [8], in which the HOG feature is adopted to represent object's appearance and the convolution in time is replaced by the dot product in frequency with the help of the circulant matrix. However, KCF tracker employs the template with the fixed size so that the tracker is not able handle the scale changes of a target. In addition, KCF tracker has not a mechanism to cope with partial occlusion. These problems all accelerate the development of tracking technologies. Compared with these methods, our tracker focuses on how to obtain a good circulant matrix when the target's displacement between successive image frames is large. To the best of our knowledge, few attempts have been made to improve the capability of KCF tracker for coping with abrupt motion.

3. Proposed tracking algorithm

In this section, we firstly review the kernelized correlation filter. Then, the improved SA is introduced. Finally, our tracking framework is given. Below we give a detail description of our tracker.

3.1 The KCF tracker

In KCF tracker, the circulant matrix X is designed by all cyclic shifts x_i based on the base image patch x_0 , which has an intriguing property that it can be diagonal by the discrete Fourier transform (DFT). This can be expressed as

$$X = Fdiag(\hat{x})F^H \tag{1}$$

Where F is a constant matrix that does not depend on x , and $\hat{x}$ denotes the DFT of the base

vector x. By obtaining the candidate samples form the circulant matrix, the classifier is trained to get the relation between $i-th$ input x_i and its label y_i. Suppose the relation takes the form $f(x_i)=y_i$, the classification problem is transformed to minimizing the objective function:

$$\min_{w}\sum_{i}L(f((w,x_i),y_i)+\lambda\|w\| \tag{2}$$

where w denotes the parameter, λ is a regularization parameter, and $L(\cdot)$ is a loss function.

3.1.1 Linear Ridge Regression

For the function f(x_i), it can be a linear operation. As the problem of linear ridge regression, we can get a closed-form solution:

$$w=(X^TX+\lambda I)^{-1}X^Ty \tag{3}$$

where the data X is one sample per row x_i, and each element of y is a regression target y_i. I is an identity matrix. To work in Foulier domain, x^T is replaced by $X^H=(X^*)^T$,

$$w=Fdiag(\frac{\hat{x}}{\hat{x}^*\odot\hat{x}+\lambda})F^H \tag{4}$$

In the Foulier domain

$$\hat{w}=\frac{\hat{x}\odot\hat{y}}{\hat{x}^*\odot\hat{x}+\lambda} \tag{5}$$

where the division is performed element-wise.

3.1.2 Non-linear Ridge Regression

To obtain good performance of the KCF classifier, mapping the inputs of a linear problem to a non-linear feature-space $\varphi(x)$ with the kernel trick is always used. The w can be expressed as $w=\sum_i\alpha_i\varphi(x_i)$.

$$f(x_i)=w^Tx=\sum_{i}^{n}\alpha_i\kappa(x_i,x_j) \tag{6}$$

Where $\kappa(x_i,x_j)=\langle\ \varphi(x_i),\varphi(y_i)\ \rangle$ is a kernel function. Let kernel matrix $K=\kappa(x_i,x_j)$. If the kernel matrix K is circulant, the variable under optimization are α, insteading of w. It can be computed as follows:

$$\alpha=(K+\lambda I)^{-1}y \tag{7}$$

Knowing which kernel we can use it to make K circulant, it is possible to diagonalize Eq.7 like the linear case, obtaining

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda} \tag{8}$$

Where $\hat{k}^{xx}$ is the first row of the kernel matrix $K = C(k^{xx})$. $C(\cdot)$ denotes circulant operation.

3.1.3 Tracking

In a new image, the target can be detected by the trained parameter α and the base image patch x . if the new sample is z , a confidence map y can be obtained by:

$$y = C(k^{xz})\alpha = \mathcal{F}^{-1}(\hat{k}^{xz}) \odot \hat{\alpha} \tag{9}$$

where $\odot$ is the element-wise product, the position with a maximum value in y can be predicted as new position of the tracking object. Since this is a filtering operation, it can be foumulated more efficiently in the Fourier domain.

3.2 The improved Simulation Annealing method

Simulated annealing (SA) [33] method is to search a good (not necessarily perfect) solution for an optimization problem and has been applied to many combinatorial optimization problems. In the search process, SA accepts not only better but also worse neighboring solutions with a certain probability. Such strategy can be regarded as a trial to explore new solution in special space. Specifically, it is a metaheuristic to approximate global optimization in a large search space.

3.2.1 The Standard SA method

The basic idea of SA is that the moves, made by an iterative improvement algorithm, are like the re-arrangement of the molecules in a liquid that occur as it is cooled and that the energy of those molecules corresponds to the cost function which is being optimized by the iterative algorithm. The probability of accepting a worse solution is larger at higher initial temperature. As the temperature decreases, the probability of accepting worse solutions gradually approaches zero. This feature means that the SA technique makes it possible to jump out of a local optimum to search for the global optimum. The SA algorithm is described as follows:

Algorithm1 the SA Algorithm

Initial: $T = T_0$, the iterative number L

Optimization:

1: For i from L

2: Generate a random solution

3: Calculate its cost using the cost function E

4: Generate a random neighboring solution with a certain rule

5: Calculate the new solution's cost
6: Compare them:
7: if $E_{new} < E_{old}$, move to the new solution
8: if $E_{new} > E_{old}$, accept the new solution with a probability
9: Repeat steps 3-5 above until an acceptable solution is found
Output: The optimal solution

The solution undergoes a local change at each iteration. The change causes a variation ΔE of the system energy. If this variation is negative, the new solution will replace the old one. Otherwise, the new solution will replace the old one with a probability based on the Metropolis-Hasting algorithm: $\rho(\Delta E) = e^{-\Delta E/T}$. Here, T is the temperature.

3.2.2 The Improved SA method

For the basic SA, the mechanism of accepting a worse solution may results in a better solution. In this case, the output of SA algorithm cannot be ensured as a global optimal value. Here, we store the better solutions according to their cost value, and then the best one is obtained from these solutions. The improved SA method is implemented as following sections.

(1) The object function: In this paper, we obtain the HOG feature of the image patch and take them for random variables. Their similarity is computed as:

$$\rho(X,Y) = \frac{Cov(X,Y)}{\sqrt{D(X)}\sqrt{D(Y)}} \tag{10}$$

where $D(\cdot)$ denotes the variance and $Cov(\cdot)$ denotes covariance. X and Y is the HOG feature of image patches respectively. The object function is defined as follows:

$$E = 2 + 2 * \rho(x, y) \tag{11}$$

The state of some physical systems, and the function E(s) to be minimized is analogous to the internal energy of the system in that state. Here, the goal is to bring the system, from an arbitrary initial state, to a state with the maximum possible energy.

(2) The state producing function: in the processing of approximating global optimization for a large search space, the key factor is how to generate the new solution. Generally, the new solutions are generated by bringing some small random perturbation to the known solution. This process can continue starting from the new solution until no further iteration. Suppose $S_i(x_i, y_i)$ is the solution in the process of iteration and $X_i = [x_i, y_i]$ is motion vector, the new solution is generated as follows:

$$\begin{cases} S_{k+1} = S_k + \Delta(E, X_{\max} - X_k) & \textit{if random value is} \text{ zero} \\ S_{k+1} = S_k + \Delta(E, X_k) & \textit{if random value} \text{ is non-zero} \end{cases} \tag{12}$$

where $X \in [0, X_{\max}]$ and S_{k+1} is the $(k+1)-th$ solution. $\Delta(E, X_{max})$ can generate a value in $[0, X]$, which is computed as follows:

$$\Delta(E, X_{max}) = X[1 - Q^{(E/E_{max})\kappa}] \tag{13}$$

Where Q is a random value and κ is the adjust parameter in $[1, 2]$ that can be called the perturbation factor. κ determines the random perturbation stepsize that impacts the quality of the new solution. The bigger perturbation stepsize can accelerate the rate of convergence. Unfortunately, this case may lose a better new solution. During the process of generating and searching new solution, the relation between perturbation stepsize and energy value each other is shown as the following **Fig. 2**.

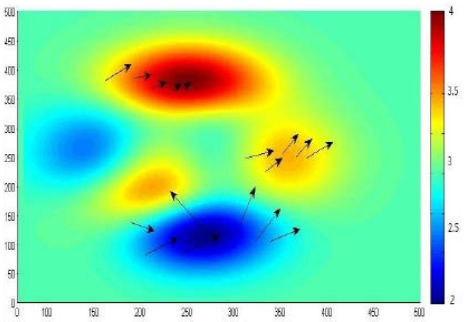


Fig. 2. The relation between perturbation stepsize and energy value

In **Fig. 2**, The x and y aix is the quantization of image width and height respectively. The colorbar denotes the quantization of the energy value ranging from 2 to 4. The arrow points the new solution and its length describes the perturbation stepsize. The smaller the energy value, the longer the arrow. Meanwhile, the perturbation stepsize also is larger.

(3) The Acceptance probabilities: according to the changes of energy value, the probability of making the transition from the current state s_k to a candidate new state s_{k+1} is specified by an acceptance probability function $P(s_k, s_{k+1}, T)$. Let $\Delta E = E(K+1) - E(K)$, the acceptance probability is got as follows:

$$P = \begin{cases} 1, & \Delta E > 0 \\ \exp(-\Delta E / T), & \Delta E \le 0 \end{cases} \tag{14}$$

Where T is the temperature value. When $\Delta E > 0$, the new solution is accepted. Otherwise, only if the random value is less than $\exp(-\Delta E / T)$ the new solution is accepted.

At each iteration, the SA heuristic considers some neighbouring state S_{k+1} of the current state S , and probabilistically decides between moving the system to state S_{k+1} or staying in state S . These probabilities ultimately lead the system to move to states of lower energy. Typically this step is repeated until a given temperature value.

(4) The optimal solution: the energy value shows the similarity between the new solution,

i.e. the HOG feature of candidate image patch, and the HOG feature of image template. In traditional SA, when the new solution is accepted the old solution would be discarded. The mechanism may get a worse solution as the optimal output. Given an initial solution x^*, we store the better solution $x^* + \Delta x_i (i = 1 \cdots n, n$ is the number of solution) in light of their corresponding energy value, where Δx is the displacement between solutions. The optimal solution is obtained as:

$$\Delta x^* = \arg\max_{\Delta x} E(\hat{q}, \hat{p}(x^* + \Delta x)) \tag{15}$$

Where $\hat{q}$ is the HOG feature of template image and $\hat{p}(x^* + \Delta x)$ is the HOG feature of the image patches associated with the better solutions.

3.3 The Annealed KCF tracker

In this paper, for enhancing the KCF tracker, we use improved SA method to search the global object candidates, which compensates the problem that the KCF tracker cannot handle large displacement motion. To this end, we provide the unified framework to track smooth and abrupt motion based on improved SA and KCF. The flow chart of the proposed method is shown as shown **Fig. 3**.

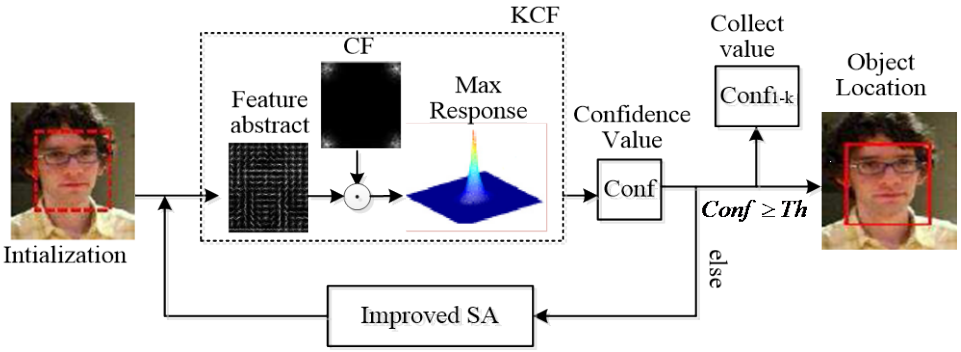


Fig. 3. A brief illustration for our method

The target is initialized by user at first frame. Then, KCF tracker gives a predict location according to maximal correlation response, i.e. maximal confidence value. If the value is larger the given threshold, the predict location will be accepted. Otherwise, the new base image patch is given by using the improved SA. Meanwhile, KCF tracker locates the target again. Finally, the given threshold is adjusted in light of collecting confidence values. Meanwhile, the object location is obtained.

3.3.1 The tracking problem

In the tracking, the object is represented by the target window as shown **Fig. 4**. At $t-1$ image, KCF tracker uses the cycle shifts version of base image sample to the dense samples. That is, the circulant matrix is purely generated by the cyclic shifts of base image patch and contains all candidate image samples from the Padding window. Obviously, when the t image occurs it is key factor how to determine the Padding window.

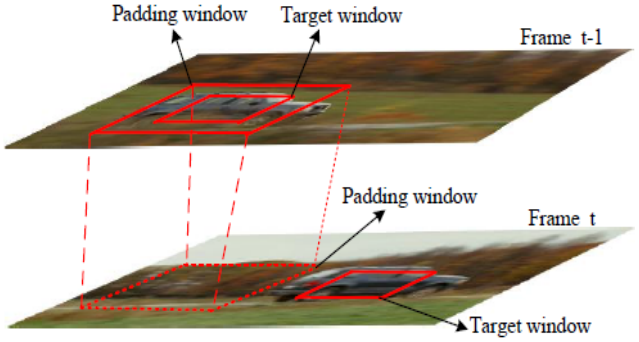


Fig. 4. Abrupt motion problem for KCF tracker

As shown **Fig. 4**, there is a larger motion displacement between image frame $t-1$ and t. On this occasion, the padding window does not cover the real targe at t frame image so as to impact the quality of the circulant matrix. If KCF tracker still locates the target in light of the maximal confidence score, the tracking may be failure.

3.3.2 The proposed tracking method

To overcome the above problem and keep the high robustness of KCF tracker, we propose a new tracking framework. The method considers the smooth and abrupt motion simultaneously, and gives a strategy of switching model according to the adaptive threshold. Meanwhile, to construct a good circulant matrix the improved SA method is applied to search the best base image patch globally. Finally, the classifier is updated in time by the effective training samples, which ensures a long-term object tracking.

(1)The adaptive threshold setup

In the KCF tracker, the object is located by using a confidence map. The confidence score can indicates the similarity between the candidate image patch and object appearance model. Here, for identifying the state of smooth or abrupt motion a confidence threshold is given. if the confidence score is larger the threshold we take the tracking results as a better candidate patch. However, in visual tracking the fixed threshold is difficult to evaluate the real motion state accurately in time. So, we propose an adaptive threshold setup method. Suppose $Cmap_t$ denotes the maximal confidence score at t frame, the confidence threshold is obtained as follows:

$$Cthr_k = \psi * median(Cmap_{k-j}) \quad j \in [1,4] \tag{16}$$

where ψ is a adjust parameter and $median(\cdot)$ is a getting the median value operation. The threshold can be modified with the changes of the different tracking results, which avoids the improper alternate model.

(2)The global motion model

The motion model considers how well each state is estimated in a given image sequence and how far it is from the previous estimated state. In traditional visual tracking, the commonest choices are a random-walk (RW) model, a nearly constant velocity (NCV) model or a Gaussian distribution (GS) model. RW model assumes that the target's velocity is a white-noise sequence and is temporally non-correlated. NCV model assumes velocity is temporally strongly correlated and nearly keep a constant change.GS model is constructed by zero-mean and variance, which inherits the target's importance for appearance model.

Obviously, the three models can not cover the abrupt motion and arbitrary motion. Different from existing trackers, KCF based method mainly needs to estimate the location of the base image sample, in which NCV motion model is adopted. In this section, we design a adaptive global motion model to estimate the location of base image sample based on the improved SA and NCV and then make them form the circulant matrix to cover abrupt motion.

$$\mathrm{x}_{t+1} = \Pi_t * r * (\mathrm{v}_t * t + \mathrm{x}_t) + (1 - \Pi_t)\hat{E}_{t+1}(w * \mathrm{x}_t) \tag{17}$$

Where $\mathrm{x}=[x, y]$ is vector and it denotes a pixel location, v is their velocity. r is a random numbers, uniformly distributed in [0,1]. w is a positive constant and $\hat{E}$ is the exploration factor. Π is the model parameter and is obtained as follows:

$$\Pi_{t+1} = \begin{cases} 1 & if\ C_{t+1}(x) > Cthr_t \\ 0 & if\ C_{t+1}(x) \le Cthr_t \end{cases} \tag{18}$$

At $t+1$ frame, the model selection depends on the parameter Π. When the highest confidence score is larger than the threshold the tracking performance is good. On the contrary, we think the tracking failure. At the moment, It is necessary to construct the circulant matrix again for search whole state space.

The exploration factor $\hat{E}_{t+1}$ adaptively adjusts the disturbance stepsize to find the new highest confidence score according to the energy value. The improved SA enables the exploration factor to escape local maxima. At each iteration, the proper location of base image patch is updated. Finally, the exploration is capable to cover abrupt motion and larger displacement between the consecutive image frames.

(3)The tracking implement

In visual tracking, the classifier is trained using the previous samples included in the circulant matrix. Then, when current image frame appear, the target’s location is seperated by the classifier. Tracker can learns the changes of the tracking object and implemented in frequency domain. The detail process can refer to the literature[8].The proposed tracking framework is summarized in Algorithm 2.

Algorithm 2 The SAKCF tracking framework

Input: Image sequence

Initialization:

Locate the target object in the first frame manually.

Get the initial state: $S_1 = X_1, T_1 = 1, e = 0.1^{30}$, Annealing temperature factor $\eta = 0.99, L = 1000000$, $f_{map} = 0.25$.

Tarcking:

1: For i from 2 to the last frame
2: Obtain the response value f_{i-1} of KCF
3: If $f_{i-1} < f_{map}$
4: for k=1:L
5: Generate the new solution S_i^k in light of Eq.12
6: Computer ΔE according to Eq.13
7: Obtain the new state S_i in light of Eq.14
8: $T = \eta * T$
9: If $T < e$
10: break
11: end If
12: end for
13: $S_i = S_{i-1}^k$
14: Else
15: $S_i = S_{i-1}$
16: End if
17: Collecting f_i value
18: Updating f_{map} according to Eq.16
19: Traking object using KCF

Output: Object state S_i in each frame.

4. Experiments

For showing the different tracking results with abrupt motion, we conduct two experiments to evaluate the efficacy of our proposed tracker. We divided the 12 image sequences into 3 groups based on the target's displacement between image frames. The first group contains the CAR2, FISH, DUDEK, DEER, and FACE1 sequences, whose motion displacement is less than 50 pixes. In the second group the displacement is more than 50 pixels and less than 100 pixels, including the FLURCAR3, ZXJ and FACE2 sequences. The third group contains FLEETFACE , and CT sequences, whose motion displacement is more than100 pixes. We compare the proposed method with other classical trackers on these sequences. We evaluate our tracker with the related 7 trackers to show the effectiveness of our method.

FHOZT BLURFACE

4.1 Experimental setup and methodolgy

We implemented the proposed tracker in MATLAB R2010a. The experiments were conducted on a PC with Intel Core i7 2.50GHz and 16GB RAM. We compared our tracker SAKCF with 7 state-of-the-art trackers, including Exploiting the Circulant Structure of Tracking-by-detection with Kernels (CSK) [34], Accurate Scale Estimation for Robust Visual Tracking (DSST) [10], Fast Compressive Tracking(FCT) [35], High-Speed Tracking with Kernelized Correlation Filters(KCF) [8],Fast Tracking via Spatio-Temporal Context Learning (STC) [12], the Structured output tracking with kernels(STRUCK) [6] and Least soft-threshold squares tracking(LSST) [36]. All parameter values of our tracker were kept consistent in all experiments. There are 12 challenging sequences in our experiments. The source of the FACE1 is the dataset AVSS2007. FACE2. ZT, FHC and ZXJ are our own. Other sequences are available on the website http://visualtracking.net. (listed in **Table 1**). Note that we extract the frames 306-310 in BLURFACE sequences and the frames 401-410 in FLEETFACE sequence, which can represent the problem of frame dropping.

Table 1. The image sequences

Video	Frame	Max displacement	X Max displacement	Y Max displacement
CAR2	913	3	3	2
FISH	476	15	15	13
DUDEK	1145	22	22	16
DEER	71	38	38	34
FACE1	380	39	22	39
BLURCAR3	357	65	65	49
ZXJ	118	70	70	18
FACE2	310	88	29	88
FLEETFACE	697	125	125	19
FHC	123	188	188	104
BLURFACE	488	202	202	71
ZT	115	256	256	149

In addition, the tracking results are evaluated by using distance precision(DP), centre location error(CLE) and overlap precision(OP) in [37]. DP is the relative number of frames in the sequence where the center location error is smaller than a certain threshold.

$$DP = \frac{N(\text{thresh})}{N} \tag{19}$$

where N is the total frames in a video, N(thresh) denotes the number of frames with CLE under threshold. We report DP values at a threshold of 50 pixels. OP is defined as the

percentage of frames where the bounding box overlap exceeds a threshold $t \in [0,1]$. Given the track region (e.g., bounding box) Tt and the groundtruth Gt, the OP is defined as

$$OP = \frac{\left\| G_t \cap T_t \right\|}{\left\| G_t \cup T_t \right\|} \tag{20}$$

where $\cap$ and $\cup$ represent the intersection and union of two regions, $\|\bullet\|$ denotes the number of pixels in the region and t is the frame number. We report the result at a threshold of 0.5.CLE is computed as the average Euclidean distance between the groundtruth and tracking result. We provide the qualitative analysis of our method with other trackers.

Parameter Settings: κ and ψ is a adjust parameter in Eq.13 and Eq.16. In our experiments, the former $\kappa = 2$ and the value keep fixed. The latter impacts the threshold setup. It's value ranges from 0 to 1 and is adjusted slightly according to the former thresholds. w in Eq.17 is a constant. It determines the iteration velocity for obtaining the optimal solution. e、L and η is the Stopping Criterion、the number of iterations and the annealing temperature factor respectively. During the course of the experiments, they keep fixed $e = 0.1^{30}$, $L = 1000000$, $\eta = 0.99$. The threshold f_m changes with the confidence values.

4.2 Qualitative analysis

4.2.1 The Slight Motion group

In the Slight Motion group, our tracker worked well and performed second only to one tracker. CAR2 image sequence has smallest motion at 3 pixels, all trackers have similar performance. For FISH sequence, nearly all trackers can catch up with the target successfully except for CSK. Our tracker and KCF performed second less than STRUCK. In DUDEK sequence, our tracker can adapt the problem resulting from illumination changes and partial occlusion at frames #795 and #797.LSST obtains the best tracking results. In DEER sequence the displacement reaches 38 pixels, although our tracker performed second less than STRUCK and CSK, our tracker performed close to them. They failed at frame #026and #032, when they were faced with abrupt large motion, STC cannot track the whole image sequence. In FACE1 sequence there has similar motion to DEER sequence, our tracker and KCF performed second only to DSST, while CSK failed at frame #205and #275, except LSST other trackers have a good tracking results. The representative tracking results are shown in Fig.5. In this case, our tracker can keep a good performance like trackers based on KCF.

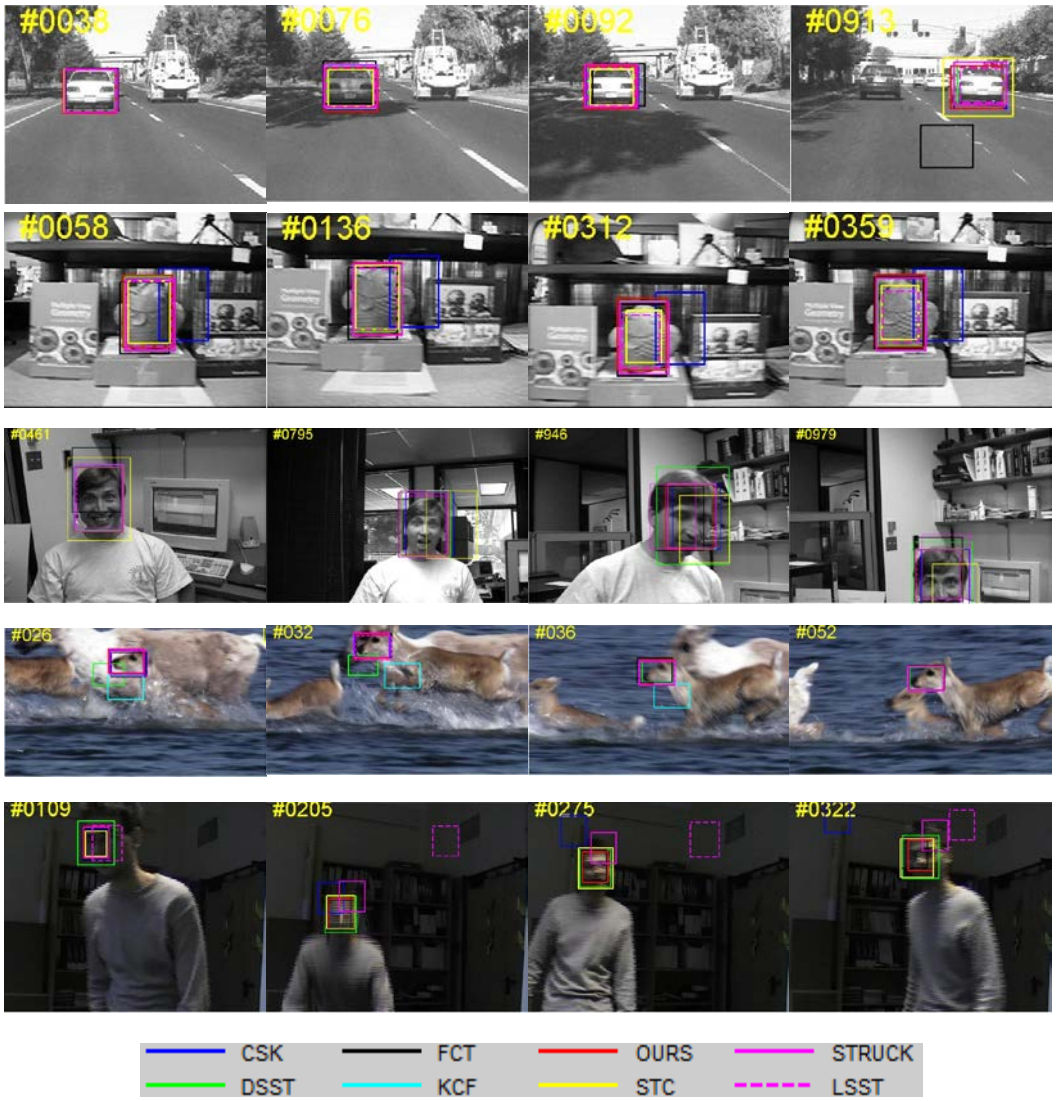


Fig. 5. A visualization of tracking results

4.2.2 The Middle Motion group

In order to verify the superior performance of our tracker, we continue to enhance the motion displacement. In BLURCAR3 sequence, there has the motion displacement of 65 pixels and shows the severe motion blur resulting from camera shaking. At the frame #119, when the displacement reaches the maximum DSST、CSK and our tracker can track the target. Other trackers are all failure. When illumination changes and motion blur occur at frame #268 and #337, DSST and our tracker still get a better performance. In ZXJ sequence, all trackers can track the target when the motion is smooth at frame #10 and #37. When the largest displacement arrives, our tracker gets the best performance. In FACE2 sequence there has the motion displacement of 88 pixels and slight illumination changing, our tracker

performed best and 7 other trackers all failed at frame #153and #160. The representative tracking results are shown in **Fig. 6**.

Fig. 6. A visualization of tracking results

4.2.3 The Large Motion group

We continue to enhance the motion displacement. In the FLEETFACE sequence, all trackers perform well at the first image frames.

When the displacement between images becomes larger, FCT does not adapt the situation. STC and our tracker follow the target closely at the frame #362. In addition, combined the tracking results at frames #411 and #635, our tracker obtains the best performance. In FHC sequence that has larger motion displacement at 188 pixels, our tracker performs better than 7 other trackers, while 7 other trackers all failed at the frame #031, #073and #081. In the BLURFACE sequence, we design the problem of the frame dropping. Meanwhile, there has a severe motion blur because the fast motion at the frames #150 and #272. STRUCK, LSST and FCT are all failure. When there is a largest displacement at frame #441, our tracker still can track the target successfully. In ZT sequence that has the largest motion at 256 pixels, our tracker performs much better than 7 other trackers, while 7 other trackers all fail at #035and #046.The representative tracking results are shown in **Fig. 7**. Obviously, our tracker has a strong superiority toward other trackers when the object undergoes a larger motion displacement between image frames.

Fig. 7. A visualization of tracking results

4.3 Quantitative analysis

Fig. 8 and **Fig. 9** reports the Distance Precision and Overlap Precision of 12 different sequences respectively. **Table 2** and **Table 3** list a per-sequence comparison of our tracker to CSK, DSST, FCT, KCF, STC, STRUCK, and LSST, while **Table 2** refers to average overlap rate and **Table 3** is concerned with average error center rate. In the tables, the best two results are shown in red and blue fonts. It is clearly seen in **Fig. 8**, **Fig. 9**, **Table 2** and **Table 3** that our tracker performs much better than 7 other trackers when there a larger motion displacement between consecutive images. In other image sequences, our tracker also shows a better performance. As a whole, the proposed method has a good merit for abrupt motion compared with 7 trackers.

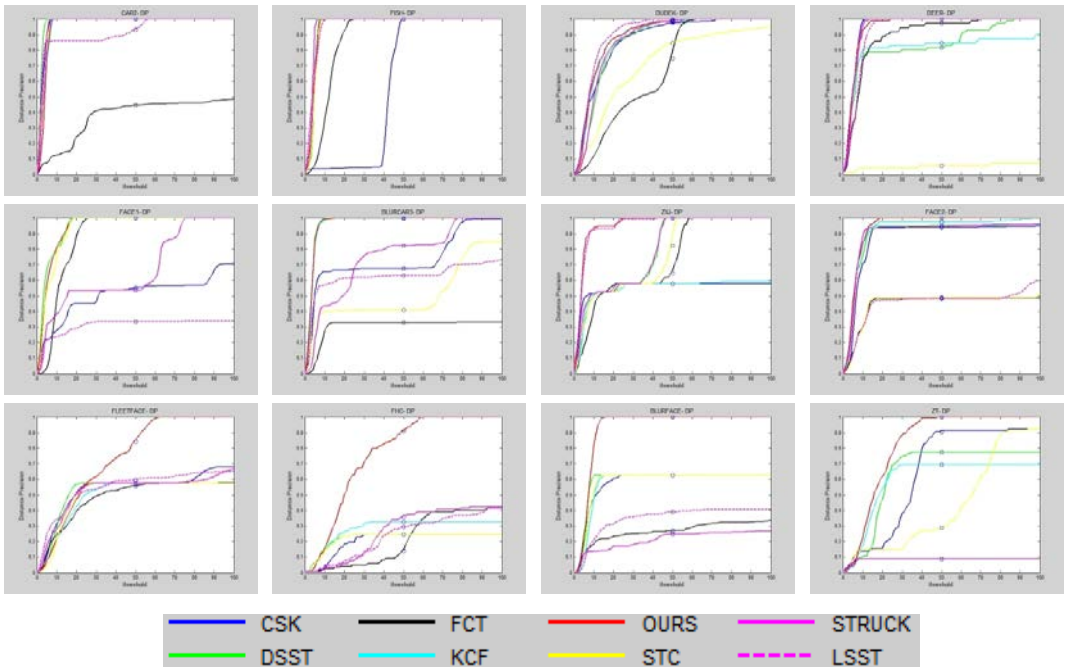


Fig. 8. The average precision of success plots

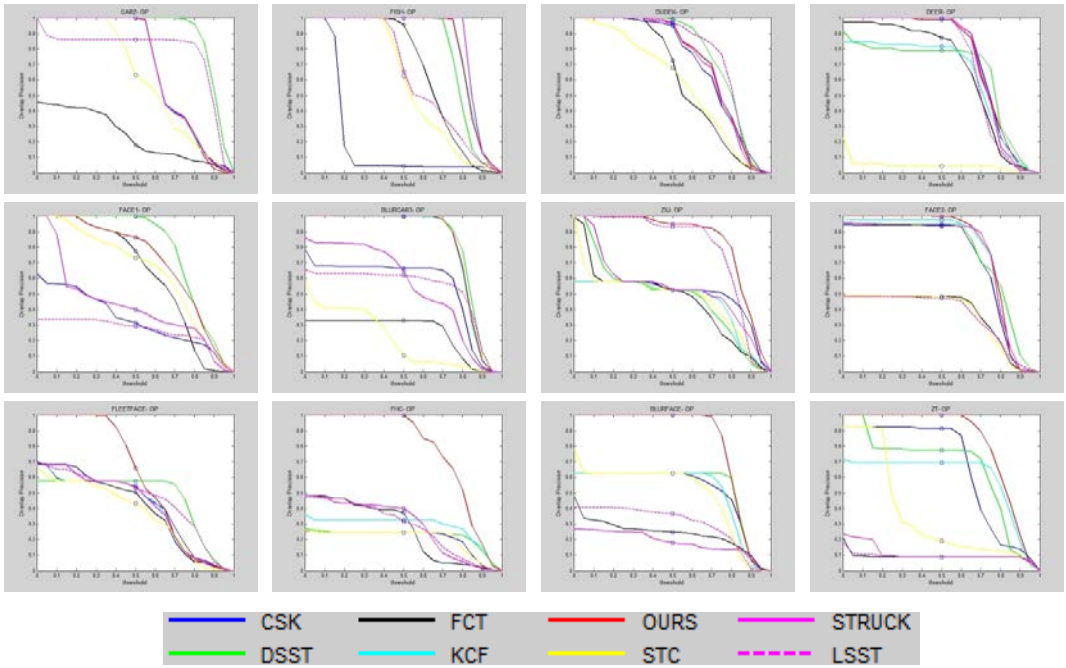


Fig. 9. The average precision of success plots

Table 2. Average overlap rates

Sequences	CSK	DSST	FCT	KCF	OURS	STC	STRUCK	LSST
CAR2	0.69	0.90	0.23	0.68	0.68	0.62	0.69	0.77
FISH	0.21	0.80	0.66	0.84	0.84	0.58	0.86	0.63
DUDEK	0.72	0.78	0.61	0.73	0.73	0.58	0.72	0.78
DEER	0.75	0.64	0.66	0.62	0.74	0.04	0.75	0.71
FACE1	0.33	0.79	0.63	0.72	0.72	0.65	0.42	0.26
BLURCAR3	0.55	0.84	0.25	0.82	0.82	0.20	0.55	0.53
ZXJ	0.49	0.48	0.45	0.45	0.84	0.46	0.53	0.79
FACE2	0.71	0.75	0.37	0.76	0.78	0.36	0.75	0.36
FLEETFACE	0.42	0.47	0.38	0.40	0.59	0.36	0.41	0.46
FHC	0.20	0.22	0.26	0.28	0.78	0.20	0.29	0.28
BLURFACE	0.51	0.53	0.23	0.51	0.81	0.48	0.18	0.30
ZT	0.66	0.65	0.09	0.59	0.82	0.35	0.11	0.09
Average	0.52	0.65	0.40	0.62	0.76	0.41	0.52	0.50

Table 3. Average center error rates

Sequences	CSK	DSST	FCT	KCF	OURS	STC	STRUCK	LSST
CAR2	3	2	85	4	4	3	3	8
FISH	41	4	12	4	4	5	4	4
DUDEK	13	14	32	11	11	29	13	9
DEER	5	17	11	21	5	510	5	7
FACE1	100	5	12	6	6	7	34	184
BLURCAR3	27	3	180	4	3	57	25	44
ZXJ	190	21	27	88	4	24	20	5
FACE2	18	14	138	8	6	107	13	116
FLEETFACE	90	131	81	108	26	113	81	81
FHC	578	616	371	365	24	576	340	392
BLURFACE	1574	75	116	85	7	90	199	162
ZT	54	54	642	128	18	100	674	685
Average	224	80	142	69	9.9	135	118	141

5. Conclusions

In this paper, for coping with the problem of the KCF-based tracker when object undergoes abrupt motion, we propose a new tracker based on swarm intelligence that is improved simulation annealed method. The method can track smooth and abrupt motion simultaneously. Meanwhile, the global motion model is constructed to estimate target's state according to the

confidence score adaptively. Experimental results show that the proposed algorithm improves the accuracy of tracking when object undergoes abrupt motion. In the future, we would like to further research the robustness of our method as well as its capability of coping with abrupt motion.

References

[1] H. Zhang, S. Hu, X. Zhang, and L. Luo, "Visual Tracking via Constrained Incremental Nonnegative Matrix Factorization," *IEEE signal processing letters*, vol. 22, no. 9, pp.1350-1353, September, 2015. Article (CrossRef Link)

[2] T. Zhang, B. Ghanem, S. Liu, C. Xu and N. Ahuja, "Robust Visual Tracking via Exclusive Context Modeling," *IEEE Transactions on Cybernetics*, vol. 46, no.1, pp. 51-63, January, 2016. Article (CrossRef Link)

[3] D. Wang and H. Lu, "Fast and Robust Object Tracking via Probability Continuous Outlier Model," *IEEE Transaction on Image Processing*, vol. 24, no. 12, pp. 5166-5176, December, 2015. Article (CrossRef Link)

[4] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang and B. Ghanem, "Robust Visual Tracking Via Consistent Low-Rank Sparse Learning," *International Journal of Computer Vision*, vol. 111, no. 2, pp. 171-190, January, 2015. Article (CrossRef Link)

[5] H. Zhang, Y. Wang, L. Luo, X. Lu, M. Zhang, "SIFT flow for abrupt motion tracking via adaptive samples selection with sparse representation," *Neurocomputing*, vol. 249, no. 2, pp. 253-265, August, 2017. Article (CrossRef Link)

[6] S. Hare, et al., "Struck: Structured Output Tracking with Kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2096-2109, October, 2016. Article (CrossRef Link)

[7] D. S. Bolme, et al., "Visual object tracking using adaptive correlation filters," in *Proc. of 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2544-2550, August, 2010. Article (CrossRef Link)

[8] J. F. Henriques, et al., "High-Speed Tracking with Kernelized Correlation Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583-596, March, 2015. Article (CrossRef Link)

[9] M. Tang, and J. Feng, "Multi-Kernel Correlation Filter for Visual Tracking," in *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3038-3046, December, 2015. Article (CrossRef Link)

[10] M. Danelljan, G. Hger, F.S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. of the British Machine Vison Conference BMVC*, September, 2014. Article (CrossRef Link)

[11] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. of Computer Vison-ECCV 2014 Workshops*, vol. 8926, pp. 254-265, September, 2014.

Article (CrossRef Link)

[12] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatiotemporal context learning," *in ECCV*, vol. 8693, pp. 127-141, 2014. Article (CrossRef Link)

[13] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proc. of the IEEE Conference on Computer Vison and Pattern Recognition*, pp. 4902-4912, October, 2015. Article (CrossRef Link)

[14] Y. Li, J. Zhu and Steven C.H., Hoi, "Reliable patch trackers Robust visual tracking by exploiting reliable patches," in *Proc of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 353-361, October, 2015. Article (CrossRef Link)

[15] S. Liu, T. Zhang, X. Cao and C. Xu, "Structural Correlation Filter for Robust Visual Tracking," in *Proc of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4312-4320, December, 2016. Article (CrossRef Link)

[16] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking." *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 25, no. 5, pp. 564-577, May, 2003. Article (CrossRef Link)

[17] D. Ross, J. Lim , R. Lin and M-H yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125-141, May, 2008. Article (CrossRef Link)

[18] D. Wang and H. Lu, "Fast and Robust Object Tracking via Probability Continuous Outlier Model," *IEEE Transaction on Image Processing*, vol. 24, no. 12, pp. 5166-5176, December, 2015. Article (CrossRef Link)

[19] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1269-1276, August, 2010. Article (CrossRef Link)

[20] X. Mei and H. Ling, "Robust visual tracking using L1 minimization," in *Proc of Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp.1436-1443, November, 2009. Article (CrossRef Link)

[21] C. Bao, Y. Wu., H. Ling and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *Proc of Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp.1830-1837, July, 2012. Article (CrossRef Link)

[22] T. Zhou, B. Harish, F. Liu and J. Yang, "Graph Regularized and Locality-constrained Coding for Robust Visual Tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. pp, no. 99, June, 2016. Article (CrossRef Link)

[23] T. Zhang, B. Ghanem and S. Liu, "Robust Visual Tracking via Structured Multi- Task Sparse Learning," *International Journal of Computer Vision*, vol. 101, no. 2, pp. 367-388, January, 2013. Article (CrossRef Link)

[24] T. Zhang, A. Bibi and B. Ghanem, "In Defense of Sparse Tracking: Circulant Sparse Tracker," *Computer Vision and Pattern Recognition (CVPR)*, pp. 3880-3888, December, 2016. Article (CrossRef Link)

[25] T. Zhang, S. Liu, C. Xu, S. Yan, B. Ghanem, N. Ahuja, and M.-H. Yang, "Structural Sparse Tracking," *Computer Vision and Pattern Recognition (CVPR)*, pp. 150-158, October, 2015. Article (CrossRef Link)

[26] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 26, no . 8, pp.1064-1072, August, 2004. Article (CrossRef Link)

[27] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. of the British Machine Vision Conference*, pp. 47-56, January, 2006. Article (CrossRef Link)

[28] B. Babenko, M. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 983-990, August, 2009. Article (CrossRef Link)

[29] N. Wang, M. Yang, and D.Y. Yeung, "Learning a deep compact image representation for visual tracking," *in Nips*, pp. 809-817, January, 2013. Article (CrossRef Link)

[30] J. Gao, T. Zhang, X. Yang and C. Xu, "Deep Relative Tracking. IEEE Transactions on Image Processing," vol. 26, no. 4, pp. 1845-1858, January, 2017. Article (CrossRef Link)

[31] H. Li, Y. Li, and F. Porikli, "Deeptrack: Learning discriminativefeature representations by convolutional neural networks for visual tracking," in *Proc. of the British Machine Vision Conference. BMVCPress*, January, 2014. Article (CrossRef Link)

[32] L.Wang, T. Liu, G.Wang, K. L. Chan, and Q. Yang, "Video tracking using learned hierarchical features. Image Processing," *IEEE Transactions on*, vol. 24, no. 4, pp. 1424-1435, April, 2015. Article (CrossRef Link)

[33] S. Bandyopadhyay, et al., "A Simulated Annealing-Based Multiobjective Optimization Alg orithm: AMOSA," *IEEE Transactions on Evolutionary Computation*, vol. 12, no.3, pp. 26 9- 283, June, 2008. Article (CrossRef Link)

[34] J. F. Henrigues, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," *in Computer Vison-ECCV 2012*, vol. 7575, pp. 702-715, 2012. Article (CrossRef Link)

[35] K. Zhang, L. Zhang, and M.-H. Yang, "Fast compressive tracking," *The IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 36, no. 10, pp. 2002-2015, 2014. Article (CrossRef Link)

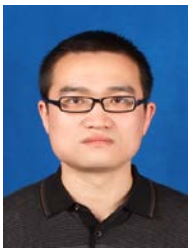
[36] D. Wang, H. Lu and M. Yang, "Robust Visual Tracking via Least Soft-threshold Squares," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1709-1721, September, 2016. Article (CrossRef Link)

[37] Y. Wu, J. Lim and M.-H. Yang, "Online Object Tracking: A Benchmark," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2411-2418, October, 2013. Article (CrossRef Link)

Huanlong Zhang received the M.S. degree from Henan University, Kai Feng, China, in 2007, and Ph.D. degree at Shanghai Jiao Tong University, Shang Hai, China, in 2015. Now, he is a teacher at Zhengzhou University of Light Industry. His research interests mainly include pattern recognition, machine learning, image processing and computer vision.

Jianwei Zhang is a professor. He received the M.S. degree from Hua-zhong University of Science & Technology, Wuhan, China, in 2001, and Ph.D. degrees from PLA Information Engineering University, Zhengzhou, China, in 2010. He is currently working at Zhengzhou University of Light Industry, Zhengzhou, China. His research interests are in broadband information network, distributed system, Information security, computer vision etc.

QingE Wu, Doctor, double Postdoctor, Professor. She received the M.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, and the Ph.D. degree from the Xi'an Jiaotong University, Xi'an, China. She is currently working at Zhengzhou University of Light Industry, Zhengzhou, China. Her main research interests include intelligent control, fuzzy control, pattern recognition, image processing, data processing, etc.



Xiaoliang Qian received his M.S. degree from Northwestern Polytechnical University, Xi an, China, in 2007, and Ph.D degree from Northwestern Polytechnical University, Xi an, China, in 2013.He is currently an associate professor at Zhengzhou University of Light Industry, Zhengzhou, China. His research interests include image processing, machine learning, scene classification and visual saliency detection.

Tong Zhou is a third year bachelor major in automation at Zhengzhou University of Light Industry, Zhengzhou, China. His research interests focus on machine learning and visual tracking.

Hengcheng Fu is a third year bachelor major in automation at Zhengzhou University of Light Industry, Zhengzhou, China. His research interests focus on machine learning and visual tracking.