Online adaptive complementation tracker

Guokai Shi[1], Tingfa Xu[1,2*], Jiqiang Luo[1], Jie Guo[1] and Zhitao Rao[1]

Abstract

Correlation filter-based trackers have recently shown excellent performance in terms of motion blur and illumination changes, but they are notoriously sensitive to deformation. It has been demonstrated that the combination of the correlation filter-based tracker and the color histogram-based tracker can alleviate the deformation and keep advantages of the correlation filter-based tracker. However, the most existing complementary tracking algorithms, which use fixed complementary weights, limit the performance of every sub tracker. This paper introduces an adaptive complementary tracker by online learning dynamic complementary weights. The strategy enables inappropriate sub tracker to be down-weighted while increasing the impact of suitable one. We jointly learn the sub trackers and their reliability weights by regression analysis of the corresponding historical tracking results. The robustness of the model also can be improved by training each sub tracker with the result of historical tracking. Finally, both qualitative and quantitative evaluations demonstrate that our tracker achieves the state-of-the-art results in a wide range of tracking scenarios.

Keywords: Correlation filter, Online learning, Visual tracking, Complementary tracker

1 Background

Visual tracking plays an important role in computer vision and has received fast-growth attention in recent years due to its wide practical application. In generic tracking, the task is to track an unknown target (only a bounding box defining the object of interest in a single frame is given) in an unknown video stream. This problem is very challenging due to the limited set of training samples and the numerous appearance changes, e.g., rotations, scale changes, occlusions, and deformations.

For solving these problems, many effective tracking models have been proposed [1] in recent years. These models can be categorized into methods based on subspace algorithms [2, 3], sparse representation [4–8], online classifiers [9–12], and so on. Recently, correlation filter-based tracking algorithms [13–17] have drawn increasing attention because of its dense sampling property and its fast computation in the Fourier domain.

Bolme et al. [13] propose the MOSSE tracker which finds a filter by minimizing the sum of squared error between the actual convolution outputs and the desired convolution outputs. The MOSSE tracker can handle several hundreds of frames per second because of the fast element-wise multiplication and division in the Fourier domain. Henriques et al. [18] extend correlation filters to a kernel space, leading to the Circulant Structure of Tracking-by-Detection with Kernels (CSK) tracker which achieves competitive performance and efficiency. To further improve the performance, the KCF method [19] integrates multiple features into the CSK tracking algorithm. The Discriminative Scale Space Tracker (DSST) tracker [20] utilizes one-dimensional correlation filters for online estimation of target scale to overcome a wide range of changes to target scale. Furthermore, Montero et al. [21] propose a scalable kernel correlation filter which introduces an adjustable Gaussian window function and a key point-based model for scale estimation. This strategy can deal with the fixed size limitation in the Kernelized Correlation Filter.

Correlation filters are inherently confined to the problem of learning a rigid template. By learning their model from circular shifts of positive examples, correlation filters fail to learn a component that is invariant to permutations. This makes them inherently sensitive to shape deformation. This is a concern when the target experiences shaped formation in the course of a sequence. Yao et al. [22] decompose the target object into several parts to exploit the spatial information of the object appearances. Their strategies improve the tracker's performance in

* Correspondence: xutingfa_bit@126.com
[1]School of Optoelectronics, Image Engineering &Video Technology Lab, Beijing Institute of Technology, Beijing 100081, China
[2]Key Laboratory of Photoelectronic Imaging Technology and System, Ministry of Education of China, Beijing 100081, China

Shi *et al. EURASIP Journal on Wireless Communications and Networking* (2018) 2018:191

Page 2 of 9

situations like occlusions and deformations. In addition, color histograms were used in many early approaches to object tracking [23, 24]; they have been demonstrated to be competitive in the Distractor-Aware Online Tracking (DAT) [25]. More recently, Luca et al. [26] propose a simple linear combination of template and color histogram scores by a fixed weight to alleviate shape deformation. However, the fixed weight limits the performance of the combination model.

To solve the discussed problem above, this paper presents an online adaptive complementary tracker. The contributions of this work are twofold. First, we propose a novel formulation for jointly learning two sub trackers and their reliability weights. These sub trackers have better generalization ability, because they use more samples during training. Second, we online analyze the reliability of every sub tracking model in the current scene to real-time adjust their importance in the complementary model. By the adjustment of the adaptive weight, our combination model shows better performance.

2 Methods

2.1 Problem formulation

It is necessary to understand the real-time state of the scene for adaptively assigning the appropriate tracker to the current scene. We try to analyze the matching between every sub model and the scene by analyzing the regression deviation between the sub tracker and some history tracking results. This strategy is based on an assumption where all history tracking results are accurate and reliable. However, in practice, this assumption is uncertain. In order to avoid the unreliable assumption, we propose to jointly learn every sub tracking model and their corresponding reliability weights in a tracking-by-detection framework. Figure 1 shows the diagram of the proposed method.

Specifically, our method jointly learns both sub trackers h_1 and h_2 and their weights α and β by minimizing a single loss function. To the best of our knowledge, we are the first to introduce the joint optimization framework by the joint loss,
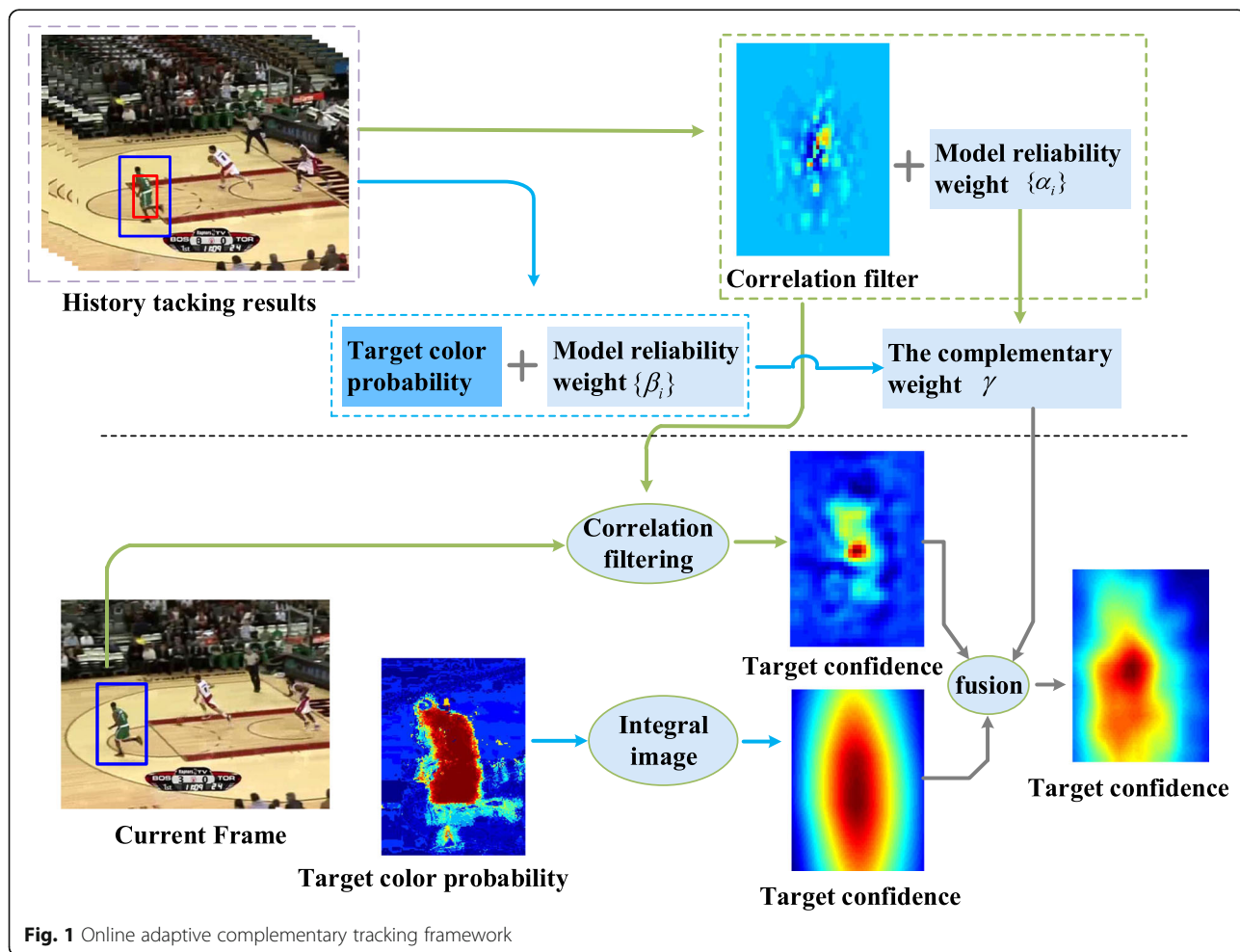

Fig. 1 Online adaptive complementary tracking framework

$$\mathscr{L}(\alpha,\beta,h_1,h_2) = \sum_{i=1}^{t} \alpha_i l_1(h_1,\phi_i) + \varepsilon\beta_i l_2(h_2,\psi_i)$$
$$+\lambda_1 \mathscr{R}_1(h_1) + \lambda_2 \mathscr{R}_2(h_2) + \eta \sum_i^{t} \mathscr{R}_3(\alpha_i,\beta_i) \qquad (1)$$

Here, l_1 and l_2 denote two regression problems; they will be used to learn two sub models which make up our complementary appearance model. h_1 is a correlation filter-based tracker, and its complementary tracker h_2 is based on color histogram scores. The weight ε assures l_1 and l_2 consistent value interval. $\mathscr{R}_1$, $\mathscr{R}_2$, and $\mathscr{R}_3$ are three independent regularization terms.

2.2 Constructing and solving the sub tracker h_1

We use the Background-aware Correlation Filters (BACF) model [27] as baseline model of h_1 because it can effectively handle the boundary effect which is a fundamental drawback to traditional correlation filter-based trackers. However, similar to many other correlation filter-based trackers, the BACF model also depends strongly on the spatial layout of the tracked object, so is sensitive to deformation. In addition, the BACF model ignores the problem of corrupted samples, and when those corrupted samples are included in the training set, the model drift may be caused.

Being motivated by [28], we develop the BACF model into Eq. 2 to learn our sub tracker h_1:

$$\mathscr{L}_{h_1}(h_1) = \frac{1}{2}\sum_{i=1}^{t}\alpha_i \sum_{j=1}^{J} \left\| \left(y_i(j) - \sum_{d=1}^{D} h_1^{d^T} P \phi_i^d [\Delta\tau_j] \right) \right\|_2^2$$
$$+ \frac{\lambda_1}{2}\sum_{l=1}^{D} \left\| h_1^l \right\|_F^2 + C_1 \qquad (2)$$

where P denotes transform matrix which is a $E \times J$ binary matrix for cropping mid E elements of sample ϕ_i^d, $E \ll J$. The cropped patch corresponding to the peak of the correlation output displays the target (positive example), and those corresponding to the zero values of the correlation output display the background content (negative examples). The transform matrix P can be pre-computed, and $P\phi_i^d$ can be efficiently performed via a lookup table. ϕ is expressed by Histogram of Oriented Gradients (HOG) features. Compared with [28], our method adopts the sampling strategy of the BACF, which introduces the transform matrix P into cyclic sampling to improve the distinction between samples. More importantly, the proposed optimization model of the parameter h_1 not only purifies the training sample set, but also verifies the reliability of the model for current sequence.

For improving computing speed, Eq. 2 can be expressed in the frequency domain as:

$$\hat{\mathscr{L}}(h_1,\hat{g}) = \frac{1}{2}\sum_{i=1}^{t}\alpha_i \|\hat{y} - \hat{\phi}_i\hat{g}\|_F^2 + \frac{\lambda_1}{2}\|h_1\|_F^2 , \qquad (3)$$
$$s.t. \ \hat{g} = \sqrt{J}\left(FP^T \otimes I_D\right)h_1$$

where $\hat{g}$ is an auxiliary variable and the matrix $\hat{\phi}_i$ is defined as $\hat{\phi}_i = [\ diag(\phi_i^1)^T, ...,\ diag(\phi_i^d)^T, ...\ diag(\phi_i^D)^T\ ]$. The matrix I_D is an $D \times D$ identity matrix, and the operator $\otimes$ is the Kronecker product. A ˆ denotes the discrete Fourier transform (DFT). The orthonormal matrix F can map any J dimensional vectorized signal to the Fourier domain. T indicates the transpose operator, which computes the conjugate transpose on a complex vector or matrix.

We can rewrite Eq. 3 into an augmented Lagrangian form, then it can be iteratively solved by the ADMM [29] technique.

A. *Solving sub problem* h_1^*

$$h_1^* = \ \arg\ \min_{h_1}\{\frac{1}{2}\sum_{i=1}^{t}\alpha_i\|\hat{y} - \hat{\phi}\hat{g}\|_2^2 + \frac{\lambda_1}{2}\|h_1\|_F^2$$
$$+\hat{\zeta}^T\left(\hat{g} - \sqrt{J}\left(FP^T\otimes I_D\right)h_1\right) + \frac{\mu}{2}\left\|\hat{g} - \sqrt{J}\left(FP^T\otimes I_D\right)h_1\right\|_2^2\},$$
$$= \left(\mu + \frac{\lambda_1}{\sqrt{J}}\right)^{-1}(\mu g + \zeta)$$
$$(4)$$

B. *Solving sub problem* $\hat{g}^*$

$$g^* = \ \arg\ \min_{\hat{g}}\{\frac{1}{2}\sum_{i=1}^{t}\alpha_i\|\hat{y} - \hat{\phi}_i\hat{g}\|_2^2$$
$$+\hat{\zeta}^T\left(\hat{g} - \sqrt{J}\left(FP^T\otimes I_D\right)h_1\right) + \frac{\mu}{2}\left\|\hat{g} - \sqrt{J}\left(FP^T\otimes I_D\right)h_1\right\|_2^2\}$$
$$(5)$$

Because $\hat{\phi}_i$ is sparse banded, each element of $\hat{y}$ is dependent only on D values of $\hat{\phi}_i(j) = [\hat{\phi}_i^1(j), ..., \hat{\phi}_i^D(j)]^T$ and $\hat{g}(j) = [conj(\hat{g}^1(j)), ..., conj(\hat{g}^D(j))]^T$. Here, the operator conj (.) indicates the complex conjugate transform. Similar to [27], solving Eq. 5 for $\hat{g}^*$ can be identically expressed as J smaller, independent objectives, solving for $\hat{g}(j)^*$:

$$\hat{g}(j)^* = \ \arg\ \min_{\hat{g}}[\frac{1}{2}\sum_{i=1}^{t}\alpha_i\|\hat{y}(j) - \hat{\phi}_i(j)\hat{g}(j)\|_2^2$$
$$+\hat{\zeta}(j)^T\left(\hat{g}(j) - \hat{h}_1(j)\right) + \frac{\mu}{2}\left\|\hat{g} - \hat{h}_1(j)\right\|_2^2] \qquad (6)$$

where, $\hat{h}_1(j) = [\hat{h}_1^1(j), ..., \hat{h}_1^D(j)]$ and $\hat{h}_1^D = \sqrt{D}(FP^T h_1^D)$.

Then, Eq. 6 is minimized by solving the normal equations:

$$
\hat{g}(j)^* = \left(J\mu I_D + \sum_{i=1}^{t} \alpha_i \hat{\phi}_i(j)\hat{\phi}_i(j)^T \right)^{-1} \\
\left(J\mu \hat{h}_1(j) - J\hat{\zeta}(j) + \sum_{i=1}^{t} \alpha_i \hat{y}(j)\hat{\phi}_i(j)^T \right).
$$
(7)

To design an efficient method, we solve Eq. 7 by the Gauss-Seidel iteration. Set $A = J\mu I_D + \sum_{i=1}^{t} \alpha_i \hat{\phi}_i(j)\hat{\phi}_i(j)^T$ which is symmetric and positive definite. The Gauss-Seidel method decomposes the matrix A into a lower triangular component L and a strictly upper triangular component U, such that $A = U + L$. The filter $\hat{g}(j)^*$ is then iteratively calculated by equation:

$$
L\hat{g}(j)^{(m+1)} = J\mu\hat{h}_1(j) - J\hat{\zeta}(j) \\
+ \sum_{i=1}^{t} \alpha_i \hat{y}(j)\hat{\phi}_i(j)^T - U\hat{g}(j)^{(m)}.
$$
(8)

Here, m is mth iteration in Gauss-Seidel iteration operation.

C. *Lagrangian vector update*

Then, we update the Lagrangian vector as:

$$
\hat{\zeta}(j)^{(n+1)} = \hat{\zeta}(j)^{(n)} + \mu\left(\hat{g}(j)^{(n+1)} - \hat{h}^{(n+1)} \right),
$$
(9)

where n indicates the nth iteration in the ADMM operation.

2.3 Constructing and solving the sub tracker h_2

Similar to [26], we also use the color histogram-based model as the complementary tracker of h_1. The difference is that we extend the training set of the model in the time dimension. The expanded model, which makes use of more diverse training samples and has better robustness, can be learn by optimizing Eq. 10:

$$
\mathscr{L}_{h_2}^{(m)}(h_2) = \sum_{i=1}^{t} \beta_i^{(m)} l_2(h_2, \psi_i) + \lambda_2 \mathscr{R}_2(h_2) \\
+ C_2.
$$
(10)

The regression problem can be described as:

$$
l_2(h_2) = \frac{1}{|\mathcal{O}_i|} \sum_{u \in \mathcal{O}_i} \left(h_2^T \psi_i[u] - 1 \right)^2 + \frac{1}{|\mathcal{B}_i|} \sum_{u \in \mathcal{B}_i} \left(h_2^T \psi_i[u] \right)^2,
$$
(11)

where $\mathcal{O}$ and $\mathcal{B}$ denote the target area and background area.

In the ridge regression problem, the solution of the per feature dimension is:

$$
h_2^{d*} = \arg \min_{h_2^d} \sum_{i=1}^{t} \beta_i^{(m)} \left(\rho^d(\mathcal{O}_i)\left(h_2^d - 1 \right)^2 + \rho^d(\mathcal{B}_i)\left(h_2^d \right)^2 \right) + \lambda_2 \| h_2^d \|_F^2 \\
= \frac{\sum_{i=1}^{t} \beta_i^{(m)} \rho^d(\mathcal{O}_i)}{\sum_{i=1}^{t} \beta_i^{(m)} \left(\rho^d(\mathcal{O}_i) + \rho^d(\mathcal{B}_i) \right) + \lambda_2},
$$
(12)

where $\rho^d(\mathcal{A}) = N^d(\mathcal{A})/|\mathcal{A}|$ is the proportion of pixels in a region for which feature d is non-zero. $N^d(\mathcal{A}) = |\{u \in \mathcal{A} : k[u] = d\}|$ is the number of pixels in the region $\mathcal{A}$ of the feature ψ.

2.4 Solving the reliability weights α and β

When h_1 and h_2 are obtained, Eq. 1 can be expressed as an optimization problem for α and β. The optimization problem about α and β can be rewritten into the following form:

$$
\mathscr{L}_w^{(m)}(w) = wL + \eta \sum_{i=1}^{t} \frac{\|w_i\|_2^2}{2p_i}, \\
s.t. \quad \|w\| = 1
$$
(13)

where $w = [w_1, w_2, ... w_i ... w_t]$, $L = [L_1, ..., L_i, ..., L_t]^T$, $w_i = (\alpha_i, \varepsilon^* \beta_i)$, $\alpha_i > 0$, $\beta_i > 0$, and $L_i = (l_1, l_2)^T$. p_i is the prior weights whose definition is similar with one in [28], $p_i > 0$ and $\sum_{i=1}^{t} p_i = 1$. In the control of the parameter p_i, recent fame is given larger attention to account for fast appearance changes.

We can use the convex quadratic programming methods in Matlab's Optimization Toolbox to solve the above optimization problem.

2.5 Solving the complementary weights

Our score function is a linear combination of template and histogram scores:

$$
f_{merge} = \gamma * f_{h_1} + (1-\gamma) * f_{h_2}.
$$
(14)

f_{h_1} denotes a template scores, which is the confidence map acquired by correlation operation between correlation filter g and spatial layout features ϕ.

$$
f_{h_1} = \mathscr{F}^{-1}(\hat{\phi}\hat{g}),
$$
(15)

f_{h_2} denotes a histogram scores, which is computed from an M-channel feature image.

$$
f_{h_2} = h_2^T \left(\sum_{x \in \mathscr{H}} \psi[u]/|\mathscr{H}| \right).
$$
(16)

The complementary weight γ is an adaptive variable instead of pre-assigned constant as one in [26]. Reliability weights obtained by Eq. 13 can reflect the reliability of the corresponding sub model. To alleviate the problem of rapid appearance changes, we choose the

reliability weights of the nearest n ($n = 10$) frames to dynamically adjust our weight γ. In addition, considering that correlation filter-based tracker is more favorable for accurate location than the color histogram-based tracker, we choose the template scores as the main tracking scores while the color histogram scores as accessorial ones. Based on the discussion above, we construct the following complementary weight model:

$$
\gamma = \begin{cases}
\dfrac{\sigma_2}{\sigma_1 + \sigma_2}, & \left(\sigma_2 < \sigma_1 \ \text{and} \ \dfrac{\sigma_2}{\sigma_1 + \sigma_2} > \mathcal{T}_{\text{low}} \right) \\
& \text{or} \left(\sigma_2 > \sigma_1 \ \text{and} \ \dfrac{\sigma_2}{\sigma_1 + \sigma_2} > \mathcal{T}_{\text{up}} \right) \\
\mathcal{T}_{\text{low}} & , \sigma_2 < \sigma_1 \ \text{and} \ \dfrac{\sigma_2}{\sigma_1 + \sigma_2} < \mathcal{T}_{\text{low}} \\
\mathcal{T}_{\text{up}} & , \sigma_2 > \sigma_1 \ \text{and} \ \dfrac{\sigma_2}{\sigma_1 + \sigma_2} < \mathcal{T}_{\text{up}}.
\end{cases}
\tag{17}
$$

Here, σ_1 and σ_2 are the variances of the data set $\{\alpha_i - p_i | \ i = t - n, ..., t\}$ and $\{\beta_i - p_i | \ i = t - n, ..., t\}$, respectively. $\mathcal{T}_{\text{low}}$ and $\mathcal{T}_{\text{up}}$ are the lower threshold and upper threshold respectively.

3 Experimental results and discussion

The proposed method in this paper is implemented in MATLAB 2016a. We perform the experiments on a PC with Intel i7-4790 CPU (3.6GHz) and 16-GB RAM memory, and the tracker runs at 15 fps. We extensively evaluate the performance of the proposed tracker with the total 100 challenging sequences from [30] and compare our tracker with the top 10 state-of-the-art trackers, including DSST [20], Structured Output Tracking with Kernels (Struck) [31], BACF [27], Tracking-by-Detection (TLD) [32], CSK [18], Spatially Regularized Discriminative Correlation Filters (SRDCF) [33], Tracking via Multiple Experts Using Entropy Minimization (MEEM) [34], Adaptive Color Tracker (ACorT) [35], Staple, and DAT [25]. The DSST, the CSK, the SRDCF, and the ACorT belong to the correlation filter-based tracker, which more attention to the structural attribute of the target. The Struck, the TLD, and the MEEM are tracking algorithms based on classifier learning, which divides the scene into target and background. The DAT belongs to the unstructured tracker, which is suitable for the tracking target with severe deformation. The Staple is a combined tracker, which has the strongest correlation with our algorithm.

3.1 Parameter setting

In the proposed tracking model, the HOG feature and the color histogram feature are two important factors affecting the computation. During the experiment, we set HOG features to 31 channels and use cell size of 4×4 pixels in

order to be unified with other algorithms, e.g., BACF, Staple, DSST, and CSK. The size of the search region is set to be 4^2 times the size of the target area in all the algorithms. Then, we normalized samples to a fixed size by a 50×50 square, so that the fps is unaffected by the height and width of video sequence. In the Alternating Direction Method of Multipliers (ADMM) optimization, we set the number of iterations to 4, which is enough to ensure algorithm robustness. Although more iteration is beneficial to improve algorithm performance, it will affect computational efficiency. The penalty factor μ is set to 0.25. The regularization factor λ_1 and λ_2 are set to 0.001 and 0.01, respectively. The iterations of the Gauss-Seidel method are set to be 5, and the learning rate is adopted as $\eta = 0.02$. The lower threshold $\mathcal{T}_{\text{low}} = 0.15$ and the upper threshold $\mathcal{T}_{\text{up}} = 0.8$. Moreover, bin color histogram is set into $32 \times 32 \times 32$, which has similar settings with the Staple tracker. All parameters remain fixed in all experiments. We run our approach approximately for 15 frames per second.

In practice, the penalty factor, the regularization factor, ADMM iteration number, and the Gauss-Seidel iteration number are independent of the specific scenario and only related to the proposed model itself. The cell size of HOG features, the size of the search region, and the bin color histogram, these three parameters will be affected by the target size and image resolution.

3.2 Qualitative evaluation

Figure 2 shows the tracking results of 10 state-of-the-art trackers and our tracker on some most challenging scenes (i.e., occlusion, deformation, fast motion, in-plane rotation, and out-plane rotation) in the OTB (object tracking benchmark) [30].

Figure 2a shows the tracking results on the *Box* and *Coke* sequences. In this figure, the targets undergo partial or short-term total occlusions. Our tracker can effectively track these objects, which benefits from two factors. First, our approach is capable of reducing or removing the impact of uncorrupted samples by readjusting the reliability weights of the sub model in training, thereby lowering the risk of drift and tracking failure. Second, the uncorrupted part of the targets and the background context can still form some uncorrupted samples for training and detection, which is similar to BACF. Most of the traditional correlation filter-based trackers DAT, CSK, ACorT, and DSST may drift after occlusions or fail to track the targets because their search areas are limited and uncorrupted samples are used during the training. BACF and SRDCF can handle some slight occlusions, but when undergo severe occlusions, they all drift away.

Figure 2b is the tracking results on the *KiteSurf* and *Couple* sequences. *KiteSurf* and *Couple* are two sequences of some representative sequences where the target is fast

Fig. 2 Tracking results of the proposed method and other algorithms on some representative sequences
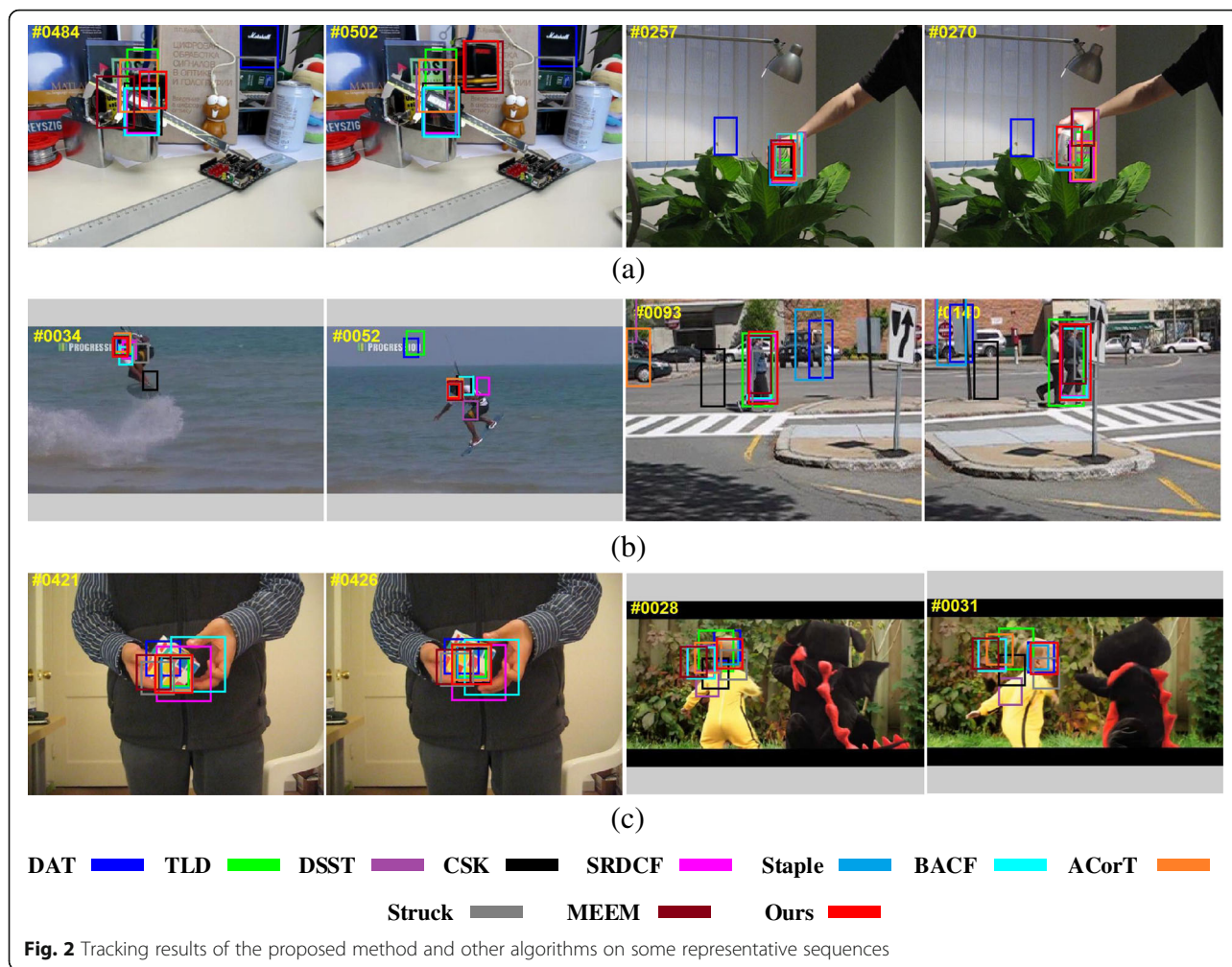
motion. The figure shows that the proposed tracker gain outstanding performance handles. This can be mainly attributed to the correlation filters and the discriminative ability of the proposed method. The proposed method works well in all these sequences. On the one hand, the color histogram information can be used effectively to handle deformations by dynamically increasing the impact of color histogram-based tracker. On the other hand, when target is in fast motion, the impact of correlation filter-based tracker will be dynamically increased. The sub tracker can effectively handle fast motion because allows larger search region. Other trackers also have some abilities against target deformations (i.e., Stape and MEEM), but most of them are failed to track targets when severe deformations are coming.

Figure 2c shows the tracking results on two representative sequences *Rubik* and *DragonBaby* s where the targets are partially or fully occluded. Our tracker performs well in in-plane rotation and out-plane rotation. These properties mainly benefit from up-weighting color histogram-based tracker.

For the six representative sequences above, we compared center location error of every sequence frame-by-frame in Fig. 3. These comparison results show that our tracker is more stable and accurate.

Table 1 reports average entry location errors for each compared tracker. In a descending order, the best three estimates are marked in red, blue, and green fonts. In one of the six sequences (the *Box* and *DragonBaby*), our approach achieved the best results. In two of the remaining four sequences (the *Rubik* and *Couple* sequences), our approach obtained the second best results and were very close to the best ones. In the remaining two sequences (*Coke* and *Kitesurf*), our tracker also performs better. Generally, our method performed well against existing trackers.

3.3 Quantitative evaluation

We use the precision plots and the success plots metric [30] to compare all trackers on OTB-100. Precision plot reports the average distance precision score at 20 pixels for each method. In the evaluation of success plots, the

Fig. 3 Frame-by-frame comparison of six representative sequences based on center location errors
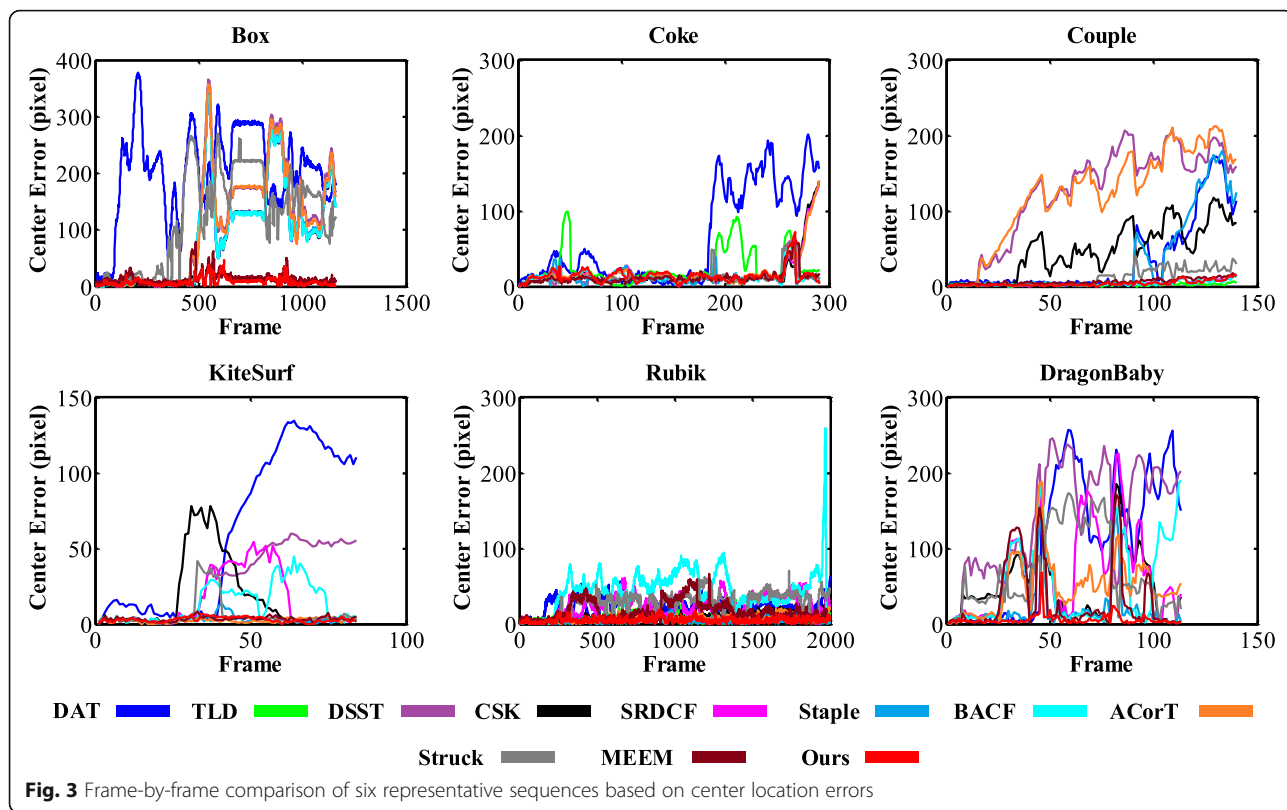
area under the curve (AUC) of success plots is used to rank the trackers.

The evaluation results of the proposed tracker and the 10 competitive trackers are demonstrated in Fig. 4. The legend of the precision plot reports the average distance precision score at 20 pixels for each method, and the legend of the success plot reports AUC

scores. The BACF tracker provides the second best results with precision plots of 82.0% and the success plots of 60.6%. The best results on this dataset is achieved by our tracker with a precision plot of 85.1% and a success plot of 62.8%, leading to a significant gain of 3.0 and 2.2% compared to BACF tracker in precision plots and success plots in turn.

Table 1 Comparison of results in terms of average entry location errors (in pixels)

Tracker	Box	Coke	DragonBaby	Kitesurf	Rubik	Couple	Average
Ours	**8.60**	14.20	**5.68**	2.87	**5.29**	**3.82**	**6.74**
DAT	199.97	62.97	106.72	59.03	20.95	34.94	80.76
TLD	**40.60**	25.08	116.58	62.10	8.76	**2.54**	42.61
DSST	106.79	12.79	142.57	28.80	**5.96**	125.15	70.43
CSK	89.13	18.65	50.40	17.27	9.38	47.56	38.73
SRDCF	89.29	18.90	69.38	15.28	24.26	**3.97**	36.85
Staple	90.85	**12.20**	**18.99**	**2.82**	**4.95**	34.54	**27.39**
BACF	89.20	12.66	52.14	13.65	44.83	4.11	36.10
ACorT	104.24	19.22	50.42	**2.26**	7.70	123.25	51.18
Struck	119.72	**12.08**	69.26	6.14	26.72	11.33	40.88
MEEM	**13.53**	**11.94**	**31.58**	3.29	16.93	6.50	**13.96**

Shi *et al. EURASIP Journal on Wireless Communications and Networking* (2018) 2018:191

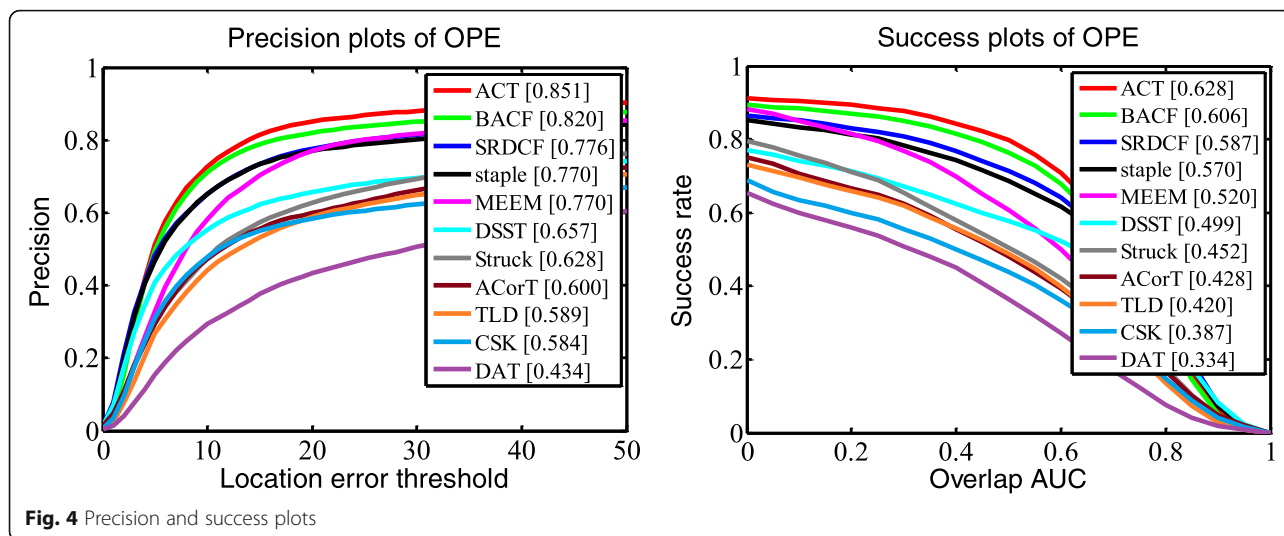Page 8 of 9

Fig. 4 Precision and success plots

Table 2 reports the AUC scores of all the compared trackers under some challenging problems which were posed in the OTB sequences including 31 sequences with target in-plane rotation (IPR), 39 sequences with target out-of-plane rotation (OPR), 19 sequences with target deformation (DEF), including 29 sequences with target occlusion (OCC) and 17 sequences with target fast motion (FM). In a descending order, the top three results are shown in red, blue, and green fonts. Our tracker achieves highest AUC scores in all these challenging attributes.

4 Conclusions

In order to alleviate the deformation, this paper presents to online learn an adaptive complementation tracking algorithm. The proposed strategy can dynamically adapt the importance of each sub tracker according to the real-time status of the scene, making the tracker more robust for various target appearance changes. In addition to, joint learning model reliability weights and tracker can also effectively control the reliability of training samples and improved the discriminability of the tracker. Experimental results show that our approach outperforms state-of-the art tracking algorithms on many challenging videos from the OTB.

Table 2 Average AUC of the ten trackers in terms of different attributes

	IPR	OPR	DEF	OCC	FM
Ours	0.62	0.62	0.61	0.62	0.60
BACF	0.59	0.60	0.59	0.61	0.58
Staple	0.55	0.56	0.54	0.56	0.50
SRDCF	0.54	0.57	0.56	0.58	0.53
DAT	0.32	0.35	0.36	0.35	0.29
TLD	0.41	0.41	0.36	0.40	0.41
DSST	0.50	0.48	0.45	0.49	0.42
CSK	0.39	0.37	0.34	0.37	0.31
ACorT	0.44	0.42	0.42	0.43	0.35
Struck	0.43	0.43	0.40	0.42	0.41
MEEM	0.50	0.52	0.48	0.50	0.49

Shi *et al. EURASIP Journal on Wireless Communications and Networking* (2018) 2018:191

Page 9 of 9

Availability of data and materials
All data are fully available without restriction.

Authors' contributions
GS and TX conceived and designed the algorithm and the experiments. GS was responsible for most of the implementation of the algorithms. JL, JG, ZR, and GS analyzed the data. GS provided suggestions for the proposed method and its evaluation, and assisted in the preparation of the manuscript. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References
1. AW Smeulders, DM Chu, R Cucchiara, S Calderara, A Dehghan, M Shah, Visual tracking: an experimental survey. IEEE Transactions on Pattern Analysis & Machine Intelligence 36, 1442–1468 (2014).
2. DA Ross, J Lim, R-S Lin, M-H Yang, Incremental learning for robust visual tracking. Int. J. Comput. Vis. 77, 125–141 (2008).
3. D Wang, H Lu, MH Yang, Online object tracking with sparse prototypes. IEEE Trans. Image Process. 22, 314–325 (2013).
4. X Mei, H Ling, in *IEEE International Conference on Computer Vision*. Robust visual tracking using ℓ 1 minimization (2009), pp. 1436–1443.
5. H Lu, X Jia, MH Yang, in *IEEE Conference on Computer Vision and Pattern Recognition*. Visual tracking via adaptive structural local sparse appearance model (2012), pp. 1822–1829.
6. T Zhang, B Ghanem, S Liu, N Ahuja, Robust visual tracking via structured multi-task sparse learning. Int. J. Comput. Vis. 101, 367–383 (2013).
7. T Zhang, B Ghanem, C Xu, N Ahuja, in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. Object tracking by occlusion detection via structured sparse learning (2013), pp. 1033–1040.
8. D Wang, H Lu, C Bo, Online visual tracking via two view sparse representation. IEEE Signal Processing Letters 21, 1031–1034 (2014).
9. S Avidan, Support vector tracking. IEEE Transactions on Pattern Analysis & Machine Intelligence 26, 1064–1072 (2004).
10. S Avidan, Ensemble tracking. IEEE Transactions on Pattern Analysis & Machine Intelligence 29, 261–271 (2007).
11. B Babenko, MH Yang, S Belongie, in *EEE Conference on Computer Vision and Pattern Recognition*. Visual tracking with online multiple instance learning (2009), pp. 983–990.
12. J Ning, J Yang, S Jiang, L Zhang, M-H Yang, in *IEEE Conference on Computer Vision and Pattern Recognition*. Object tracking via dual linear structured SVM and explicit feature map (2016), pp. 4266–4274.
13. DS Bolme, JR Beveridge, BA Draper, YM Lui, in *EEE Conference on Computer Vision and Pattern Recognition*. Visual object tracking using adaptive correlation filters (2010), pp. 2544–2550.
14. M Danelljan, G Häger, F Khan, M Felsberg, in *British Machine Vision Conference, Nottingham, September*. Accurate scale estimation for robust visual tracking (2014), pp. 1–5.
15. H JF, R C., P M., J B., High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis\s&\smachine Intelligence 37, 583 (2014).
16. C Ma, X Yang, C Zhang, MH Yang, in *EEE Conference on Computer Vision and Pattern Recognition*. Long-term correlation tracking (2015), pp. 5388–5396.
17. M Wang, Y Liu, Z Huang, Large margin object tracking with circulant feature maps. *IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp.4800–4808.
18. R Yao, S Xia, F Shen, Y Zhou, Q Niu, Exploiting spatial structure from parts for adaptive kernelized correlation filter tracker. IEEE Signal Processing Letters 23, 658–662 (2016).
19. JF Henriques, R Caseiro, P Martins, J Batista, High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. 37, 583–596 (2015).
20. M Danelljan, G Häger, FS Khan, in *British Machine Vision Conference*. Accurate scale estimation for robust visual tracking (2014), pp. 65.1–65.11.
21. AS Montero, J Lang, R Laganière, in *IEEE International Conference on Computer Vision Workshop*. Scalable kernel correlation filter with sparse feature integration (2016), pp. 587–594.
22. T Liu, G Wang, Q Yang, in *IEEE Conference on Computer Vision and Pattern Recognition*. Real-time part-based visual tracking via adaptive correlation filters (2015), pp. 4902–4912.
23. P Pérez, C Hue, J Vermaak, M Gangnet, Color-based probabilistic tracking. ECCV, 2002 I, 661–675 (2002).
24. D Comaniciu, V Ramesh, P Meer, Kernel-based object tracking. IEEE Transactions on Pattern Analysis & Machine Intelligence 25, 564–575 (2003).
25. H Possegger, T Mauthner, H Bischof, in *IEEE Conference on Computer Vision and Pattern Recognition*. In defense of color-based model-free tracking (2015), pp. 2113–2120.
26. L Bertinetto, J Valmadre, S Golodetz, O Miksik, PHS Torr, in *Computer Vision and Pattern Recognition*. Staple: complementary learners for real-time tracking (2016), pp. 1401–1409.
27. HK Galoogahi, A Fagg, S Lucey, in *IEEE Conference on Computer Vision and Pattern Recognition*. Learning background-aware correlation filters for visual tracking (2017), pp. 1135–1143.
28. M Danelljan, G Häger, FS Khan, M Felsberg, in *IEEE Conference on Computer Vision and Pattern Recognition*. Adaptive decontamination of the training set: a unified formulation for discriminative visual tracking (2016), pp. 1430–1438.
29. S Boyd, N Parikh, E Chu, B Peleato, J Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations & Trends in Machine Learning 3, 1–122 (2010).
30. Y Wu, J Lim, M-H Yang, Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. 37, 1834–1848 (2015).
31. S Hare, S Golodetz, A Saffari, V Vineet, M-M Cheng, SL Hicks, et al., Struck: structured output tracking with kernels. IEEE Trans. Pattern Anal. Mach. Intell. 38, 2096–2109 (2016).
32. Z Kalal, K Mikolajczyk, J Matas, Tracking-learning-detection. IEEE Transactions on Pattern Analysis & Machine Intelligence 34, 1409–1422 (2012).
33. M Danelljan, G Hager, F Shahbaz Khan, M Felsberg, in *IEEE International Conference on Computer Vision*. Learning spatially regularized correlation filters for visual tracking (2015), pp. 4310–4318.
34. Zhang J, Ma S, Sclaroff S. MEEM: robust tracking via multiple experts using entropy minimization. *European Conference on Computer Vision* (2014), pp. 188–203.
35. M Danelljan, FS Khan, M Felsberg, JVD Weijer, in *IEEE Conference on Computer Vision and Pattern Recognition*. Adaptive color attributes for real-time visual tracking (2014), pp. 1090–1097.